

TECNOLOGIA INFORMATICA DEL MAS ALTO NIVEL

ANALES

COMDEX/INFOCOM ARGENTINA '97

**CONGRESO Y EXPOSICION INTERNACIONAL
DE INFORMATICA Y TELECOMUNICACIONES**

BUENOS AIRES, 20 AL 23 DE MAYO DE 1997

Queda hecho el depósito que indica la Ley 11.723

Se autoriza la reproducción parcial o total
bajo cualquier forma o medio a condición de indicar la fuente

Copyright 1997 INFOCOM

Compilado por

Ramón García Martínez (Instituto Tecnológico de Buenos Aires)

Bibiana Rossi (Universidad Tecnológica Nacional - F.R.B.A.)

Ramón Gerónimo Brenna (Universidad del Salvador)

Editado por **Nadia Albajari** (INFOCOM)

COMDEX/INFOCOM ARGENTINA '97

AUTORIDADES DE LAS ENTIDADES ORGANIZADORAS

<i>JORGE CASSINO</i>	Presidente - CESSI
<i>LUIS PERAZO</i>	Presidente - CICOMRA
<i>MARCO GIBERTI</i>	Director General - E. J. KRAUSE Y ASOC. ARGENTINA
<i>RICHARD BLOUIN</i>	Vicepresidente - SOFTBANK COMDEX
<i>GABRIEL LOPEZ</i>	Presidente - UNIFORM ARGENTINA
<i>NORBERTO TORRERA</i>	Presidente - USUARIA

AUTORIDADES DEL CONGRESO

<i>NESTOR PERPETUA</i>	Presidente del Congreso - INFOCOM
<i>LUIS CANESSA</i>	Vicepresidente del Congreso - INFOCOM
<i>PATRICIO O'KON</i>	Gerente de Conferencias - E. J. KRAUSE Y ASOCIADOS ARGENTINA

MIEMBROS DEL CONGRESO

GRUPO DE QUALITY ASSURANCE

GRACIELA ARDILES
JUAN JOSE DELL ACQUA

MIGUEL ANGEL PEREZ AMICONE
MANUEL WIAGGIO

RESPONSABLES DE AREA TEMATICA

TECNOLOGIA EDUCATIVA Y SOCIEDAD

RAUL ARRIBAS
RAMON BRENNA

ALICIA CARR
CARLOS MARZOA URBINA

STELLA MARIS VALIENTE
JUAN CARLOS VENDITTI

TECNOLOGIA Y NEGOCIOS

VIVIANA BLANCO
SERGIO CUTRERA

ALEJANDRO DEBENEDET
JAVIER OURET

RICARDO KARPOVICH
SUSANA PISANTE

COMITE ACADEMICO

*DA VID AIRALA
HORACIO CAÑEQUE
EDUARDO F. CASULLO
PABLO COSSO*

*BERNARDO J. DELL 'ORO
EDUARDO FONTANA
RAMON GARCIA MARTINEZ*

*ESPEDITO PASARELLO
SILVIA RAMOS
BIBIANA ROSSI
ADRIAN SETTON*

GRUPO DE APOYO DEL CONGRESO

*SERGIO ABEDUTTO
EDUARDO BALEANI
FRANCISCO BARCIA
CARLOS BISCAY
JORGE CLOT*

*VERONICA GANDINI
PATRICIA GUTIÉRREZ
ANTONIO MARSIGLIO
LUCIANA ORECCHIO
RUBEN ORECCHIO*

*LILIANA PERSICO
ADRIANA PRADO
LIDIA SERATTI
ALBERTO UHALDE
MARIANO WECHSLER*

COORDINACION LOGÍSTICA U ADMINISTRATIVA

*NADIA ALBAJARI
HECTOR MARINIONI*

*PATRICIA OUTEDA
HUMBERTO RIVERO*

EVALUADORES

Raúl Arribas. Colegio De La Salle
Ramón Brenna. Universidad del Salvador
Paola Britos. Universidad Tecnológica Nacional - FRBA // ITBA
Fernando Caram. Universidad de Buenos Aires
Horacio Cañeque. Ministerio de Defensa
José F. Carbone. Instituto Superior de Formación Técnica
Alicia Carr. Comisión de Educación - USUARIA
Sandra Cartagenova. Universidad FASTA
Eduardo Casullo. Universidad de Belgrano
Edgardo Raúl Claverie. Universidad Tecnológica Nacional - FRBA
Jorge Clot. Universidad Católica Argentina
Sergio Cutrera. S. S. A.
Alejandro Debenedet. BMW
Luis Donadio. Universidad de Mar del Plata
Alejandra Esteban. Universidad de Mar del Plata
Victor Fernández. Universidad de Buenos Aires
Héctor Gonzalez. Universidad Tecnológica Nacional - FRBA
Ramón García Martínez. Instituto Tecnológico de Buenos Aires
Edmundo Noé Gramajo. Instituto Tecnológico de Buenos Aires
Guillermo Hudson. Instituto Tecnológico de Buenos Aires
Laura Luchini. Universidad Nacional de Luján
Antonio Marsiglio. Universidad Nacional de Mar del Plata
Carlos Marzoa Urbina. Estudio Risso
Gnillermo Merlo. Universidad de Buenos Aires
Mario Oloriz. Universidad Nacional de Luján
Adolfo E. Onaine. Universidad de Buenos Aires
Ruben Orecchio. Comisión de Educación - USUARIA
Espedito Pasarello. Estudio Pasarello y Asociados
Miguel Angel Pérez Amicone. Asociación de Analistas de Sistemas
Florencia Pollo. Universidad Tecnológica Nacional - FRBA
Ramiro Priegue. Universidad de Buenos Aires
Juan Ramonet. Universidad de Buenos Aires
Silvia Ramos. Universidad de Buenos Aires
Carlos Rico. Universidad de Mar del Plata
Bibiana Rossi. Universidad Tecnológica Nacional - FRBA // ITBA
José Tana. Universidad Tecnológica Nacional - FRBA
Stella Maris Valiente. Universidad Nacional de Mar del Plata
Juan Carlos Venditti. W44 Consultora
Manuel Wiaggio. Museo de la Informática

PROLOGO

Todo Congreso es un acontecimiento inigualable para articular la transferencia de tecnologías entre los centros de investigación y los centros de producción.

Cada uno de los trabajos que forman este Anal de **COMDEX/INFOCOM ARGENTINA '97 - Congreso y Exposición Internacional de Informática y Telecomunicaciones** - trasunta la confirmación alentadora de que gradualmente va cristalizándose una asociación entre Producción y Centros de Generación de Conocimiento con el propósito de enfrentar los numerosos desafíos que se derivan de un mundo tecnificado que exige altos estándares de calidad a las industrias regionales como requisito de incorporación al circuito de comercio internacional.

Nos complace presentar a la comunidad una síntesis documentada de este proceso de asociación.

MIEMBROS DEL CONGRESO
Buenos Aires, Mayo de 1997

COMDEX/INFOCOM ARGENTINA '97

ANALES

CONGRESO INTERNACIONAL DE INFORMATICA Y TELECOMUNICACIONES

APLICACIONES / APPLICATIONS

Una evaluación a nivel de conexiones virtuales de métodos combinados para la asignación del ancho de banda y Ba atención del tráfico en ATM.

VILLAPOL., M. ; ISERN, G.

Grupo de investigación de Redes, Arquitectura y Sistemas. Escuela de Computación. Facultad de Ciencias.

Universidad Central de Venezuela.

Sistema de adquisición de datos y control para un frenómetro de rodillos de vehículos de carga y transporte

16

ANSALDI, J. (1); LOPEZ, E. (2)

(¹)Facultad Regional Concepción del Uruguay - Entre Ríos

Universidad Tecnológica Nacional

(²) Instituto Superior de Electrónica de Entre Ríos

Sistema VLSI CAD para la resolución de circuitos secuenciales

26

BELFORTI, G.; GAYOSO, C.; GONZALEZ, C. ; ARNONE, L.

Laboratorio de Componentes Electrónicos. Facultad de Ingeniería.

Universidad Nacional de Mar del Plata.

EDUCACION Y SOCIEDAD / EDUCATION & SOCIETY

Ortografía 1: Software Educativo

37

PRUZZO, V. ; DI PEGO, C.

Instituto Superior de Estudios Scopedagógicos y Sociales. Facultad de Ciencias Humanas.

Universidad Nacional de la Pampa

Using microcomputers to teach policy studies substance

48

NAGEL, S.

University of Illinois

Primeros pasos en transformaciones lineales

57

CALIGARIS, M.; RODRIGUEZ, G. ; CALIGARIS, R.

Grupo de Informática Educativa. Facultad Regional San Nicolás.

Universidad Tecnológica Nacional.

Nuevas herramientas en la enseñanza de la física: circuitos eléctricos	72
CALIGARIS, R. ; CALIGARIS, M. ; MANSILLA, G. <i>Grupo de Informática Educativa. Facultad Regional San Nicolás. Universidad Tecnológica Nacional.</i>	
Relación de la enseñanza de computación con las teorías del aprendizaje	87
HAVLIK, J. <i>Universidad de Buenos Aires</i>	
An experience in a distance graduate course on software engineering	97
GARCÍA MARTÍNEZ, R.; ROSSI, B. ; GRAMAJO, E. <i>CAPIS. Graduate School, Buenos Aires Institute Of Technology</i>	
El Teletrabajo	104
ALLES, A. <i>Top Management</i>	
Ambiente basado en hipermedios como soporte al aprendizaje de la tecnología de objetos	117
NICOLAU, S.; OLSINA, L.; LAFUENTE, G. ; BERTONE, E. <i>Grupo de Investigación y Desarrollo en Ingeniería de Software. Departamento de Computación. Facultad de Ingeniería. Universidad Nacional de La Pampa</i>	

PROCESAMIENTO DE IMAGENES / IMAGE PROCESSING

Técnicas aceleradas de visualización de volúmenes	128
GIULIETTI, A. ; DELRIEUX, C. <i>Departamento de Ingeniería Eléctrica. Instituto de Ciencias e Ingeniería de Computación. Universidad Nacional del Sur</i>	
Extracción de parámetros en imágenes forenses mediante morfometría analítica	143
GONZÁLEZ, M.; OLMO, D.; MOLER, E.; PESSANA, F. ; BALLARIN, V. <i>Laboratorio de Procesos y Mediciones de Señales. Facultad de Ingeniería. Universidad Nacional de Mar del Plata.</i>	

SISTEMAS INTELIGENTES / INTELLIGENT SYSTEMS

Posicionamiento automático de celdas normalizadas mediante redes neuronales 154

QUIJANO, A.; GAYOSO, C. ; STELLI, I.

*Laboratorio de Componentes Electrónicos, Facultad de Ingeniería.
Universidad Nacional de Mar del Plata*

Revisión racional de teorías basadas en plausibilidad 164

DELRIEUX, C.

*Departamento de Ingeniería Eléctrica. IDID - Instituto de Ciencias e Ingeniería de Computación.
GIIA - Grupo de Interés en Inteligencia Artificial
Universidad Nacional del Sur*

Aplicación de redes neuronales para la predicción de series de tiempo 175

FORNARI, J.; MARTÍNEZ, E. ; PÉREZ, G.

INGAR. Instituto de Diseño y Desarrollo

An overview of an autonomous intelligent system with an embedded Machine Learning Mechanism 184

GARCÍA MARTÍNEZ, R.

*CAPIS Buenos Aires Institute Of Technology & Intelligent Systems Laboratory, Computer Science
Department, School Of Engineering.
University Of Buenos Aires*

Calidad de bases de conocimiento. Una medida de evaluación 191

GARCÍA MARTÍNEZ, R.

*Laboratorio de Sistemas Inteligentes, Departamento de Computación. Facultad de Ingeniería.
Universidad de Buenos Aires.
CAPIS. Escuela de Posgrado. Instituto Tecnológico de Buenos Aires.*

MODELOS / MODELS

Implementación computacional a las soluciones clásicas del problema del regateo 199

QUINTAS, L. (1) ; CAVALIE, P. (2) ; WELCH, D. (2)

*(1) Departamento de Matemática - IMASL (Instituto de Matemática Aplicada San Luis) Facultad de Ciencias Físico Matemáticas y Naturales. Universidad Nacional de San Luis. (2) Departamento de informática. Facultad de Ciencias Físico Matemáticas y Naturales.
Universidad Nacional de San Luis.*

Análisis de refinamientos del equilibrio de NASH usando implementaciones computacionales 213

QUINTAS, L. ; NECCO, C.

*IMASL - Instituto de Matemática Aplicada San Luis. Facultad de Ciencias Físico Matemáticas y Naturales.
Universidad Nacional de San Luis.*

SPF - Un ambiente de programación para FP 228

QUINTAS, L.; FUNES, A. ; KAVKA, C.

IMASL - Instituto de Matemática Aplicada San Luis. Facultad de Ciencias Físico Matemáticas y

Universidad Nacional de San Luis.

Determinación de rutas óptimas de distribución para la ciudad de Bahía Blanca 242

CASAL, R. ; CORRAL, R. ; LOPEZ, N. ; ZITO, E. ; LOVOTTI, L.

Departamento de Ingeniería.

Universidad Nacional Del Sur.

INGENIERIA DE SOFTWARE / SOFTWARE ENGINEERING

Cuando la normalización conduce a un modelo de datos ilógico 257

ROSSI, B.; PICA, A.; GONZALEZ, H. ; BALBI, R.

Grupo LIDIS. Facultad Regional Buenos Aires.

Universidad Tecnológica Nacional.

HERRAMIENTAS / TOOLS

Ignatius: una herramienta para construir sistemas de supervisión de procesos industriales 272

BENÍTEZ, S.; SEOANE, J.; WAINER, G.; BEVILACQUA, R.

Departamento de Computación, Facultad de Ciencias Exactas y Naturales.

Universidad de Buenos Aires

ALFANEGO: Sistema de relevamiento, análisis y recomendación de actividades agropecuarias 283

UZAL, R.; LUCERO, M., SANCHEZ, A.; DASSO, A. ; MONTEJANO, G.

Departamento de Informática

Universidad Nacional de San Luis

Control estadístico de materiales 297

CARTAGENOVA, S. ; LISNIZER, M.

Universidad F. A.S. TA. - Pharma Argentina

APLICACIONES / APPLICATIONS

TITULO:	UNA EVALUACION A NIVEL DE CONEXIONES VIRTUALES DE METODOS COMBINADOS PARA LA ASIGNACION DEL ANCHO DE BANDA Y LA ATENCION DEL TRAFICO EN ATM.
AUTOR:	VILLAPOL, MARIA ELENA ; ISERN, GERMINAL.
INSTITUCION:	GRUPO DE INVESTIGACION DE REDES, ARQUITECTURA Y SISTEMAS. ESCUELA DE COMPUTACION. FACULTAD DE CIENCIAS. UNIVERSIDAD CENTRAL DE VENEZUELA.
DIRECCION:	APARTADO 49125 - CARACAS - VENEZUELA

TITULO:	SISTEMA DE ADQUISICION DE DATOS Y CONTROL PARA UN FRENOMETRO DE RODILLOS DE VEHICULOS DE CARGA Y TRANSPORTE
AUTOR:	ANSALDI, JUAN CARLOS PABLO (1) ; LOPEZ, ESTEBAN HORACIO (2)
INSTITUCION:	(1) FACULTAD REGIONAL CONCEPCION DEL URUGUAY - ENTRE RIOS UNIVERSIDAD TECNOLOGICA NACIONAL (2) INSTITUTO SUPERIOR DE ELECTRONICA DE ENTRE RIOS
DIRECCION:	(1) ING. PEREIRA 676 - (3260) - C. DEL URUGUAY - ENTRE RIOS - ARGENTINA

TITULO:	SISTEMA VLSI CAD PARA LA RESOLUCION DE CIRCUITOS SECUENCIALES
AUTOR:	BELFORTI, GABRIEL; GAYOSO, ARTURO; GONZALEZ, CLAUDIO ; ARNONE, LEONARDO.
INSTITUCION:	LABORATORIO DE COMPONENTES ELECTRONICOS. FACULTAD DE INGENIERIA. UNIVERSIDAD NACIONAL DE MAR DEL PLATA.
DIRECCION:	J. B. JUSTO 4302 - (7600) - MAR DEL PLATA - BUENOS AIRES

Una evaluación a nivel de conexiones virtuales de métodos combinados para la asignación del ancho de banda y la atención del tráfico en ATM

G. Isern, M. Villapol

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Grupo de Investigación de Redes, Arquitectura y Sistemas
Apartado 49125, Caracas Venezuela
e-mail: isern@reacciun.ve, mvilla@reacciun.ve

Palabras claves: BISDN, ATM, ancho de banda, reservación del ancho de banda, conexión virtual, clases de tráfico.

Resumen

Este artículo trata el problema de la asignación del ancho de banda en BISDN. En el trabajo se evaluaron estrategias de reservación las cuales se dividieron en dos grandes categorías, estrategias para la distribución de la asignación del ancho de banda y estrategias para la atención de distintas clases de tráfico. La solución está basada en los siguientes aspectos: (a) las redes evaluadas están compuestas por nodos de acceso e intermedios (conmutadores de VC/VP), (b) hay un servidor en cada nodo el cual atiende las diferentes clases de tráfico y asigna el ancho de banda, (c) se, utiliza simulación por eventos discretos para evaluar las distintas estrategias, (d) el rendimiento de cada una de las estrategias se mide en base al número de llamadas rechazadas y es evaluado usando un programa elaborado para tal fin.

1. Introducción

La Red Digital de Servicios Integrados (BISDN: *Broadband Integrated Services Digital Network*) se define como una red de área amplia de alta velocidad (sobre los 100 Mbps) capaz de soportar servicios de voz, datos, imagen, vídeo y audio, los cuales serán transmitidos en forma digital y de manera integrada,, es decir, se usa un mismo tipo de conmutador para transportar todos estos tipos de información. Las primeras especificaciones de los diferentes aspectos que conforman esta red fueron publicadas por el CCITT como parte de las recomendaciones de la Serie-I en 1988. Uno de los aspectos mas relevantes fue la decisión del ITU-T de desarrollar BISDN usando el Modo de Transferencia Asíncrono (*ATAM: Asynchronous Transfer Mode*). ATM define la forma de conmutar y multiplexar las información que viaja en paquetes de longitud fija cuyo tamaño es de 53 bytes. Para ello ATM provee dos tipos de conexiones virtuales, los Caminos Virtuales (*VPs: Virtual Paths*) y los Canales Virtuales (*VCs: Virtual Channels*) [18].

Uno de los puntos más importantes que debe tomarse en cuenta en una red de banda ancha es la reservación de los recursos solicitados para una conexión, de forma tal de poder controlar y garantizar que se va a cumplir con un cierto contrato de servicio asociado a la misma. Esto debido, principalmente, a que estas redes soportan velocidades de transmisión superiores a los 100 Mbps e integran diversos tipos de servicios de voz, dato, audio, imagen y vídeo los cuales tienen distintos requerimientos a nivel de la máxima tasa de celdas, del retardo fin-fin de la celda y de la tolerancia al retardo por celda, entre otros.

La asignación del ancho de banda es una función que se incluye en el proceso de Control de Tráfico y Congestión en B-ISDN. Las funciones de control de tráfico y congestión se pueden estructurar en tres niveles ([3], [4], [7], [8], [9], [17]): nivel de VP o nivel de Red, nivel de VC o nivel de llamada y nivel de celda. Las funciones de asignación y control de la reservación del ancho de banda se encuentran distribuidas en estos niveles.

a) Nivel de VP o de Red: en este nivel se ejecutan las funciones de control y manejo de los recursos de la red (*NRM: Network Resource Management*). Entre estas se destacan la asignación del ancho de banda por grupo de conexiones. Las funciones de control y manejo se realizan a nivel de los VPs; los cuales agrupan una serie de VCs, haciendo más simple y flexible la asignación y reservación de los recursos.

Las funciones a nivel de VP controlan y ejecutan la asignación lógica del ancho de banda por un período determinado, basándose en las mediciones de la red en tiempo real y de las demandas pronosticadas.,.

6) Nivel de VC o de Llamada: en este nivel se ejecuta el procesamiento y admisión de las llamadas de acuerdo a la cantidad de ancho de banda reservado por 'el VP en el período en cuestión. Una llamada es admitida si existen suficientes recursos para garantizar los parámetros de tráfico y calidad de servicio requeridos [18]. La decisión de admisión de una llamada se basa en el contenido del descriptor de tráfico, el ancho de banda disponible en las Conexiones de Camino Virtual (*VPCs: Virtual Path Connections*) entre origen y destino y las rutas existentes entre estos últimos puntos.

c) Nivel de Celda: este nivel controla que durante el progreso de una llamada, esta se comporte de acuerdo a las características acordadas en el establecimiento de la conexión. En él se realizan las funciones de control de congestión preventivo (antes de que se produzca congestión) o reactivo (para aliviar la misma una vez que se ha producido). En este nivel se ubican las funciones correspondientes al Control de los Parámetros de Uso/Red (*UPC/NPC: Usage Parameters Control/ Network Parameters Control*) [11].

Los niveles de red y de llamada están asociados con el problema de la asignación del ancho de banda sobre los VPs y sobre los VCs, respectivamente. En el nivel de VP se decide como las distintas clases de tráfico compartiran el ancho de banda disponible. sobre las Conexiones de VP (VPCs) [5]. Dicha información es pasada al nivel de VC el cual usa la misma junto con los parámetros- de tráfico y de Calidad de Servicio (QOS: Qualite of Service) para tomar una decisión con respecto si la llamada será aceptada o rechazada y hace efectiva ta asignación del ancho de banda en el caso correspondiente.

La asignación de ancho de banda permite reservar al mismo durante una conexión o durante periodos de ráfaga. Sin embargo, en este trabajo se trata sólo con la reservación del mismo en el período de duración 'de una conexión, a nivel de red o VP y de llamada o VC.

9-A. Estrategias para la asignación del ancho de banda

Las clases de servicio son usadas por una red ATM para diferenciar los tipos de conexión solicitadas a través de sus requerimientos de tráfico y de QOS (especificados en los parámetros). Las clases de servicio definidas. por el ATM Forum [1] son: Tasa de Bit Constante (CBR: Constant *Bit Rate*), Tasa de Bit Variable-Tiempo Real/tiempo no Real (VBR ($RT/(NRT)$: *Variable Bit Rafe (Real Time)/(Non-Real Time)*)), Tasa de Bit Disponible (*ABR: Available Bit Rafe*), Tasa de Bit No Especificada (*UBR: Unspecified Bit Rafe*). Sin embargo, para fines de este trabajo estos tipos de servicio se reagruparon de forma tal de reflejar sus necesidades a nivel de reservación del ancho de banda, de la siguiente *manera: categoría de servicio con ancho de banda no reservado, categoría de servicio con ancho de banda reservado y categoría de servicio con ancho de banda reservado durante la ráfaga (*burst*) [18]. En esta investigación se trata sólo con aquellos servicios que realizan una reservación del' ancho do banda para toda la conexión, tales como los servicios CBR y VBR.

Por otra parte, de los parámetros de tráfico y de QOS definidos por el ITU-T [1], [12] sólo se considera el de la Tasa de la Celda Máxima (*PCR: Peak Celt Rafe*) el cual se define como la tasa máxima instantánea a la cual el usuario puede transmitir [12]. La asignación del ancho de banda en esta investigación se realiza en base a este parámetro.

De esta forma, las estrategias para la asignación del ancho de banda se dividen en dos categorías: estrategias para la distribución de la asignación del ancho de banda y estrategias para la atención de las múltiples clases de tráfico. Las estrategias de atención de las distintas clases de tráfico son: Primero en entrar primero en salir (FIFO: *Firsf In Firsf Out*), por Prioridad y Cíclica. La estrategia FIFO se basa en seleccionar al tráfico encolado en el nodo origen de la red BISDN de acuerdo al tiempo de arribo al sistema, sin

tomar en cuenta ninguna prioridad. En la segunda estrategia, la basada en prioridades, atiende primero los requerimientos con mayor prioridad, donde los criterios para establecer esta última pueden ser varios tales como el PCR requerido y la importancia del tipo de tráfico para la red en cuestión. En este trabajo los niveles de prioridad están limitados a cinco y el mayor nivel es el cero. Por último, en la estrategia de control cíclico se combinan las estrategias FIFO y la de prioridad. Para ello las distintas colas asociadas a cada clase de tráfico son atendidas en forma cíclica, e igual que en la estrategia de prioridades, cada requerimiento dentro de una cola es atendido usando una estrategia FIFO. De esta manera a cada clase de tráfico se le da una misma prioridad de atención.

Las estrategias para la distribución de la asignación del ancho de banda son: la estrategia de compartición total y la estrategia de establecimiento de particiones. En ellas se considera la asignación del ancho de banda sobre conexiones virtuales, específicamente sobre Conexiones de VP (*VPCs: Virtual Path Connections*). Donde un VPC está formado por un conjunto de VPs existentes entre dos puntos definidos como conmutadores de VPNC [2].

En la distribución del ancho de banda por compartición total, el ancho de banda que se encuentra disponible en un VPC puede ser asignado indistintamente a cualquier conexión, es decir, el ancho de banda está disponible para ser compartido por todos los requerimientos asociados a las distintas clases de tráfico. A pesar de que esta estrategia es sencilla de implantar presenta el inconveniente de que el ancho de banda disponible puede ser usado por cualquier tipo de tráfico, así el tráfico que necesita mas ancho de banda tiene que competir con el tráfico que requiere menos ancho de banda.

Por otro lado, la estrategia de establecimiento de particiones divide el ancho de banda en canales los cuales tendrán una porción del ancho de banda total del VPC. Para ser mas preciso sea MAD, el *máximo ancho de banda disponible* para un VPC, cada partición tendrá un máximo de ancho de banda para la partición, MAP_i, donde $i = 0,1,2,\dots,n$, con n el número de particiones establecidos para un VPC. La relación entre estas dos variables sería :

$$\sum_{i=0}^n MAP_i \leq MAD$$

Esto es igual para cada VPC donde se tiene un MAD distinto para cada una, es decir, un *máximo ancho de banda disponible por VPC*, MADP_j, con $j=0,1,\dots,m$, m es el número de VPCs en un momento dado. Generalizando la fórmula anterior para todos los VPCs en un nodo determinado se tiene:

Estrategias para la distribución de la asignación del ancho de banda / Estrategias para la atención de las múltiples clases de tráfico	Compartición total	Particiones
First In First Out (FIFO)	aplicable	aplicable
Prioridad	aplicable	aplicable
Cíclico	aplicable	aplicable

Tabla 1: Relación entre las estrategias para la distribución de la asignación del ancho de banda y las de atención de múltiples clases de tráfico.

$$\sum_{i=0}^n MAP_{ij} \leq MADP_j$$

$$j= 0,1,\dots, m$$

Esta estrategia es más compleja que la anterior, pero presenta la ventaja de ser más flexible porque permite asignar diferentes, particiones y tamaños de particiones a las distintas clases de tráfico basado en el PCR asociado a cada una de ellas.

El problema de la asignación del ancho de banda en redes integradas que soportan múltiples tipos de tráfico ha sido estudiado desde la mitad de los años 80s [7], [14], [16], [17]. Por ejemplo, algunos son estudios aplicados a cualquier tipo de red integrada [16], [17] y otros han basado la evaluación de las estrategias de asignación en modelos analíticos [7], [16]. Este trabajo utiliza un programa, creado para evaluar las diferentes estrategias de asignación de ancho de banda sobre conexiones virtuales, en base a la ejecución de múltiples pruebas sobre diferentes configuraciones de red.

El objetivo principal de esta investigación es evaluar las diferentes combinaciones de estrategias mostradas en la Tabla 1, en base al número de llamadas rechazadas en función de la carga de tráfico usando una programa de simulación creado para tal fin. Esta evaluación permite dar ciertas recomendaciones que podrían mejorar el rendimiento de una red en términos de la utilización del ancho de banda y el número de llamadas rechazadas,

Este documento está organizado de la siguiente manera. En la sección II, se presenta el conjunto de pasos seguidos para resolver el problema de la evaluación de las diferentes combinaciones de estrategias de asignación del ancho de banda descritas en la sección I-A. El modelo de encolamiento se describe en la sección II-A. En la sección II-B se presenta el proceso de

simulación. El programa desarrollado para evaluar las estrategias de asignación de ancho de banda se describe en la sección II-C. En la sección III, se muestra el rendimiento obtenido con cada una de las distintas estrategias de asignación de ancho de banda en términos de las llamadas rechazadas en función de la carga total de tráfico con base a un caso de estudio. Los parámetros de salida se dan en la sección III-A. Mientras que los resultados son dados en la sección III-B. Por último la sección IV resume las conclusiones del trabajo,

II. Evaluación de las estrategias de ancho de banda

La solución al problema de la evaluación de las estrategias descritas en la sección I se basa en los siguientes pasos: definición de la configuración de la red, definición del modelo de encolamiento, especificación del proceso de simulación y elaboración del programa para la implantación, evaluación y comparación de las estrategias de asignación del ancho banda.

Con respecto a la configuración de la red, se considera que la misma está formada por un conjunto de nodos de acceso a través de los cuales los usuarios realizan sus requerimientos y de los nodos intermedios los cuales, realizan funciones de relevo de las celdas asociadas a cada conexión establecida. Todos estos nodos son conmutadores de VP/VC.

II-A Modelo de encolamiento

Cada nodo de acceso tiene múltiples colas, una por cada clase de tráfico. En cada nodo de acceso existe un servidor que realiza las funciones de control de admisión y de administración de los recursos de red. En cada nodo intermedio hay un servidor que trata de reservar el ancho de banda para un VPC determinado, basado en la tabla de enrutamiento correspondiente.

A cada nodo de acceso llegan muchas llamadas, cuyos arribos se distribuyen según una Poisson. La duración de la conexión, por otra parte, sigue una distribución exponencial. Así, un tipo de tráfico i se caracteriza por una tasa de arribo h_i llamadas en la unidad de tiempo (en este caso segundos), una duración promedio de la conexión de $1/\mu_i$ y el PCR,.

II-B Proceso de simulación

La evaluación de las estrategias de ancho de banda se realiza usando la simulación por eventos discretos. Los diferentes eventos para cada nodo se describen a continuación. Para cada nodo de acceso:

a. Llegada de una llamada: una llamada correspondiente al tipo de tráfico llega al nodo j .

b. Fin de la asignación del ancho de banda: se termina de tratar de asignar el ancho de banda para un requerimiento del tipo de tráfico *i*, basado en alguna combinación de las estrategias descritas en el capítulo anterior. Se puede reservar o no el ancho de banda requerido dependiendo de la disponibilidad del mismo en las rutas existentes entre el nodo origen y el destino.

c. liberación de una conexión: se finaliza el tiempo de conexión para una llamada del tipo de tráfico *i*.

Para el caso de un nodo intermedio se tienen todos los eventos anteriores excepto por el arribo de una llamada.

Cuando una llamada llega a un nodo de acceso el servidor la atiende inmediatamente, si está libre, en caso contrario la misma se encola. Cuando el servidor termina de asignar el ancho de banda siguiendo alguna de las estrategias de distribución de la asignación del ancho de banda, el toma otra llamada de las colas asociadas a cada tipo de tráfico siguiendo, alguna estrategia de atención de las múltiples clases de tráfico. El proceso de asignación del ancho de banda puede concluir exitosamente lo cual implica que se encontró la cantidad de ancho de banda requerida a lo largo de la ruta formada por los VPCs, o pudo terminar sin éxito alguno lo cual implica que no se pudo reservar el ancho de banda sobre la ruta determinada. En este último caso, la llamada es reencolada en el nodo origen si y solo si existe otra ruta disponible. Si no existe otra ruta, la llamada es rechazada. Finalmente, cuando se finaliza el tiempo de conexión de una llamada el ancho de banda asignado es liberado a lo largo de toda la ruta.

Cabe destacar que en este trabajo se considera el enrutamiento como estático, es decir las tablas de rutas se cargan en el momento de inicializar la red y permanecen sin cambios mientras la red este en funcionamiento. Puede existir una ruta entre cada nodo origen-destino o múltiples rutas hasta un máximo de tres.

II-C Programa para la evaluación de estrategias para la asignación del ancho de banda en BISDN/ATM

Se desarrolló un programa para, evaluar las diferentes combinaciones de estrategias de asignación de ancho de banda. Esta herramienta está basada en la simulación por eventos discretos y el modelo de encolamiento de servidor único descrito en la sección II-A. Este programa, igualmente, soporta enrutamiento estático simple o alterno.

El programa acepta múltiples clases de tráfico las cuales son definidas por el usuario. Estas clases de tráfico se corresponden con los tipos de tráfico con ancho de banda reservado durante la conexión como se explicó en la

sección 1. Adicionalmente, el software reserva el ancho de banda en el conjunto de VPCs que forman la ruta entre el origen y el destino.

Los aspectos más importantes que caracterizan el programa son, ejecuta funciones de control de admisión en base al ancho de banda demandado por cada tipo de tráfico, chequea el ancho de banda disponible para saber si puede o no satisfacer los requerimientos de ancho de banda, controla el ancho de banda disponible y ocupado en cada VPC, soporta todas las combinaciones de estrategias de asignación de ancho de banda que se muestran en, la tabla 1, mantiene un registro del número de llamadas rechazadas y aceptadas por clase de tráfico y por nodo, soporta la existencia de múltiples colas (una por cada clase de tráfico) en cada nodo de acceso, acepta la definición de múltiples niveles de prioridad hasta un límite de tres, soporta la definición de múltiples particiones de ancho de banda en cada VPC, las mismas pueden tener diferentes tamaños siempre y cuando la suma de estos tamaños sea menor al máximo ancho de banda disponible en el VPC, soporta muchas configuraciones de red las cuales pueden variar en el número de nodo y en la topología, y, por último, puede aceptar el establecimiento de otras distribuciones probabilísticas para el número de arribos de las llamadas y el tiempo de conexión de la misma.

III. Evaluación de los resultados

Existen muchas configuraciones de red cuyo rendimiento a nivel de las estrategias de asignación de ancho de banda estudiadas puede ser evaluado. Sin embargo, en esta sección se presenta un ejemplo para mostrar los resultados obtenidos usando el programa de computación descrito anteriormente.

III-A Parámetros de salida

Los parámetros de salida tomados en cuenta en la evaluación del rendimiento de las estrategias de asignación de ancho de banda fueron, el número total, de llamadas que llegan a la red, número de llamadas que entran al sistema (las llamadas que fueron encoladas), número de llamadas procesadas (aquellas llamadas que son tomadas de la cola de tráfico en el nodo de acceso y se les trato de asignar el ancho de banda requerido), el número total de llamadas rechazadas porque no había suficiente ancho de banda disponible sobre ninguna ruta, el número total de llamadas aceptadas, es decir, que se les asignó el ancho de banda requerido, el porcentaje de llamadas rechazadas y el porcentaje de llamadas aceptadas. Adicionalmente, se tiene que una clase de tráfico i se caracteriza por el tráfico ofrecido, $A_i = h/u$ medido en Erlangs, siendo la carga de tráfico definida como $S_i = A_i * PCR_i$. Así la carga total de tráfico S , en un nodo j se define como:

$$S_j = \sum_{i=1}^M S_{ij} \quad M = \text{número de clases de tráfico}$$

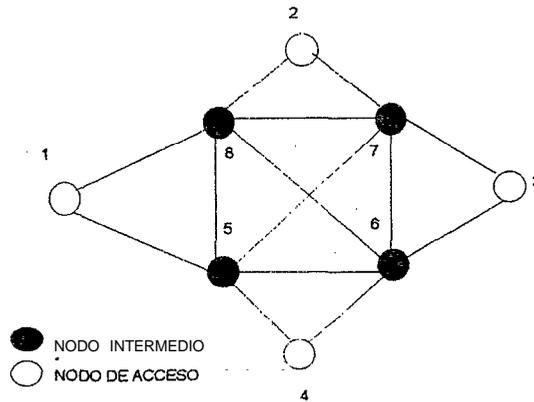


Fig. 1: Configuración de la red del caso de estudio.

y la carga de tráfico para un tipo de tráfico i en toda la red es:

$$S_i = \sum_{j=1}^M c_j S_{ij}, \quad M = \text{número de clases de tráfico}$$

Todos los parámetros son registrados por cada nodo y para la red completa.

III-B Caso de estudio

El siguiente caso de estudio demuestra como se puede aplicar el modelo de colas y el proceso de simulación descrito en la sección II y que resultados se pueden obtener usando el programa de computación creado para tal fin, para cada una de las combinaciones de estrategias de asignación de ancho de banda. La configuración de la red para el caso en cuestión se muestra en la fig. 1 y los valores de los parámetros de entrada en la tabla 2. El rendimiento es medido en función del porcentaje total de llamadas rechazadas y en función de la carga total de tráfico S_i , donde i representa la clase de tráfico.

Las figuras de la 2 a la 7 muestran los resultados obtenidos para cada una de las combinaciones de estrategias de distribución de la asignación del ancho de banda y de atención a las múltiples clases de tráfico.

Los resultados correspondientes al uso de la estrategia de asignación de ancho de banda, por compartición total (figuras 2,3,4) favorecen al tráfico con menor ancho de banda requerido, tráfico 0. Esta situación cambia un poco cuando se utiliza la estrategia de atención de tráfico cíclica y por prioridad, donde mejora el rendimiento del tráfico 1 y 2. Esto debido a que se le asigna una mayor o igual prioridad a estos tipos de tráfico para subsanar el hecho de que al tráfico de menor ancho de banda, por generar llamadas mas rápidamente, se le asigne mas ancho de banda del que queda disponible.

El uso de una estrategia de particiones, basada en la asignación de particiones tal como se muestra en la tabla 2, mejora el porcentaje de llamadas aceptadas en cargas bajas o medias (por ejemplo S = 3000 Mbps). Sin embargo, cuando la carga aumenta el porcentaje de llamadas rechazadas aumenta y llega a ser peor que en el caso de la estrategia de compartición

Topología de la red										
Longitud de cada troncal:		100 Kmts.								
Rutas										
Número de rutas:		2								
VPC										
Ancho de banda de cada VPC:		1500 Mbps								
Nro. de particiones por VPC:		1 0								
Características de Tráfico										
		Tiempo promedio de conexión (seg.)						PCR (en Mbps)		
Tráfico 0:		60						1		
Tráfico 1:		60						10		
Tráfico 2:		60						100		
Tráfico										
Prioridad Tráfico 0										
Prioridad Tráfico 1:										
Prioridad Tráfico 2:		0								
Tráfico aceptado por partición.										
	0	1	2	3	4	5	6	7	8	9
T0	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
T1										
T2										
T1										
T2										

Tabla 2: Parámetros de entrada para el caso de estudio.

total. Una explicación a este hecho es que en cargas altas el ancho de banda asignado por partición se queda corto para los requerimientos hechos por los tipos de tráfico 1 y 2.

De esto se puede concluir que las estrategias que establecen niveles de prioridad, como lo son la cíclica y la de prioridad, mejoran los valores de llamadas rechazadas para los tipos de tráfico 1 y 2 sin afectar el tráfico 0, que es el que tiene menor prioridad.

La estrategia de prioridad no disminuye considerablemente el porcentaje de llamadas rechazadas, pero una mejor asignación de las particiones por tipo de tráfico en combinación con la estrategia de prioridad si puede mejorar los valores correspondientes al porcentaje de llamadas aceptadas. Por ejemplo, se podría usar una combinación de las estrategias de compartición total y de partición, donde se le asigne mayor cantidad de particiones a los tipos de tráfico que tienen mayores requerimientos y las otras se compartan entre las restantes clases de tráfico, con la finalidad de no favorecer solamente las solicitudes de mayor cantidad de ancho de banda.

Porcentaje de llamadas rechazadas según la carga total de tráfico usando CT + FIFO

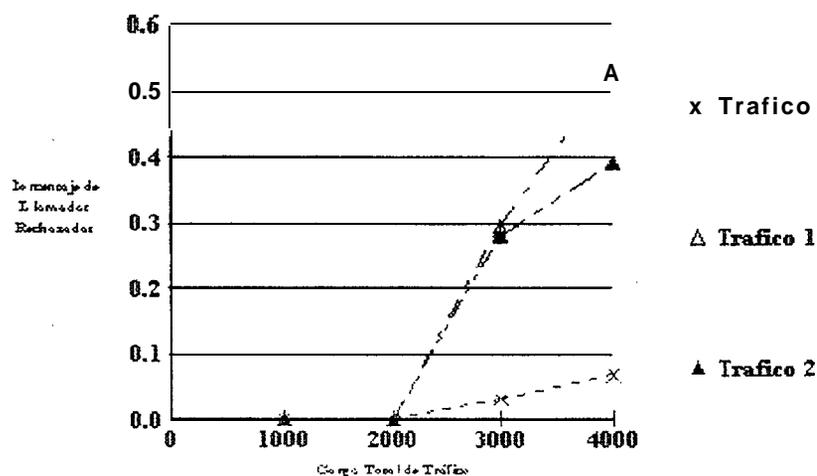


Fig. 2: Porcentaje de llamadas rechazadas en función de la carga total de tráfico usando la estrategia de compartición total y FIFO.

Porcentaje de llamadas rechazadas según la carga total de tráfico usando CT + CI

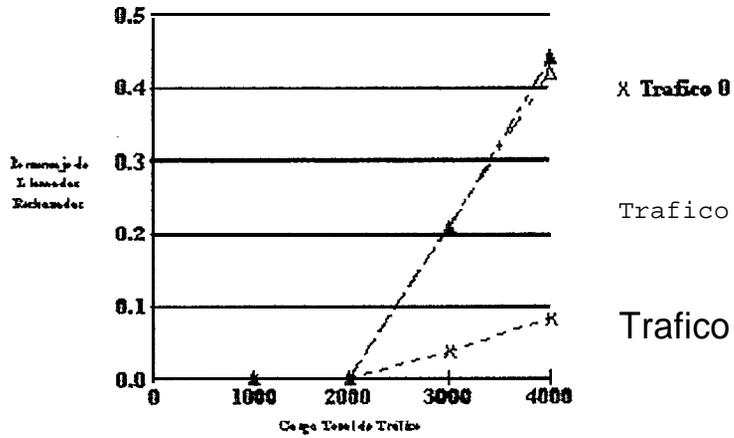


Fig. 3: *Porcentaje de llamadas rechazadas en función de la carga total/ de tráfico usando la estrategia de compartición total/ y cíclico.*

Porcentaje de llamadas rechazadas según la carga total de tráfico usando CT + PRI

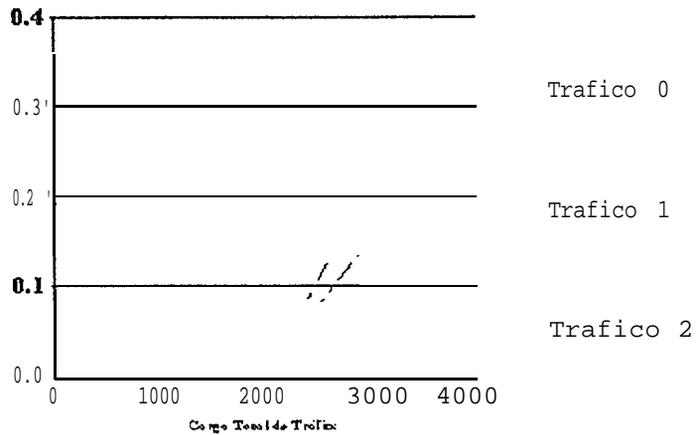


Fig. 4: *Porcentaje de llamadas rechazadas en función de la carga total de tráfico usando la estrategia de compartición total y prioridad*

Porcentaje de llamadas rechazadas según la carga total de tráfico usando PAR + FIFO

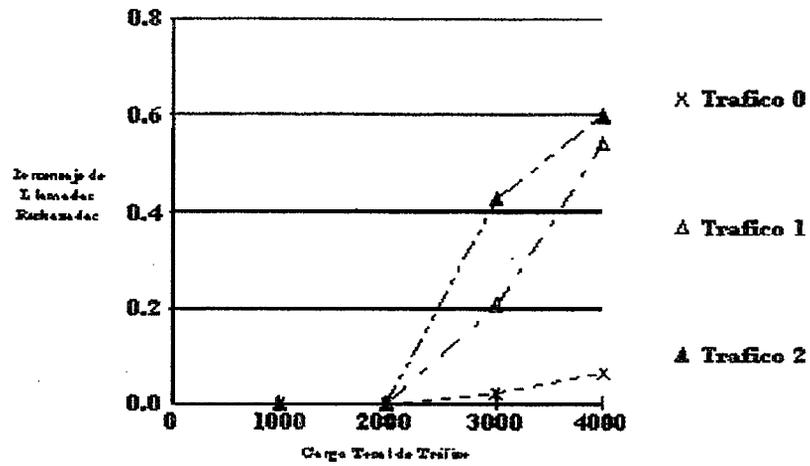


Fig. 5: Porcentaje de llamadas rechazadas en función de la carga total de tráfico usando la estrategia de particiones y FIFO.

Porcentaje de llamadas rechazadas según la carga total de tráfico usando PAR + CI

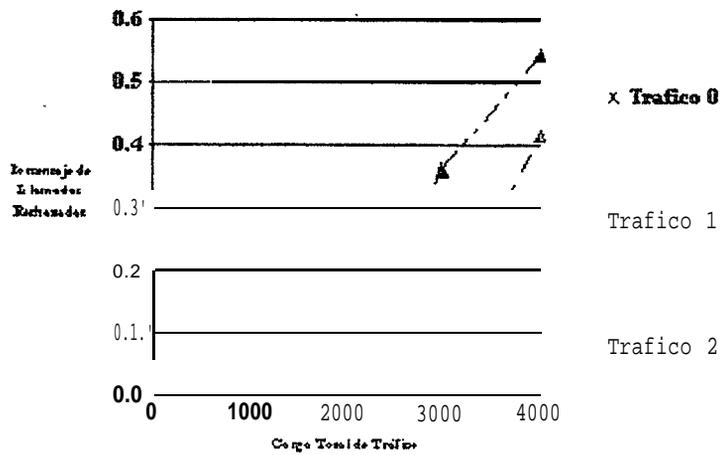


Fig. 6: Porcentaje de llamadas rechazadas en función de la carga total de tráfico usando la estrategia de particiones y cíclica.

Porcentaje de llamadas rechazadas según la carga total de tráfico usando PAR + PRI

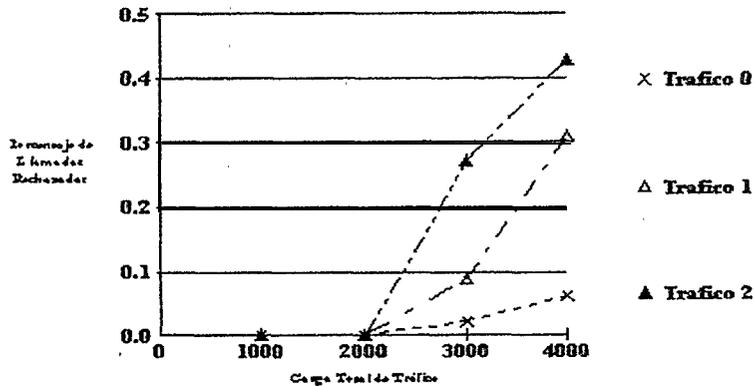


Fig. 7: Porcentaje de llamadas rechazadas en función de la carga total de tráfico usando la estrategia de particiones y por prioridad.

IV. Conclusiones

Las redes como lo es BISDN, pueden aceptar muchas clases de tráfico y que además usan sistemas de transmisión que soportan un gran ancho de banda deben contar con estrategias para realizar las funciones de manejo de los recursos de la red y de control de admisión para tratar de satisfacer la mayor cantidad de requerimientos de las llamadas, asociadas a tales tipos de tráfico, que tratan de entrar al sistema y para poder cumplir con los distintos contratos de tráfico establecidos. Estas funciones de control de admisión están destinadas a regular el acceso de las nuevas llamadas basadas no sólo en estos requerimientos de ancho de banda, sino en otros parámetros de tráfico de forma tal de cumplir con el contrato de servicio de las mismas durante su tiempo de conexión. Desde hace más de una década varios investigadores han estudiado el problema de la asignación del ancho de banda para redes integradas, vale citar [7], [16], [17], entre otros. Ellos han planteado diversas estrategias para resolverlo y han llevado a cabo ciertas evaluaciones de las mismas. Sin embargo, una característica especial que distingue esta investigación de las realizadas anteriormente, es que ella se basa en la comparación de estrategias de manejo de la distribución del ancho de banda combinadas con estrategias para el control de la atención del tráfico, las cuales habían sido estudiadas hasta este momento de manera aisladas, en trabajos, como el de [16] que se concentra en las estrategias de atención de las múltiples clases de tráfico solamente.

Por otra parte, en este trabajo se desarrolló un programa computacional el cual está basado en el modelo de encolamiento de servidor único por nodo, usa la simulación de eventos discretos, soporta múltiples clases de tráfico con diferentes características. soporta múltiples topologías. permite e!

establecimiento de múltiples rutas (basado en el enrutamiento alterno), soporta múltiples características de tráfico por tipo de tráfico y permite realizar pruebas y evaluaciones de todas las combinaciones de estrategias planteadas en este estudio, sobre conexiones virtuales, mas específicamente sobre VPCs (como lo recomienda el ITU [11]) , en base al porcentaje de llamadas rechazadas en función de la carga de trabajo por tráfico.

En el desarrollo del programa se probaron las diferentes combinaciones de estrategias de distribución de la asignación de ancho de banda y de atención de las múltiples clases de tráfico descritas en la sección II con el propósito de concluir acerca de cual de estos esquemas podría mejorar el rendimiento de la red en función del porcentaje de llamadas rechazadas, para una configuración de la misma y características de tráfico determinadas..

Respecto a la evaluación realizada, se pudo observar que es posible encontrar una combinación de estrategias de distribución de la asignación de ancho de banda y de atención de múltiples clases de tráfico, que pueda satisfacer los requerimientos de banda ancha y angosta. Adicionalmente, se puede concluir que el combinar estas estrategias ayuda a subsanar los problemas provenientes de usar una compartición total del ancho de banda por VPC o de usar esquemas no apropiados de establecimiento de particiones.

Referencias

- [1] A. ALLES. *ATM Network Design*. Cisco Systems, Inc. May 1994.
- [2] T. M. CHEN, S.S. LIU. *Management of ATM Networks*. IEEE Networks, July 1994.
- [3] L. A. CRUTCHER, A.G. WATERS. *Managing ATM Networks*. IEEE Network, November 1992.
- [4] A.E. ECKBERG. *ISDN Traffic and Congestion Control*. IEEE Network, September 1992.
- [5] A. GERSHT, K. J. LEE. *Virtual-Circuit Load Control in Fast Packet-Switched Broadband Networks*. GTE Laboratories.
- [6] GERSHT, K. LEE. *A Congestion Control Framework for ATM Networks*. IEEE Journal on Selected Areas in Communications, Vol 9, No 7, September 1991.
- [7] A. GERSHT, A. SHUIMAN, J. VUCETIC, J. KELISON. *Dynamic Bandwidth Allocation, Routing, and Access Control in ATM Networks*. GTE Laboratories. March 1994.
- [8] H. GILBERT, O. ABOUL-MAGD, V. PHUNG. *Developing a Cohesive Traffic Management Strategy for ATM Networks*. IEEE Communications Magazine, October 1991.
- [9] I.W. HABIB, T. SAADAWI. *Controlling Flow and Avoiding Congestion in Broadband Networks*. IEEE Communications Magazine. October 1991.
- [10] J. Y. HUI. *Resource Allocation for Broadband Networks*. IEEE Journal on Selected Areas in Communications, Vol 6, No 9, December 1988.
- [11] ITU-T. *Recomendación 1.371. Traffic Control and Congestion Control in B-ISDN*. Geneva. March 1994.
- [12] R. JAIN. *Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey*. Department Of Computer and Information Science, The Ohio State University. January 1995.
- [13] D. LIOU, A. UU. *A Model for Dynamic Channel Allocation in an ISDN*. Proceedings of the IEEE, Vol 79, No 2, Feb 1991.
- [14] A. PATTAVINA. *Multichannel Bandwidth Allocation in a Broadband Packet Switch*. IEEE Journal on Selected Areas in Communications, Vol 6, No 9, December 1988.
- [15] K-I SATO, S. OHTA, T. TOKIZAWA. *Broad-Band Network Architecture Based on Virtual Paths*. IEEE Transactions on Communications, vol 38, No 8, August 1990.
- [16] B. KRAIMECHE, M. SCHWARTZ. *Bandwidth Allocation Strategies in Wide-Band Integrated Networks*. IEEE Journal on Selected Areas in Communications, Vol SAC-4, No 6, September 1986.
- [17] E.D. SYKAS, K.M. VLAKOS, IS. VENIERS, E. N. PROTONOTARIOS. *Simulative Analysis of Optimal Resource Allocation and Routing in B-ISDNs*. IEEE Journal on Selected Areas in Communications, Vol 9, No 3, April 1991.
- [18] VILLAPOL M.E. *Asignación del Ancho de Banda en B-ISDN: una evaluación a nivel de conexiones virtuales*. Trabajo Especial de Grado presentado ante la Universidad Central de Venezuela para optar por el título de Magister en Ciencias de la Computación. Escuela de Computación. Caracas. Mayo 1996

SISTEMA DE ADQUISICION DE DATOS Y CONTROL PARA UN FRENÓMETRO DE RODILLOS DE VEHÍCULOS DE CARGA Y TRANSPORTE

Juan Carlos Ansaldi 1, 2 - Esteban Horacio López ²

1 Facultad Regional C. del Uruguay de la U.T.N.
2 Instituto Superior de Electrónica de Entre Ríos
Ing. Pereyra 676 (3260) Concepción del Uruguay - Entre Ríos
Tel: ++54+442-25541
Fax: ++54+442-23803
e-mail: ansaldij@frcu.utn.edu.ar

1 - INTRODUCCIÓN

A partir del año 1993, se comenzaron a realizar en la Argentina la verificación obligatoria de los vehículos de transporte de carga y pasajeros. Entre las verificaciones que se deben realizar, una que adquiere importancia relevante es la referente a la capacidad de frenado del vehículo, la cual se hace normalmente utilizando un “frenómetro de rodillos”.

Dentro de este contexto se vio la necesidad de desarrollar un equipo totalmente nacional que cumpliera con las especificaciones pedidas por la Comisión Ejecutiva Nacional de Transportes (CENT) y que pudiera competir con los equipos importados en calidad, prestaciones y costo, a pedido de un taller dedicado a realizar este tipo de verificación.

El diseño, cálculo y construcción del equipo mecánico, que se describe más adelante, estuvo a cargo del Ingeniero Electromecánico Omar Jones, egresado de la F.R.C. del Uruguay de la U.T.N.

A nosotros nos correspondió la tarea de automatizar dicho equipo, para lo cual se desarrollo un sistema que es el que presentamos en este trabajo y que debía ser capaz de cumplir con las siguientes características técnicas:

Registro de los siguientes parámetros:

- Pesos de los ejes
- Fuerza de frenado por rueda
- Resbalamiento por rueda
- Fuerza sobre el pedal
- Presión en el circuito de aire

Automatización completa del ensayo, lo que incluye:

- Determinación de la existencia de un eje de un vehículo sobre el frenómetro
- Puesta en marcha de los motores una vez verificada dicha existencia
- Desconexión de los motores por resbalamiento

Todos estos pasos los realiza para cada uno de los ejes que tiene el vehículo.

2- DESCRIPCIÓN DEL EQUIPO MECÁNICO A AUTOMATIZAR

El equipo mecánico a automatizar se compone de pares de rodillos de 280 mm de diámetro que giran con una velocidad tangencial aproximada a 2,5 Km/h, recubiertos de una superficie antideslizante, accionados por medio de una transmisión a cadenas y un motorreductor por cada rueda.

El motorreductor transmite la fuerza que se genera al frenar el vehículo a una celda de carga, la cual, luego de aplicadas las relaciones de transmisión y torque correspondientes, nos permite obtener la fuerza de frenado. Este conjunto se encuentra suspendido dentro de un bastidor metálico por medio de cuatro celdas de carga que nos dan el peso de la rueda.

Combinando los valores anteriores, fuerza de frenado y peso del eje, se calcula la desaceleración del eje.

En la Figura 1 que se encuentra a continuación, se detalla en forma esquemática el equipo a automatizar. Se debe tener en cuenta que el frenómetro está compuesto por dos de estos equipos, uno por rueda.

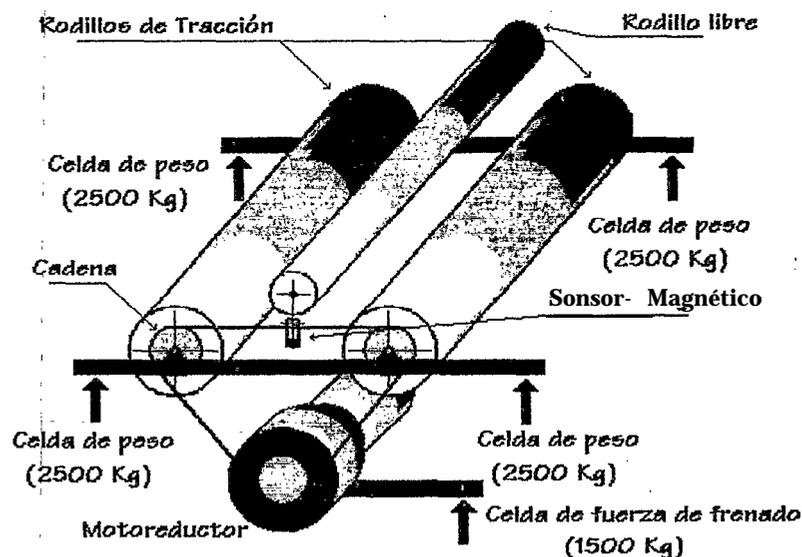


Fig. 1-Esquema del equipo mecánico

3- DESCRIPCIÓN DE LA ELECTRÓNICA DE ADQUISICIÓN Y CONTROL

Para la determinación del peso por eje del vehículo se utilizan cuatro celdas de carga de 2500 Kg. para cada una de las ruedas lo que permite tener un peso máximo de 10000 Kg. por rueda, 20000 Kg. por eje

Las fuerzas de frenado se registran también por medio de celdas de carga de 1500 Kg. cada una.

El resbalamiento de las ruedas se evalúa midiendo la velocidad de rotación de un rodillo que gira accionado por el rozamiento con la rueda del vehículo. La medición se realiza con sensores magnéticos que actúan sobre el rodillo y que desconectan los motores de accionamiento cuando la velocidad de rotación de los mismos disminuye en un porcentaje del valor nominal, el cual se encuentra parametrizado.

Los valores registrados por las celdas se someten una etapa de amplificación, para luego ser convertidos a valores digitales utilizando conversores A-D de 16 bit e ingresados a un microcontrolador 68hc11, el cual también es el encargado de disparar las conversiones.

Se utilizan las interrupciones disponibles en el microcontrolador para realizar las mediciones de la velocidad de giro de la rueda, por medio de los sensores magnéticos.

Además, se utilizan las salidas digitales para accionar, por medio de relés, el encendido y apagado de los motores.

El microcontrolador se encuentra conectado a la PC por medio de interfaz serie RS-232C transmitiendo los datos a una velocidad de 9600 bps. La Figura 2 representa esquemáticamente el diagrama de bloques del sistema electrónico de adquisición de datos y control.

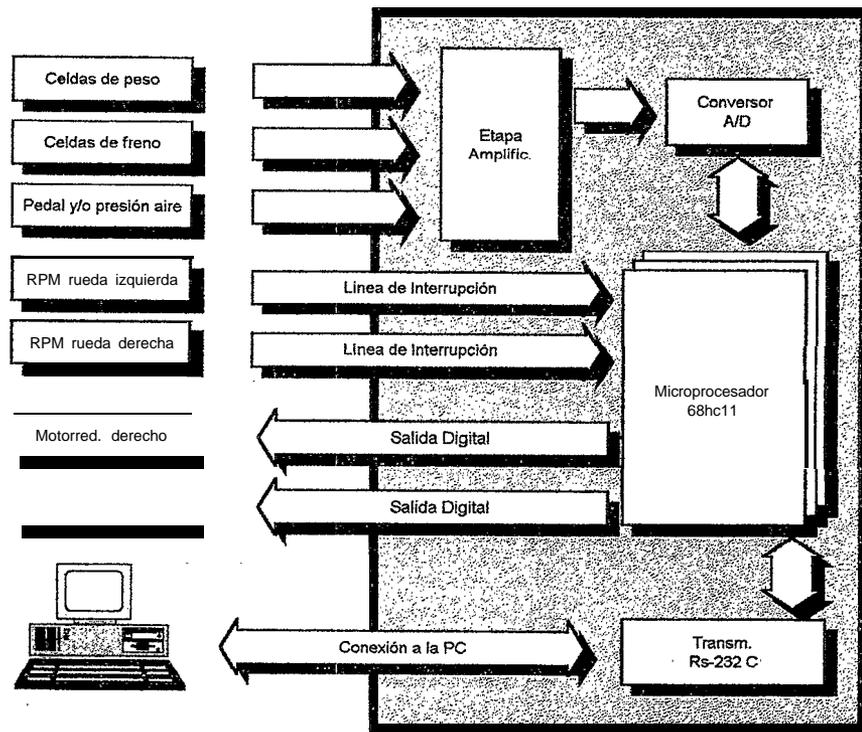


Fig. 2 - Diagrama del sistema electrónico

4- SOFTWARE DEL SISTEMA

El desarrollo del software del sistema está realizado utilizando programación 'visual orientado a objetos, bajo Windows (Microsoft Visual Basic + Microsoft Visual C++ y/o C++).

De esta manera se pretendió que la implementación del software fuese tal 'que su uso resulte sencillo e intuitivo, para lo cual optamos por este tipo de entorno de interfaz gráfica con el usuario (GUI), con ayudas en línea y con programación 'orientada a objetos que permita posteriores ampliaciones sin grandes modificaciones.

Se codificaron en Microsoft Visual Basic rutinas de representación gráfica de curvas, rutinas de representación numéricas, pantallas de captura de datos, listados, impresión, etc. Todas las rutinas se realizaron modularmente, de manera de poder realizar modificaciones en forma rápida, y buscando que sean reutilizables.

Además se codificaron rutinas para la adquisición de los parámetros y el control de los mismos en tiempo real.

A continuación se explican resumidamente los módulos que componen el software, prestándole mayor atención a aquellos que se destacan por su importancia dentro del sistema.

4.1- Ingreso de Datos y Configuración

Para esta etapa se diseñaron formularios de ingreso de datos y configuración de los parámetros del sistema y se le adjuntaron los procedimientos necesarios que no describiremos por razones de espacio. Ellos son:

Datos del Cliente	Grabar Archivo de Ensayo	Escalas
Datos del Taller	Backup	Salir
Leer Archivo de Ensayo	Configurar Parámetros	

4.2- Ensayo

Esta serie de módulos permite comenzar el ensayo de frenos, el cual se habilita luego de que se han ingresado los datos del cliente, y opcionalmente realizar el calibrado del equipo. Está compuesto por:

Comenzar Ensayo	'Balanza
Ensayar un Eje	Calibrar Frenómetro.
Descartar Ensayo	

Mientras se realiza el ensayo, se presenta en la pantalla de la computadora, por medio de barras, las fuerzas de frenado en ambas ruedas y la diferencia porcentual que existe entre ellas,

La Figura 3 corresponde al calibrado del frenómetro de rodillos. Este procedimiento resulta de importancia si consideramos que la automatización del ensayo se basa en la determinación, por peso, de la existencia del eje del vehículo sobre los rodillos, por lo que el peso de la estructura determinado durante la calibración, es el que se toma como "cero". Este peso, más una franja de seguridad que se ingresa al configurar los parámetros del sistema, son los que debe vencer el vehículo para que se habilite el arranque de los motores. De esta manera se evita, por razones de seguridad, el accionamiento con poco peso, normalmente 200 Kg. por rueda es el mínimo necesario.

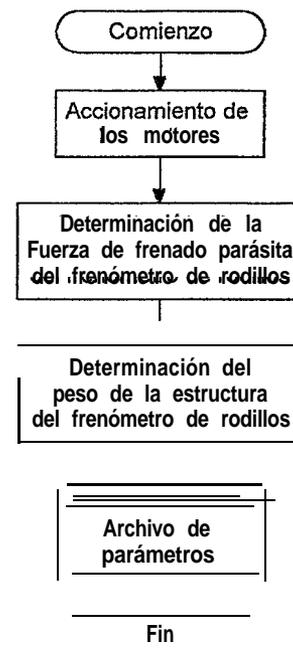
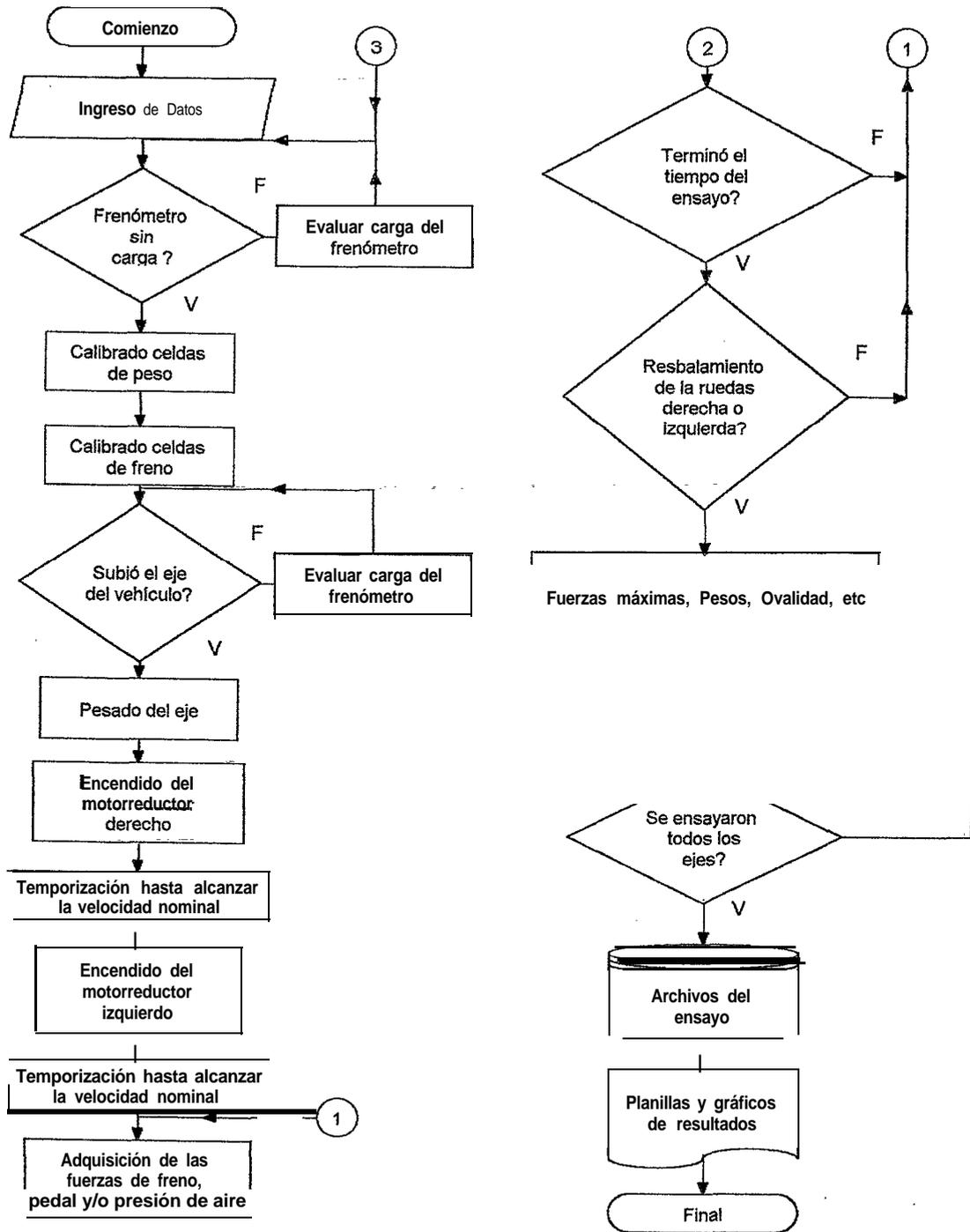


Fig. 3 - Diagrama de calibración

En la Figura 4 se observa el diagrama de flujo resumido del proceso principal del sistema, el ensayo de los ejes del vehículo.



2

Fig. 4 - Diagrama del ensayo

4.3- Resultados del Ensayo

Los resultados del ensayo se pueden observar por pantalla y también se puede generar un informe impreso de los mismos, que es el que sirve de control para las autoridades fiscalizadoras (CENT). Las opciones del mismo son: Resultados por eje y Resultados totales.

Estos módulos permiten observar tanto para los ejes de servicio como para los ejes de estacionamiento los siguientes datos:

Rozamiento al Rodar: También llamada Fuerza de Frenado Parásita o Resistencia de Rodadura. Valor que viene dado en Kg. Fuerza, y que corresponde al máximo obtenido durante el ensayo de cada rueda del eje en cuestión.

Fuerza de Frenado: Representa la máxima fuerza de frenado registrada para la rueda considerada.

Diferencia de Frenado entre Ruedas: Se determina relacionando la diferencia de fuerza de frenado entre las ruedas de un mismo eje dividido la fuerza mayor, expresado en porcentaje:

$$\text{Diferencia} = (F \text{ max} - F \text{ min}) / F \text{ max} * 100 \%$$

Ovalidad; Variación de la fuerza de frenado, manteniendo la presión constante para una única rueda, divido la fuerza máxima expresada en porcentaje.

$$\text{Ovalidad} = (F. \text{ Oval.max} - F. \text{ Oval. min}) / F. \text{ Oval.max} * 100 \%$$

Peso de la Rueda: Peso de cada una de las ruedas del eje.

Frenado de la Rueda: Porcentaje de frenado de la rueda respecto al peso de la rueda.

Desaceleración: Se determina el valor de la desaceleración por rueda:

$$a = (F. \text{ Frenado Rueda} * 9.81) / \text{Peso Rueda m/s}^2$$

En todos los datos anteriores se han discriminado los valores de acuerdo a:

Rueda Izquierda, derecha y diferencia total.

Peso Admisible: Peso total del vehículo con carga incluida.

Frenado del eje: Porcentaje de frenado del eje respecto del peso admisible.

En la Figura 5 se puede apreciar la ventana de resultados para el caso de un eje de servicio.

EJE DE SERVICIO	Izquierda	Derecha	Difer.	Total
Rozamiento al Rodar:	0909 Kg	0907 Kg	11,60 %	0014 Kg
Fuerza de Frenado:	0133 Kg	0134 Kg	00,75 %	0268 Kg
Ovalidad:				
Peso de la Rueda:	0319 Kg	0289 Kg	09,90 %	0608 Kg
Frenado de la Rueda:	41,83 %	46,72 %	00,75 %	44,15 %
Desaceleración:	04,10 m/s ²	04,88 m/s ²		04,33 m/s ²
Peso Admis. [Kg]:	1900			
Peso Total del Ens [Kg]:	1091			
Fren. del Eje [%] - Con P. Adm.:	17,45			Con P. Real: 25,48

Pedal [Kg]	Fuerza de Frenado [Kg]	Difer. [%]	Z			
Izq.	Der.	Total	[%]			
00,03	0096	0007	00014	10,10	02,25	
00,36	0013	0019	00028	12,52	04,76	
00,53	0001	0005	00006	73,20	01,00	
01,49	0010	0011	00021	15,71	03,45	

Fig. 5 - Resultados del Ensayo

Además, se presenta una tabla de valores de los datos obtenidos durante el ensayo, ordenados de mayor a menor en función de la fuerza aplicada al pedal de freno. Los resultados totales, se refieren a los datos anteriores totalizados para los ejes ensayados, divididos en ejes de servicio y de estacionamiento.

5- GRÁFICOS

Los gráficos son un elemento de suma importancia en este tipo de ensayos ya que permiten analizar el funcionamiento del sistema de frenos del vehículo de manera sencilla y rápida, por lo que nos extenderemos un poco más en su tratamiento. Los mismos se pueden ver en pantalla y si así se lo desea, se pueden imprimir. Los diferentes gráficos por eje se pueden tener en pantalla al mismo tiempo, y además se pueden maximizar o minimizar utilizando las opciones normales de Windows.

5.1- Gráfico de Fuerza de Frenado vs Tiempo

Este gráfico representa la fuerza de frenado (en Kg. Fuerza) en función del tiempo que dura el ensayo.

La representación se realiza para cada una de las ruedas del eje seleccionado y además también se grafica la fuerza ejercida sobre el pedal de freno.

La escala de fuerzas de frenado se representa en el eje de la izquierda, mientras que la escala de fuerza sobre el pedal en el eje de la derecha.

Luego de haber realizado un ensayo, se podrán representar los gráficos de cada uno de los ejes, para lo cual en la ventana del gráfico en cuestión aparece un botón **Próx. Eje** que nos permite ver las curvas de todos los ejes uno por uno.

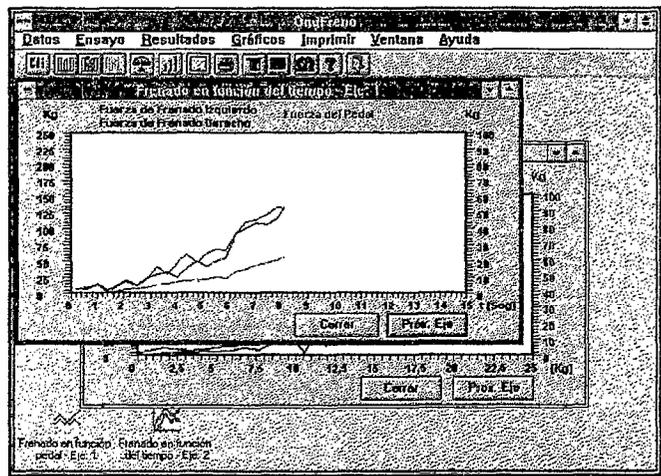


Fig. 6 - Frenado en función del tiempo

5.2 Gráfico de Fuerza de Frenado vs Fuerza en el Pedal

Este gráfico representa la fuerza de frenado (en Kg. Fuerza) en función de la fuerza ejercida sobre el pedal de freno (en Kg. Fuerza). También se representan las mismas funciones, pero en porcentaje

Los valores representados están ordenados en función de la fuerza ejercida sobre el pedal de menor a mayor. La representación se realiza para cada una de las ruedas del eje seleccionado y además también se grafica la fuerza ejercida sobre el pedal de freno,, por lo que para su visualización se debe activar la opción que permite sentir la fuerza en el pedal de freno (o la presión en el circuito de freno) al configurar los parámetros del sistema.

La escala de fuerzas de frenado se representa en el eje de la izquierda, mientras que la escala de fuerza sobre el pedal en el eje de la derecha (tener en cuenta que esta 'última será una recta).

5.3- Gráficos de Freno Total y Desaceleración

Este gráfico representa la fuerza de frenado total como porcentaje del peso del vehículo, en función de la fuerza ejercida sobre el pedal de freno (en Kg. Fuerza):

Como en el gráfico anterior, los valores representados están ordenados en función de la fuerza ejercida sobre el pedal de menor a mayor, por lo que para su visualización se debe activar la opción correspondiente al configurar los parámetros del sistema.

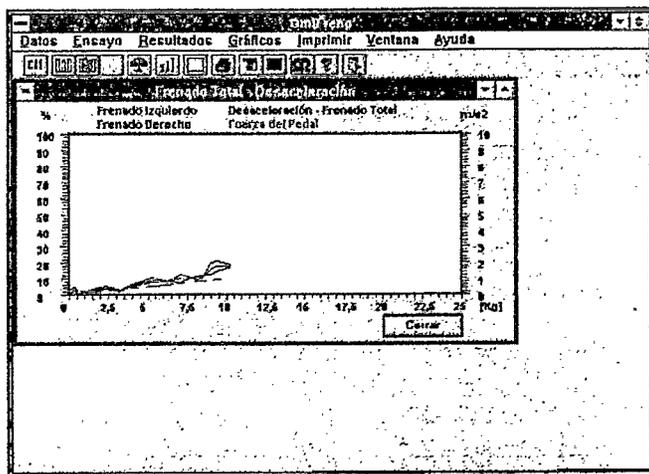


Fig. 7 - Frenado total y desaceleración

Además se representa la desaceleración total del vehículo, en m/s².

La escala de porcentajes de frenado se representa en el eje de la izquierda, mientras que la escala de desaceleración en el eje de la derecha.

6- ARCHIVOS DE DATOS

Como se vió en el diagrama de flujo del ensayo, por cada eje que se ensaya se generan archivos temporarios. Se optó por generar archivos temporarios, debido a que durante el ensayo es factible que se produzca cortes de energía u otros inconvenientes que interrumpen el ensayo, y de esta manera se evitaría la pérdida de los datos de los ejes ensayados previamente. Los archivos temporarios son:

Archivo de Datos del Cliente.

Archivo de Valores particulares del ensayo.

Archivo de Mediciones (un archivo por eje).

Archivo de Mediciones ordenadas en forma creciente (un archivo por eje) 'por el método Quick Sort.

A continuación se detalla la estructura de los archivos de datos utilizada por el sistema.

Datos del Cliente (Registro único)

Campo	Tipo	Descripción
Nombre	String * 40	Nombre del cliente
TipoVehiculo	String * 40	Tipo de Vehículo a ensayar (Semi, Ómnibus, Chasis, etc.)
MarcaVehiculo	String * 30	Marca del Vehículo
Patente	String * 10	Patente del Vehículo
Peso Admisible	Single	Peso del Vehículo con su Carga Máxima
NroEjesServicio	Integer	Cantidad de Ejes de Servicio del Vehículo a Ensayar
NroEjesEstacion	Integer	Cantidad de Ejes de Estacionamiento del Vehículo
EjesSinFreno	String * 3	Número de Ejes sin Sistema de Freno
Observaciones	String * 40	

Resultado del Ensayo (Un registro por eje ensayado)

Campo	Tipo	Descripción
RozalIzquierdo	Single	Fuerza de Frenado Parásita Rueda Izquierda
RozalDerecho	Single	Fuerza de Frenado Parásita Rueda Derecha
FrenoMaxIzq	Single	Fuerza de Frenado Máxima Rueda Izquierda
FrenoMaxDer	Single	Fuerza de Frenado Máxima Rueda Derecha
PesoRuedalIzq	Single	Peso de la Rueda Izquierda
PesoRuedaDer	Single	Peso de la Rueda Derecha
OvalidadIzq	Single	Ovalidad de la Rueda Izquierda
OvalidadDer	Single	Ovalidad de la Rueda Derecha
NumerodeMed	Integer	Número de Mediciones Efectuadas en el Eje

Resultados Totales (Registro único)

Campo	Tipo	Descripción
FrenoServicio	Single	Fuerza de Frenado Máxima de Servicio del Vehículo
FrenoEstacion	Single	Fuerza de Frenado Máxima de Estacionam. del Vehículo
PesoTotal	Single	Peso Total del Vehículo en el Ensayo
FrePesoEnServ	Single	Fuerza de Fren. de Servicio Referida al Peso del Ensayo
FrePesoEnsEst	Single	Fuerza de Fren. de Estacion. Referida al Peso del Ensayo
FrePesoAdmSer	Single	Fuerza de Fren. de Servicio Referida al Peso Admisible
FrePesoAdmEst	Single	Fuerza de Fren. de Estacionam. Referida al Peso Admisible
DesacelTotal	Single	Desaceleración con la Fuerza Máxima y Peso del Ensayo
DesacelAdm	Single	Desaceleración con la Fuerza Máxima y Peso Admisible

Mediciones y Mediciones Ordenadas

Campo	Tipo	Descripción
FrenolIzquierdo	Single	Fuerza de Frenado Rueda Izquierda
FrenoDerecho	Single	Fuerza de Frenado Rueda Derecha
Pedal	Single	Fuerza Ejercida en el Pedal
Tiempo	Single	Tiempo en que se efectuó la medición

Estos archivos se pueden agrupar en un único archivo en formato binario de resultados del ensayo, el cual se genera cuando se desea guardar un ensayo en particular. En este archivo binario, se guardan uno a continuación del otro los archivos temporarios resultantes del ensayo. La lectura y posterior recomposición de los archivos originales, está basada en el número de ejes y en la cantidad de mediciones por ejes, que son datos de los cuales se conoce previamente su posición dentro del archivo binario.

En la Figura 8 se puede observar la ventana que nos permite abrir un archivo de ensayo.

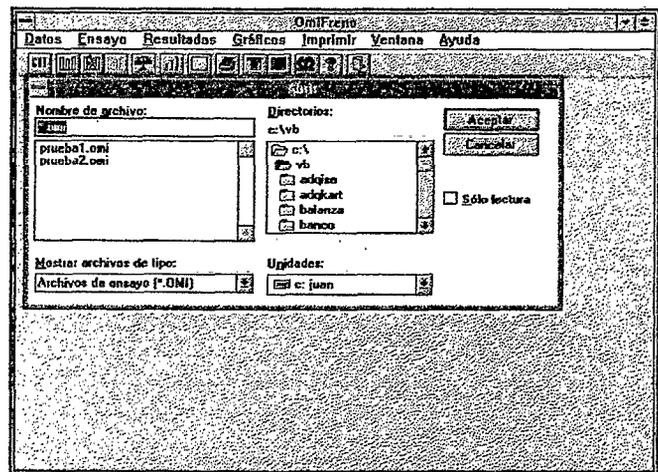


Fig. 8 - Ventana de archivos

7- CONCLUSIONES

El primer equipo que se desarrolló está funcionando desde finales del año 1994 y hasta la fecha no ha 'tenido mayores inconvenientes. Además de este equipo hemos instalados dos más en la provincia de Entre Ríos y uno en la provincia de Misiones que funcionan perfectamente desde el punto de vista electrónico y mecánico.

Se está estudiando la posibilidad de mejorar la presentación de los resultados en tiempo real por medio de 'indicadores controlados por motores por pasos y en un futuro integrar además a este sistema, un banco de comprobación de amortiguadores y un alineador al paso.

Al software original se lo fue actualizando y agregándoles opciones que fueron surgiendo con la utilización 'del sistema, sobre todo, la manera en que se debían presentar los datos.

El mayor problema se presentó y se continúa presentando al momento de competir con la financiación de los equipos importados, que ya tienen una experiencia reconocida en el área.

8- BIBLIOGRAFÍA

MC68hc11 Reference Manual
Motorola - 1989 :

CMOS Application - Specific Standard IC's
Motorola - 1991

Visual Basic for Windows Developer's Guide
D.F. Scott - SAMS Publishing - 1993

Visual'Basic Programmer's Guide to the Windows API
Daniel Appleman - ZD Press - 1993

Enciclopedia de Visual Basic
Feo. Javier Ceballos - RA-MA - 1994

Microsoft Visual Basic "Programming Guides, Language Reference & Users Guide"
Microsoft

Microsoft Visual C++ "Programming Guides, Language Reference & Users Guide"
Microsoft

Computadoras y Microprocesadores
A.C. Downton - Addison Wesley - 1993

El libro del RS232
Joe Campbell - Anaya Multimedia - 1988

MAXIM Evaluation Kit Data Book
Maxím - 1994

Sistema VLSI CAD para la Resolución de Circuitos Secuenciales.

El presente trabajo está orientado a la generación automática del layout de un microchip específico para sistemas secuenciales con aplicaciones adicionales en el desarrollo de un compilador de silicio.

Keywords: VLSI CAD, Sequential Synthesis, Combinational Synthesis, Silicon Compiler, Standard Cells.

Autores: Gabriel Belforti(belforti fi.mdp.edu.ar), Carlos Arturo Gayoso (cgayoso@fi.mdp.edu.ar), Claudio González (cmgonzal fi.mdp.edu.ar), Leonardo Arnone (leosrn@fi.mdp.edu.ar).

Universidad Nacional de Mar del Plata.
Facultad de Ingeniería.
Laboratorio de Componentes Electrónicos.
J. B. Justo 4302 - 7600 - Mar del Plata - Argentina
TE: (54 23) 81-6600 ext. 253 - FAX: (54 23) SI-0046

Gabriel Belforti. Cursa el último año de la carrera Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Realiza su proyecto final en el Laboratorio, de Componentes Electrónicos de la misma Universidad. Actualmente efectúa una pasantía rentada regular en la Dirección de Comunicaciones e Informática (DICEI) de la Organización Techint.

Ing. Carlos Arturo Gayoso. Profesor Adjunto de las asignaturas Electrónica e Introducción al Diseño de Circuitos Integrados de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Investigador del Laboratorio de Componentes Electrónicos de la misma Universidad.

Ing. Claudio Marcelo González. Profesor Adjunto de las asignaturas Circuitos Lineales, Dispositivos Electrónicos e Introducción al Diseño de Circuitos Integrados de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Investigador del Laboratorio de Componentes Electrónicos de la misma Universidad.

Ing. Leonardo José Arnone. Auxiliar Docente de las asignaturas Teoría de Circuitos II, Dispositivos Electrónicos e Introducción al Diseño de Circuitos Integrados de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Investigador del Laboratorio de Componentes Electrónicos de la misma Universidad

Abstract.

Se desarrolla una herramienta de CAD para la creación automática del layout de un microchip específico para sistemas secuenciales. Se tienen en cuenta diversas consideraciones para cumplir con todas las restricciones impuestas. Un compilador de silicio actúa de kernel para la herramienta en cuestión. El compilador está diseñado para operar en forma independiente en otros proyectos.

1. Introducción.

El presente trabajo está orientado al desarrollo de una herramienta de CAD para la *creación* automática del layout de un microchip específico para sistemas secuenciales. Entre las principales características figuran:

- Independencia del fabricante.
- Optimización en la síntesis de la lógica secuencial y combinacional.
- Minimización del área empleada.
- Enlace con otros softwares comerciales.
- Consideraciones de potencia y márgenes de ruido.

A través de un compilador de silicio se obtienen diseños no ligados a unas reglas geométricas en particular, por lo que se logra independizarse del fabricante. En particular la herramienta de CAD desarrollada controla las reglas de diseño y genera los archivos bajo formato CIF necesarios para el layout final del sistema deseado.

La optimización de la lógica secuencial y combinacional redundará no sólo en las consideraciones de potencia y área empleada, sino que también posibilitará simplificar el esquema de conexión y mayor rapidez en la planificación automática del layout final.

Es uno de los objetivos de este diseño la minimización del área empleada. En una producción en serie, a mayor área, mayores recursos monetarios empleados. Si se mantiene el área en un mínimo desde el origen del proyecto, se lograrán mejores resultados en el prototipo final.

La idea fundamental es organizar este sistema en bloques fácilmente intercambiables. ES decir, si el usuario final decide por algún motivo cambiar el software de minimización por otro *mejor o* más completo, simplemente adapta la salida de su software al formato de entrada del layout designer y ajusta este módulo para que anule la opción de minimización y todo el sistema se adaptará al nuevo software. También, y en una segunda etapa, se adaptará a los estándares de VHDL para una mayor compatibilidad con el software comercial existente.

Tanto la potencia como los márgenes de ruido deben permanecer dentro de límites aceptables para los estándares. Esto debe ser analizado con sumo cuidado para que el diseño sea físicamente realizable y sea confiable bajo condiciones adversas de funcionamiento.

En definitiva, existen muchos paquetes integrados que realizan las tareas de síntesis, análisis y diseño de circuitos secuenciales. La idea de este sistema es proveer a los diseñadores de hardware de una herramienta asequible, fácil de usar y modificar y, por sobre todas las cosas, independiente de las reglas de diseño de cada fabricante.

2. Desarrollo.,

Uno de los principales problemas involucrados en este proyecto era la flexibilidad. Luego de analizar varias ideas, se llegó a la conclusión de que el uso de la tecnología de compiladores iba a proveer al sistema de un potencial importante.

Por ello, el sistema fue dividido en cuatro partes:

- Generador de Grupos.
- Ubicador.
- Ruteador.
- Sistema de resolución de circuitos combinacionales y Secuenciales.

De esta manera, se obtuvo un compilador independiente del proyecto en sí, capaz de Ser utilizado en otros desarrollos. Esta posibilidad lo dota de una flexibilidad mayor de la esperada.

2.1. Tecnología de Bloques y Grupos.

Como es bien sabido, muchas veces los tiempos de diseño y los costos hacen que el sistema a implementar no sea optimizado respecto a la cantidad de transistores, es decir área y potencia, e incluso respecto a la frecuencia del reloj que el sistema debe operar. Es decir se recurre a diseños semi custom.

Para estos casos, lo más aconsejable es usar un paquete de software que automatice la mayor cantidad de procesos posibles y que focalice el problema del diseñador en las simulaciones y pruebas del circuito pre y pos layout. Más aún, dotar a un paquete de software de herramientas fáciles de usar y simples son puntos básicos a tener en cuenta en el análisis y diseño del CAD.

El proyecto toma estas premisas y las lleva a la práctica. El uso del CAD desarrollado nos dio experiencia respecto a esos puntos, tan importantes como la tecnología VLSI en sí misma.

Al hablar de tecnología de bloques y grupos nos estamos refiriendo precisamente al USO de un sistema en el que se pueda automatizar fácilmente la creación del layout. Como el proyecto tiene como base una PAL para resolver el circuito combinacional y un conjunto de flip-flops para la solución secuencial, la necesidad de repetibilidad es indiscutible.

Esta técnica combinada con el uso de algoritmos de compilación, el paradigma de objetos y un uso preciso de las reglas de diseño nos posibilitaron lograr un desarrollo independiente de las reglas del fabricante y al mismo tiempo optimizar el área, la potencia, los márgenes de ruido e incluso la máxima frecuencia de funcionamiento.

Cada bloque se construye directamente con las reglas de diseño del fabricante y otras variables agregadas a los efectos de poder influir de forma externa -imponiendo restricciones- en el armado del bloque de diseño. Es responsabilidad del diseñador construir estos bloques de forma tal de cumplir correctamente con las reglas de diseño. Es muy importante destacar que al tener que definir uno mismo el conjunto de reglas a utilizar de antemano, la complejidad o simpleza para la resolución del proyecto no depende del compilador en sí. Este es un punto sobresaliente respecto a otros compiladores existentes.

Una vez construidos los bloques, se debe definir la planilla donde van a ser ubicados. El conjunto de bloques que conforman la planilla se denomina grupo. Una *celda* se define como la mínima parte vacía de una planilla. Una celda ocupada es un bloque. Una característica distintiva del sistema desarrollado consiste en tener la posibilidad de imponer condiciones sobre las celdas que componen la planilla. Por ejemplo, se puede condicionar una celda de la planilla para que inserte un bloque u otro dependiendo de ciertas restricciones.

El tener la posibilidad de construir cuantos grupos queramos nos permite separar la próxima etapa de nuestro desarrollo y emplear otros paquetes de software mediante una interface. De esta forma, podemos construir otros grupos con otros CADs y luego unir todos los grupos con el ubicador y el ruteador.

El ubicador nos permite decidir la posición de cada grupo. Se debe tener en cuenta que un mal posicionamiento de los grupos va a conducir a que el ruteador emplee más espacio para poder cumplir con las condiciones de conexión.

Mediante el tuteador podemos unir los distintos grupos y así conformar el circuito final deseado. Si se desea usar otros paquetes de software simplemente se debe escribir un archivo conteniendo una serie de datos indispensables para que el ruteador realice su tarea,

De esta forma se resuelve el problema de la obtención de un archivo CIF con el circuito secuencial deseado y cumpliendo no solamente con las reglas de diseño del fabricante sino también con diversas restricciones propias de la tecnología CMOS involucrada. La Fig. 1 representa la primera parte del sistema.

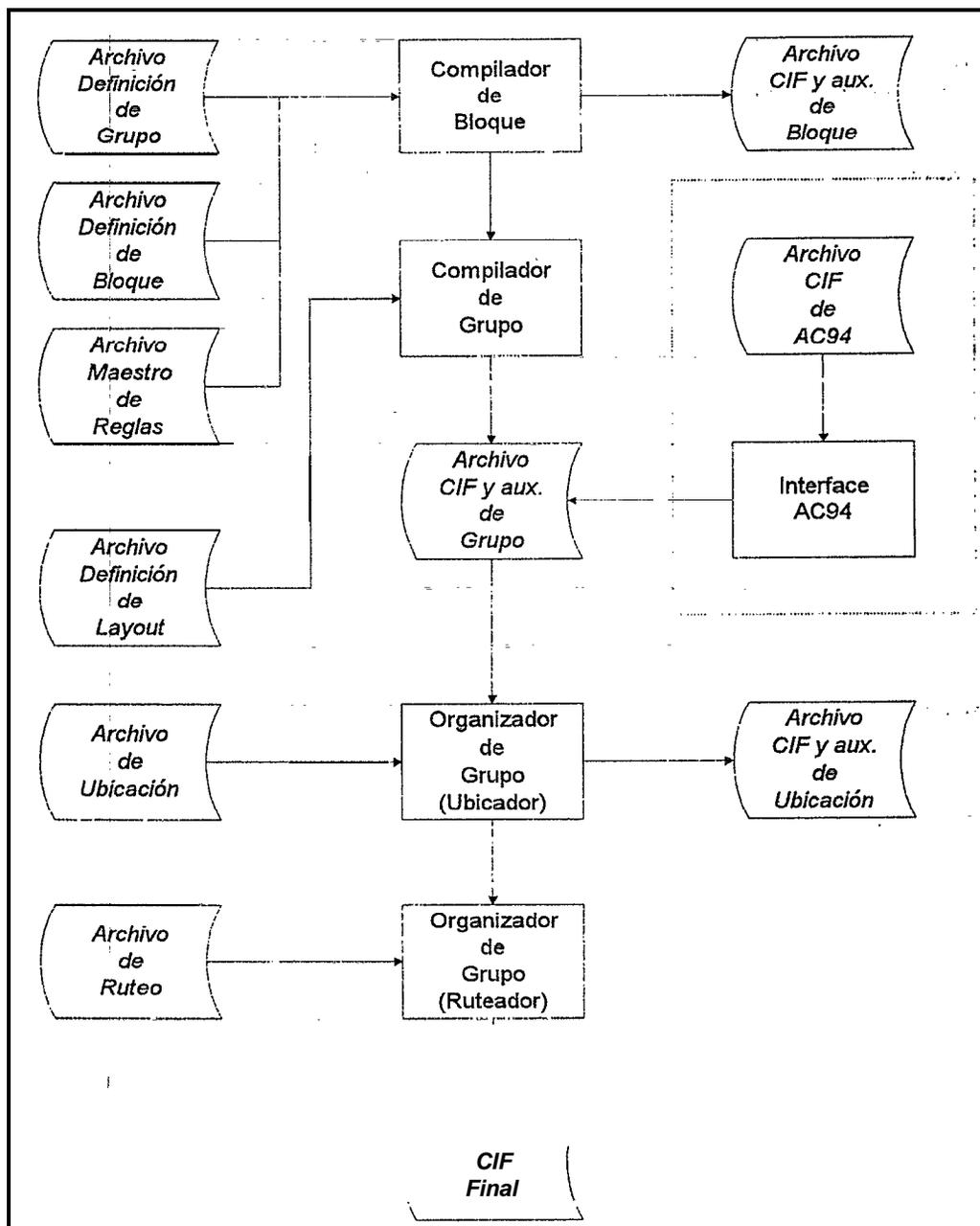


Fig 1. Diagrama del VLSI CAD genérico.

2.2. Generador de Grupos.

Una de las tareas más importantes dentro de este proyecto es precisamente la construcción de los bloques y grupos. Empleando la tecnología anteriormente descrita y aprovechando las técnicas de desarrollo de compiladores, se creó un sistema integrado que permite no sólo ampliar el conjunto de instrucciones del compilador, sino también, en una forma rápida generar el archivo CIF y de perfil de cada bloque y cada grupo.

Una ventaja de este sistema es la forma de organización de la información. Dado que no se cuenta con una interface gráfica propiamente dicha, el entorno de trabajo se cuidó al extremo de

automatizar y organizar perfectamente cada paso. De esta forma se logra trabajar con un lenguaje escrito -con todas las desventajas que poseen- y a su vez ahorrar tiempo.

A continuación se describirán los módulos fundamentales del proyecto. En ellos se observará la fusión de diversas tecnologías de software involucradas.

2.3. Compiladores.

Un compilador es un programa o grupo de programas que traducen un lenguaje en otro. Los compiladores tradicionales se separan en diferentes secciones denominadas “pasadas”, que se intercomunican a través de archivos temporarios. La estructura típica de un compilador consiste en cuatro pasadas.

La primera, pasada es el preprocesador. Normalmente el preprocesador hace la sustitución por macros, anula los comentarios y realiza otra serie de tareas secundarias. La segunda pasada es el corazón del compilador. Se compone del analizador de léxico, el parser y el generador de código. Se encarga de traducir el código fuente en un lenguaje intermedio más parecido al lenguaje final. La tercera pasada es el optimizador, el cual mejora la calidad del código intermedio generado, y la Cuarta pasada es el back end, el cual traduce el código optimizado en el lenguaje final. Por supuesto, existen muchas variaciones sobre esta estructura. Nuestro desarrollo se basa en un compilador de dos pasadas.

2.3. 1. Compilador de Bloques.

El punto de partida es la generación de los bloques. Para ello, un potente editor de bloques ayuda en forma interactiva al desarrollador. Para cada grupo se define un “mundo” o *proyecto* en donde se incluyen las reglas de diseño del fabricante seleccionadas, las reglas de restricciones y variables generales que se emplearán más adelante.

Una vez definida la parte general del “mundo” del grupo, se crea una estructura de árbol de un nivel o rama en donde cada nodo posee:

- reglas macro, conformadas por la combinación de diversas reglas de diseño. Se utilizan para simplificar los parámetros en el diseño del bloque.
- sub-bloques, para crear bloques en forma más rápida.
- bloques, son la parte fundamental de la estructura.

Es posible crear varios nodos desde la raíz, pero no se pueden crear nodos que dependan de otros nodos. Es decir, se acepta un solo nivel.

Se procede a continuación a generar un bloque. Se definen todas las capas del layout en el lenguaje propio del compilador de bloques y se indican los puertos de entrada/salida para su futura extensión. Esto será explicado en la sección siguiente.

Una característica importante del compilador es que permite que el conjunto de instrucciones disponibles para la definición de los bloques sea ampliado.

Para agregar sub-bloques simplemente se indica con una instrucción y se define la posición que va a tomar dentro del bloque.

Las variables generales y las reglas de diseño son empleadas en todos los nodos. Cada nodo puede generar nuevas reglas macro pero estas reglas son propias del nodo y no pueden ser compartidas con otros nodos.

2.3.2. Compilador de Grupo.

El Editor de Grupo es el encargado de definir la planilla. Se debe definir un bloque o conjunto de bloques para cada celda. En este último caso, se debe agregar una condición para que el Compilador defina que bloque va a insertar en la celda. Para la resolución de estas condiciones es necesario emplear las variables generales definidas en el mundo del grupo.

El compilador no permite que se superpongan bloques en las celdas. El Editor ‘es ‘el responsable de traducir la información gráfica a un archivo de definición de grupo.

Definida la planilla, el Compilador de Grupo verificará que todos los bloques a Utilizar *estén* previamente compilados. Caso contrario, llamará al Compilador de Bloques para que efectúe la tarea.

La siguiente tarea consiste en alinear todos los bloques. Dado que los bloques *están* definidos en forma paramétrica, no es posible saber de antemano las dimensiones de los mismos. Si bien el grupo aparece como algo totalmente estructurado, en realidad los bloques tienen dimensiones totalmente disímiles. Por lo tanto, el compilador efectuará lo que se denomina: extensión de bloques.

Cuando se construyeron los bloques, se definieron puertos de salida/entrada. Estos puertos son los que permiten que el bloque se conecte con sus bloques adyacentes.

Para que todos los bloques de una fila o columna tengan la misma extensión es necesario extender (Fig. 2) cada uno de los bloques hasta que iguale al mayor de todos los bloques en la fila y columna a la que pertenece. Para que la interconexión entre bloques tenga coherencia, se deben definir los puertos que se desean interconectar con el mismo nombre y el compilador se encargará de que estos puertos se unan luego de la extensión.

Una vez que todos los bloques que correspondan se hayan extendido, el compilador generará un archivo CIF con todo el grupo y un archivo de perfil donde guardara información indispensable para las próximas etapas.

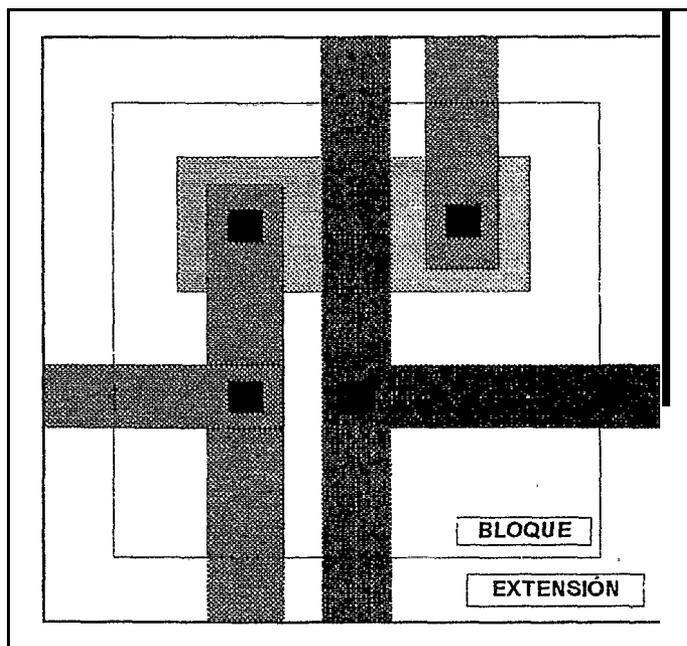


Fig. 2: Proceso de extensión de bloque según los puertos definidos.

2.4. Ubicador.

Antes de proceder a rutear todos los grupos que conforman el proyecto, es necesario plantear la forma final que tendrá el chip.

El Ubicador nos suministra un área gráfica en donde podremos insertar en forma esquemática los bloques ya compilados.

Nos es necesario dejar espacio para los canales de conexión ya que los mismos serán agregados en forma automática por el Ruteador, encargado de la tarea final.

2.5. Ruteador.

Una vez compilados todos los grupos que formarán parte del proyecto, la siguiente tarea consiste en el ruteo de todas las entradas y salidas de cada grupo para formar el layout final que conformará el sistema deseado.

Si bien existen diversas técnicas de ruteo, en el caso de nuestro desarrollo hemos preferido utilizar las técnicas más tradicionales y dejar abierto el camino para la implementación de otras técnicas más específicas. Recordemos que al ser un sistema abierto y dada su estructura modular es posible adaptar cualquier algoritmo teniendo en consideración los formatos de entrada y salida para cada caso.

Además, esta planteada la posibilidad de adaptar otros tipos de desarrollos, tales como los diseños realizados en AC (sistema CAD para el diseño de circuitos integrados semi dedicados, desarrollado en el laboratorio).

Para la implementación del tuteurador se ha utilizado el algoritmo de interconexión de canal descrito en el libro 'Microelectrónica', Capítulo 3, Sección 3.10.2. Dados dos conjuntos enfrentados de puntos de acceso a unir, se definen pistas de interconexión que se resuelven mediante búsquedas en un grafo equivalente. Para la interconexión se necesitan solamente dos niveles, por ejemplo, metal 1 y metal 2, uno para las líneas verticales y otros para las horizontales.

Cuando la conexión no sea enfrentada, una serie de líneas (buses) ubicadas entre los grupos, realizarán la tarea de transportar el punto de acceso hacia el destino final predeterminado.

Finalmente, un grupo especial de entrada/salida nos brindará la posibilidad de obtener un grillado a la entrada/salida.

2.6. Sistema de Resolución de Circuitos Secuenciales.

El predefinir los bloques y grupos de antemano es una de las principales ventajas del sistema (Fig. 3). Por lo tanto, las únicas tareas a llevar a cabo son:

- * la realización de la síntesis combinatorial y secuencial;
- * la generación de los bloques de PAL y Flip-Flops necesarios para completar el diseño deseado, respetando las restricciones impuestas;
- 0 la generación de la ubicación y el ruteo de los bloques.

2.6.1. Síntesis Secuencial.

El punto de partida de todo análisis secuencial es la construcción de un conjunto de estados que identifiquen al problema a resolver. Existen diversos procedimientos que no serán discutidos.

Una vez encontrado el conjunto de estados, se deben traducir a la tabla de estados. Dado que el número de flip-flops en un circuito aumenta a medida que aumentan los estados, es fundamental mantener la cantidad de estados al mínimo para así ahorrar área en el layout final. Para minimizar la cantidad de estados se empleó el método de simplificación por Tablas de Implicación.

Terminado el proceso de análisis, se debe ingresar el diagrama de estado por medio de la tabla de estados o una descripción escrita de los estados. Para este último punto, se analizó con éxito la posibilidad de implementar un subconjunto de instrucciones del Lenguaje de Descripción de Hardware.

Por el momento, el sistema no posee la capacidad de discernir un sistema secuencial de memoria finita. Por lo tanto, dichos circuitos secuenciales serán tratados como sistemas secuenciales generales.

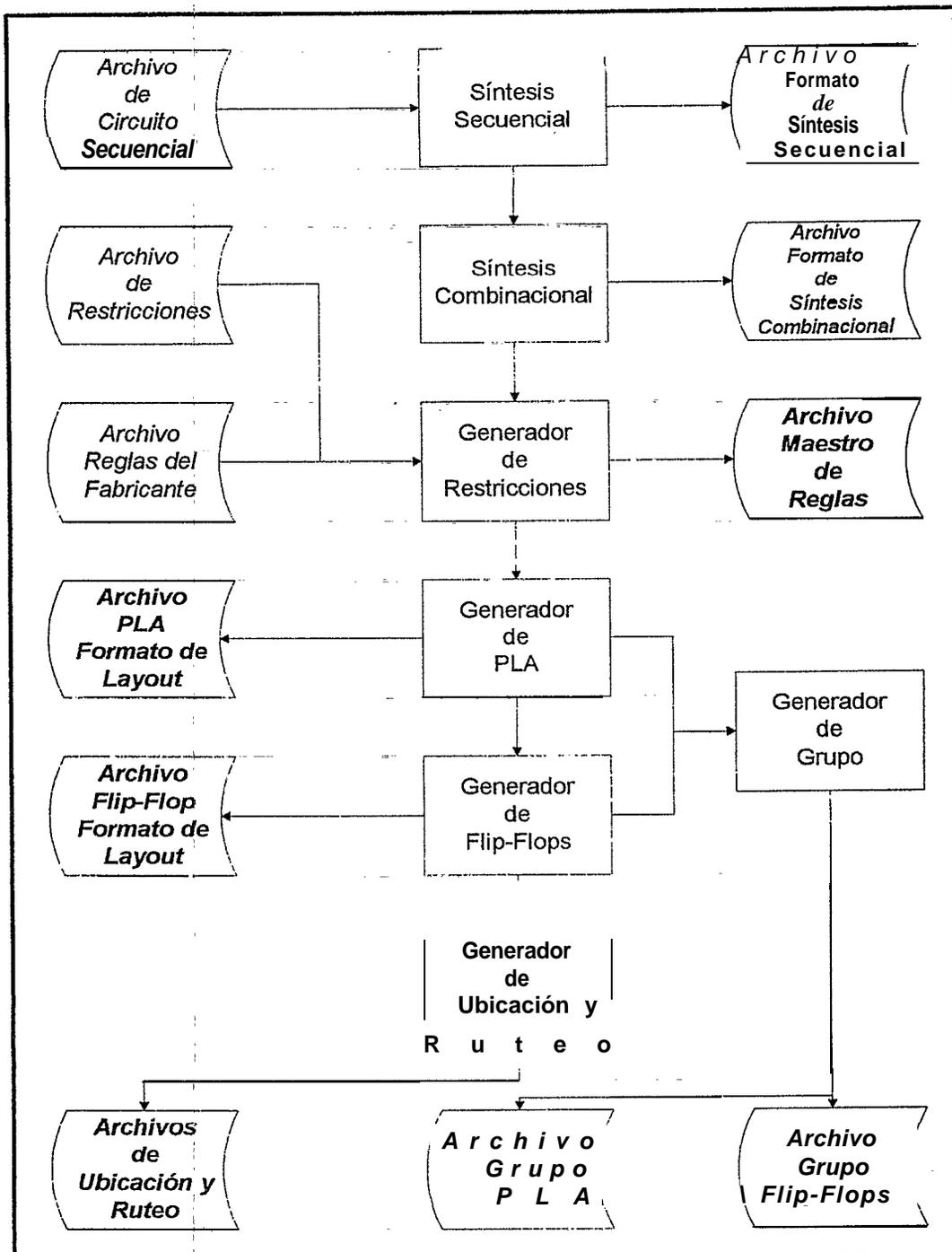


Fig. 3: Sistema CAD VLSI para la Resolución de Circuitos Secuenciales.

Para todo el desarrollo se adoptó el modelo Moore en el cual la salida es una función del estado presente. Se debe tener en cuenta que el modelo Moore puede ser convertido al modelo Mealy sin mayores inconvenientes.

Una vez efectuada la simplificación de los estados, resta diseñar el Circuito que lleve a hardware el diseño. Entonces, falta toda la lógica combinacional que implemente las salidas y las ecuaciones de entrada a los circuitos secuenciales (flip-flops) que conforman el sistema.

Puede ocurrir que alguna tabla de asignación distinta a la que obtuvimos minimizando nos conduzca a una lógica combinacional más simple; sin embargo, se optó por priorizar la minimización de la lógica secuencial.

La asignación de estados consiste en imponer una combinación específica de valores de las variables de estado a cada estado. Este tipo de asignación es muy simple en los circuitos Moore en donde cada estado corresponde a una salida singular. También es muy simple en el modelo de memoria finita.

Determinar la mejor asignación de estados no es un problema fácil de resolver. Por ejemplo, la cantidad de asignaciones posibles para 9 estados y 4 variables de estado es de 10,810,800. Verdaderamente imposibles de evaluar!

Una alternativa a la asignación de estados que se utiliza a veces para la 'realización de unidades de control es el one-hot. En esta técnica, el número de elementos de memoria es igual al número de variables de estado, y solamente una memoria asumirá el valor 1 en cada estado.

Lamentablemente no existe aún un método formal para la asignación de estados. De acuerdo con Hill y Peterson todos los programas dedicados a la implementación de circuitos secuenciales desde la tabla de estados utilizan métodos empíricos desarrollados normalmente en sistemas expertos.

En nuestro sistema la asignación de estados no emplea ningún algoritmo especial pero efectúa vanas asignaciones y elige la mejor de todas. La cantidad de asignaciones puede ser predefinida o puede utilizarse una ventana de tiempo para la evaluación.

Como resultado de la asignación de estado, se genera un conjunto de ecuaciones lógicas correspondientes a las salidas y los bloques de memoria es generado.

2.6.2. Síntesis Combinacional.

El conjunto de ecuaciones lógicas obtenidas para las salidas y los bloques de memoria mediante la síntesis secuencial debe ser minimizado para lograr el área mínima. Para ello, el sistema posee un módulo básico con el algoritmo de Quine-McCluskey. Este algoritmo puede implementar Circuitos con múltiples salidas para mejorar las consideraciones de área. A su vez, no se descarta el uso de otros sistemas tales como el Espresso. Para ello, existen dos formas de integrar nuevos métodos de minimización al sistema. El primero consiste en un programa que modifique los datos de salida del sistema usado para adaptarlo a nuestro sistema. El segundo consiste en diseñar una rutina para que efectúe la minimización y agregarla a las opciones de métodos de minimización que ofrece el sistema.

Para los casos en que los tiempos de proceso sean muy largos, se prevén puntos de corte de acuerdo a parámetros tales como: mínima cantidad de cubos para todo el sistema, mínima cantidad de cubos por función: ventana de tiempo.

Este módulo brinda información adicional respecto a la cantidad de entradas y salidas empleadas, cubos generados y otra serie de datos adicionales necesarios para la próxima etapa de diseño: el Generador de PLA.

2.6.3. Consideraciones Generales.

Las consideraciones generales para los bloques son definidas a través de variables que afectan al compilador en el momento de la creación del grupo. Entre las consideraciones a tener en cuenta para la generación del circuito secuencial figuran:

- Consideraciones de área: por medio de la síntesis lógica se obtiene una notable reducción de la PLA y de la cantidad de flip flop's a utilizarse. Esto redundará en la optimización de área. Además, la parametrización de los bloques ajustándose a las reglas de los fabricantes permiten lograr diseños de tamaño más reducidos.
- Consideraciones de potencia y velocidad: existen dos componentes de potencia asociadas al circuito. La potencia estática debida a la corriente que circula por el p-MOS de pull-up cuando la salida es nula. La potencia dinámica debida a la transición de 0 a 1 o viceversa. Esto resulta en un pulso de corriente de Vdd a Vss. También es necesaria corriente para cargar y descargar los

capacitores de carga de salida. Ambos términos son necesarios para calcular la potencia y dependiendo de la carga uno prevalecerá sobre el otro. La velocidad es un factor también decisivo en la potencia ya que la comente de carga y descarga depende, entre otras cosas, de los tiempos de subida y bajada. Existe una relación proporcional entre la máxima frecuencia de trabajo y la potencia disipada.

- Consideraciones de margen de ruido: existe un compromiso entre margen de ruido y velocidad. Se evaluaron con simulaciones SPICE diversas configuraciones para mejorar los márgenes de ruido en los bloques de la PLA.

2.6.4. Generador de PLA.

Para la generación de la PLA se utilizó lógica pseudo-nMOS en una configuración del tipo NOR-NOR. Entre las principales ventajas figuran la gran velocidad, el gran fan-in de las compuestas NOR y la minimización de la cantidad de transistores empleados. Una desventaja de esta lógica es el consumo de potencia.

Diversos tipos de PLA predefinidos son considerados de acuerdo a las restricciones impuestas con la cantidad de entradas y la cantidad de minitérminos a generar. La cantidad de entradas va a afectar a los caminos de metalización que transportan a Vdd y Vss. La cantidad de minitérminos afectará a la geometría de los p-MOS de pull-up y en caso de ser necesario se agregarán etapas en cascadas para poder manejar mayor capacidad de carga.

2.65 Generador de Flip-Flops.

Se generaron tres grupos que implementan los siguientes flip-flops: D, JK y T. La cantidad y tipo de flip-flops a emplear es conocida una vez efectuada la síntesis secuencial. Por lo tanto, es tarea del Generador de Flip-Flops efectuar la compilación de los grupos necesarios que Conformarán el conjunto de flip-flops a utilizar en el diseño e informar al generador de Ubicación y Ruteo sobre la forma de conexión de los mismos.

2.6.6. Generador de Ubicación y Ruteo.

Una vez desarrollada la PAL y los Flip-Flops, la siguiente tarea consiste en .describir la posición de cada grupo y sus conexiones para que el ruteador se encargue de unir todos los grupos.-.

Al igual que los bloques, la ubicación de cada grupo es predefinida. La única tarea a realizar por este generador consiste en crear un archivo con los valores de largo y ancho obtenidos para cada grupo y con toda la distribución ya definida.

El generador toma los archivos de perfil creados por los generadores de PAL y flip flops y traduce esta información para que pueda ser interpretada posteriormente por el Ruteador. Recordemos que la tarea fundamental está dada por el Ruteador y que este generador Simplemente traduce la información.

3. Conclusiones.

Implementar un CAD que cubra todos los pasos necesarios para el desarrollo de un sistema secuencial es una tarea ardua.

Más aún, desarrollarlo como sistema abierto de forma tal que sea posible expandirlo o modificarlo en cada uno de los principales algoritmos es una tarea de precisión.

Y si además le agregamos la opción de transportarlo entre distintas plataformas de sistemas operativos, le estamos dando a este paquete de software un potencial por demás interesante.

Nuevas ideas y mezclas de tecnologías de otras áreas, unidas con técnicas bien conocidas y tradicionales, hacen de nuestro CAD una herramienta muy útil a la hora de implementar diseños secuenciales.

Pensando en el futuro, el compilador de sicilio insertado dentro de este proyecto, permitirá hacer otros tipos de desarrollos con nuevas tecnologías.

Por otra parte, la independización de las reglas de diseño según el fabricante alcanzada a través de una forma nueva de parametrización es un punto sobresaliente de esta herramienta.

Más allá de todas estas ventajas, la posibilidad de influir en forma dinámica a través de los parámetros de restricciones nos permite acelerar los tiempos de diseño de una forma realmente increíble.

4. Referencias.

- [1] Computer Aided Logical Design with Emphasis on VLSI - F. Hill, G. Peterson - John Wiley & Sons, Inc.
- [2] Microelectrónica - J. Galatro, C. Gayoso, C. González, L. Arnone - Secretaría de Ciencia y Técnica, República Argentina.
- [3] Compiler Design In C - A. Holub - Prentice Hall.
- [4] Introduction to VLSI Systems - C. Mead, L. Conway - Addison Wesley.
- [5] VHDL Programming With Advanced Topics - L. Baker - Wiley & Sons, Inc.
- [6] Software L-Edit - Tanner Research.
- [7] Principles of CMOS VLSI Design. A Systems Perspective - N. Weste, K. Eshraghian - Addison Wesley.
- [8] Interaction Semantics of a Symbolic Layout Editor for Parameterized Modules - P. Andersson, L. Philipson - IEEE Trans. on CAD of Integrated Circuits and Systems - Vol. 12, No. 8, August 1993.
- [9] Combining Disk Storage Layout Partitioning and Cell Hierarchies for Efficient Layout Editing - M. Anido C. Oliveira .e Anales del Segundo Workshop IBERCHIP.

EDUCACION Y SOCIEDAD / EDUCATION & SOCIETY

TITULO:	ORTOGRAFIA 1: SOFTWARE EDUCATIVO
AUTOR:	PRUZZO DE DI PEGO, VILMA EULALIA ; DI PEGO, CLAUDIA.
INSTITUCION:	INSTITUTO SUPERIOR DE ESTUDIOS SICOPEDAGOGICOS Y SOCIALES. FACULTAD DE CIENCIAS HUMANAS. UNIVERSIDAD NACIONAL DE LA PAMPA.
DIRECCION:	GIL 353 - (6300) SANTA ROSA - LA PAMPA - ARGENTINA

TITULO:	USING MICROCOMPUTERS TO TEACH POLICY STUDIES SUBSTANCE
AUTOR:	NAGEL, STUART.
INSTITUCION:	UNIVERSITY OF ILLINOIS
DIRECCION:	361 LINCOLN HALL, 702 S. WRIGHT STREET URBANA - ILLINOIS 61801-3696 - USA

TITULO:	PRIMEROS PASOS EN TRANSFORMACIONES LINEALES
AUTOR:	CALIGARIS, MARTA; RODRIGUEZ, GEORGINA; CALIGARIS, ROBERTO.
INSTITUCION:	GRUPO DE INFORMATICA EDUCATIVA. FACULTAD REGIONAL SAN NICOLAS. UNIVERSIDAD TECNOLOGICA NACIONAL.
DIRECCION:	COLON 332 - (2900) - SAN NICOLAS - BUENOS AIRES

TITULO:	NUEVAS HERRAMIENTAS EN LA ENSEÑANZA DE LA FISICA: CIRCUITOS ELECTRICOS
AUTOR:	CALIGARIS, ROBERTO ; CALIGARIS, MARTA ; MANSILLA, GRACIELA.
INSTITUCION:	GRUPO DE INFORMATICA EDUCATIVA. FACULTAD REGIONAL SAN NICOLAS. UNIVERSIDAD TECNOLOGICA NACIONAL.
DIRECCION:	COLON 332 - (2900) - SAN NICOLAS - BUENOS AIRES

TITULO:	RELACION DE LA ENSEÑANZA DE COMPUTACION CON LAS TEORIAS DEL APRENDIZAJE
AUTOR:	HAVLIK, JARMILA.
INSTITUCION:	UNIVERSIDAD DE BUENOS AIRES
DIRECCION:	CANALEJAS 266 - (1609) - BOULOGNE - BUENOS AIRES

TITULO:	AN EXPERIENCE IN A DISTANCE GRADUATE COURSE ON SOFTWARE ENGINEERING
AUTOR:	GARCIA MARTINEZ, RAMON; ROSSI, BIBIANA ; GRAMAJO, EDMUNDO.
INSTITUCION:	CAPIS. GRADUATE SCHOOL. BUENOS AIRES INSTITUTE OF TECHNOLOGY.
DIRECCION:	MADERO 399 - (1106) - BUENOS AIRES

TITULO:	EL TELETRABAJO
AUTOR:	ALLES, MARTA ALICIA.
INSTITUCION:	TOP MANAGEMENT
DIRECCION:	MAIPU 942 - (1340)- BUENOS AIRES

TITULO:	AMBIENTE BASADO EN HIPERMEDIOS COMO SOPORTE AL APRENDIZAJE DE LA TECNOLOGIA DE OBJETOS
AUTOR:	NICÓLAU, S.; OLSINA, L.; LAFUENTE, G. ; BERTONE, E.
INSTITUCION:	GRUPO DE INVESTIGACION Y DESARROLLO EN INGENIERIA DE SOFTWARE. DEPARTAMENTO DE COMPUTACION. FACULTAD DE INGENIERIA. UNIVERSIDAD NACIONAL DE LA PAMPA
DIRECCION:	GIL 353 - (6300) SANTA ROSA - LA PAMPA - ARGENTINA

XIV CONGRESO DE INFORMÁTICA, TELEINFORMÁTICA, Y
TELECOMUNICACIONES -USUARIA '97

IX FORO DE SISTEMAS ABIERTOS Y COMUNICACIONES
UNIFORUM '97

V SIMPOSIO DE INTELIGENCIA ARTIFICIAL- SIAR '96

II SIMPOSIO SOBRE CIENCIAS HUMANAS, INFORMÁTICA Y
TELECOMUNICACIONES

ÁREA: EDUCACIÓN

TRABAJO: ORTOGRAFÍA 1: SOFTWARE EDUCATIVO

AUTORAS:

Dra. VILMA PRUZZO de DI PEGO.
A. en Comp. CLAUDIA DI PEGO de MAC ALLISTER.

INSTITUCIONES RESPONSABLES:

Facultad de Ciencias humanas. Universidad Nacional de La Pampa.
Instituto Superior de Estudios Sicopedagógicos y Sociales.

DIRECCIÓN:

Oliver 125 CP 6300. Santa Rosa, La Pampa. TE/ FAX: (0954) 54324 o FAX:
(0954) 33037

E-mail: postmaster@isess.lp.mdc.edu.ar

1. INTRODUCCIÓN

1919

El Inspector General de escuelas, Juan F. Jáuregui, en el Prólogo de la obra "Apuntes de gramática Castellana" de Luisa Alternare, editado en 1919, plantea concretamente el problema escolar que representa el aprendizaje de la **ortografía**:

"...debe enseñarse muy mal puesto que la deficiencia más saliente que se revela en los niños que egresaron de la escuela primaria, consiste en su carencia de aptitudes para escribir: **la ortografía es mala y la construcción es peor...**"

1997

Ya casi a fines de siglo el problema esbozado por el Inspector Jáuregui continúa vigente. Nuestros niños después de siete años de escolaridad, egresan con dificultades marcadas en su escritura. Según la Evaluación Nacional realizada por el Ministerio de Cultura y Educación de la Nación, el rendimiento escolar en Lengua se ubica en un nivel que los evaluadores han calificado **como señal de alerta***. Se especifica que dentro de la evaluación de Nociones y reglas gramaticales, presentó dificultad especialmente, el reconocimiento de familias de palabras, 'el reconocimiento de clases de palabras y la-

ortografía. Un viejo problema que aún subsiste y que ha sido incluido nuevamente con peso dentro de los contenidos Básicos Comunes de la Escuela General Básica aprobados por el Consejo Federal de Educación.

La enseñanza de la ortografía ha perdido vigencia en las escuelas en sucesivas oleadas de argumentos: primero porque era un resabio mecanicista de la escuela tradicional, porque era necesario respetar el estilo de escritura del alumnos, y últimamente porque los alumnos ‘solos irían construyendo las nociones correspondientes. Esta visión autónoma del aprendizaje ha cedido su postura radicalizada para reconocer las posibilidades de la intervención docente en la construcción de las nociones lectoescritoras. Mientras los docentes se debatían ante estas contradictorias posturas, la ortografía siguió siendo un conocimiento valorado socialmente. Al requerirse puestos de trabajos, la misma entrada a un Banco implica la redacción manuscrita de una carta de presentación donde los aspectos sintácticos, gramaticales y especialmente ortográficos son tenidos en cuenta. Por lo tanto se hace necesario reformular en la actualidad la enseñanza de la ortografía, su finalidad, sus tiempos curriculares, las metodologías que posibiliten la construcción de aprendizajes significativos, los soportes instruccionales que se adecuan a su aprendizaje. Por eso este software, ayuda al docente o psicoterapeuta para que se derive a un sistema autoasistido el aprendizaje de una convención social como es la ortografía, dejando su tiempo docente liberado a los aspectos productivos del lenguaje.

Sin embargo, ‘no sólo se hace necesario revisar la enseñanza de la ortografía sino que se debe dar respuestas a los psicoterapeutas que tienen en tratamiento casos severos de disortografía. En este sentido el software que presentamos puede destinarse tanto a la enseñanza como a la recuperación psicopedagógica de las disortografías.

2. LA ENSEÑANZA DE LA ORTOGRAFÍA

En nuestra perspectiva la enseñanza de la ortografía no lleva un fin implícito en sí misma. No es sino un aspecto más de la enseñanza tendiente a promover la competencia lingüística. Por eso no se parte de reglas ortográficas sino que nuestra propuesta transforma la palabra misma en núcleo problemático, que encierra a la vez significados, forma, sonido. Se toma pues a la palabra con una visión globalizadora, sin dar prioridad a la forma sobre el significado, para transformarla en formas, a veces, portadora de múltiples significados. En el proceso de comunicación, el mensaje mismo puede hallarse comprometido cuando parte del código no se emplea con corrección. Alteraciones gráficas y ortográficas pueden impedir la decodificación del mensaje. Se presenta entonces a la palabra en su multifacético aspecto: como sostén del pensamiento, unidad significativa, eslabón básico de la comunicación, estructura portadora’ de una única y posible’ composición. Y al decir del poeta Rilke, la palabra como instauradora del mundo.

En Ortografía 1, se han seleccionado doscientas palabras en las que se analizan sonidos, formas y Significados; se desentraña su raíz originaria para reconocerla como palabra primitiva o derivada, organizándolas en familias de palabras; se las emplea en sentido estricto o figurado para jugar con la polisemia que encierra. Y en este juego se amplía el repertorio léxico que implica el crecimiento cognitivo de las redes conceptuales del alumno.

Como el aprendizaje ortográfico se ha planificado a través de tres niveles, la metodología abarca’ el Nivel I, II y III, de los cuales con Ortografía I se presenta sólo el primero. Los tres niveles podrían desarrollarse en tercero, cuarto y quinto grados,

respectivamente, con lo cual no sólo habremos logrado culminar el aprendizaje ortográfico en quinto grado sino que se habrán incorporado seiscientas palabras al repertorio léxico del alumno. Las investigaciones han demostrado que un vocabulario visual de mil palabras asegurarían la comprensión lectora de los textos del nivel medio. Por lo tanto este sistema más allá de la ortografía, también apunta la comprensión lectora. Recordamos que para la investigadora latinoamericana María Carbonell de Grompone (1980), la carencia de vocabulario en la memoria del lector, lo obliga a retornar a etapas arcaicas de lectura, descifrando los caracteres para poder leer, porque se halla comprometido el barrido rápido que sólo es posible cuando las palabras figuran en los archivos de la memoria. De esta manera la lentitud lectora que caracteriza a nuestros alumnos con dificultades, compromete finalmente la comprensión del mensaje por el esfuerzo decodificador que implica.

Las doscientas palabras a su vez, son portadoras de alguno de los doce grupos ortográficos seleccionados: b, c, g, h, j, ll, q, r, s, v, x, z. Según la clasificación estructural de los errores ortográficos de María Carbonell de Grompone podemos analizar más allá de la existencia del error ortográfico, la calidad del mismo, si es más o menos grave según el grado que cursa el alumno. Para Carbonell, existirían cinco niveles de dificultades:

a.- Clave primaria de escritura, en la que a cada grafema le corresponde un fonema. A la forma **m** le corresponde un único sonido.

Pertenerían a esta clave, las vocales, la p, t, f, l y las combinaciones ca, co, cu, ga, go, gu, ja, jo, ju, que no tienen otra forma para esos sonidos.

b.- Clave secundaria de escritura: Se avanza en complejidad ya que se incluyen en este nivel: . **fonemas** que se signan con dos grafemas (rr, ch, ll) ; **grafemas** cuya escritura varía según la sílaba de que forman parte (gue, gui; güe, güi; que, qui; ce, ci).

c.- Clave terciaria o nivel ortográfico I: Aparece en este nivel los aspectos verdaderamente complejos, de la escritura española: grafema h que no corresponde, a ningún fonema; fonema b que puede signarse b o v; fonema j que puede signarse j o g; fonema q que puede signarse q, c, o k; fonema y que puede signarse y o ll; fonema ñ que tiene escasas diferencia con nía, nie, nio, niu; combinación np, nb que deben signarse mp y mb; fonema s que puede signarse s, c, z, o x;

Los niveles IV y V de mayor complejidad no se abordan en Ortografía I Nuestra actual propuesta conforma la parte inicial de lo que hemos llamado el Método Integrador y Cíclico de la Lengua, el MIC 1 que se centra en general en la clave secundaria y en el nivel ortográfico I pues se considera aprendida la clave primaria al concluir nuestro Método Generativo de la Lengua de 2º grado. Sin embargo, se han incorporado a este MIC 1 algunas de las claves primarias de escritura que suelen confundir los niños: uso de ga, go, gu; ja, jo, ju. De esta manera quedan seleccionados los **contenidos ortográficos de MIC I**.

A partir de estos contenidos nos hemos planteado los **objetivos** a lograr, concebidos los mismos como procesos constructivos que involucran operaciones mentales aplicadas sobre los contenidos seleccionados:

Discriminar fonemas y grafemas correspondientes a las doscientas palabras seleccionadas.

- . Explicar significados en su empleo estricto o figurado.
- . Incrementar el repertorio léxico.
- . Construir familias de palabras teniendo en cuenta las palabras primitivas y las derivadas.
- . Separar las palabras en sílabas para reconocer la sílaba tónica.
- . Interpretar las consignas para realizar tareas y evaluaciones.
- . Generalizar el empleo de los grafemas analizados a otras palabras y contextos.

3. LA METODOLOGÍA PLANTEADA

En la perspectiva de Ana Quiroga la función de sostén que ejerce la madre cuando enseña a su bebé implica la **mostración** por parte de la madre del objeto del mundo con cuya apropiación el bebé va a satisfacer sus necesidades. En su función de sostén el docente tiene que "mostrar" este objeto cultural que es la escritura para que el niño pueda ejercer la función discriminadora, pero mostrarla de una forma que implique la necesidad del alumno de apropiársela porque esa tarea resulta gratificante. Por lo tanto, se pensó un sistema que permitiera la **mostración**, implicar-a al niño en una actividad y que la misma resultara atractiva y gratificante.

En el MIC la **mostración** asume las más variadas formas de impacto visual: está el soporte icónico, con imágenes representativas de las palabras seleccionadas y animaciones de las mismas; se juega con los significados; se destaca gráficamente el grupo ortográfico que se trabaja. Y está siempre presente el principio de actividad del viejo Comer-río: se aprende a escribir, escribiendo. Por eso el niño tiene planteadas numerosas actividades que lo involucran en el uso de algún grupo ortográfico, en la búsqueda del sentido estricto o figurado de una expresión, en la construcción de familias de palabras. La siguiente pantalla está diseñada para presentar el uso de ce, ci en tres palabras de uso común. Las mismas se muestran con su escritura correcta y la tarea del niño consiste en completar lagunas otorgando significados por el contexto. A su vez, aparece el grupo ce, ci destacado en el ángulo superior para que la tarea no resulte una escritura adivinanza sino que cuente con los elementos de análisis que implique el pensar en el hacer (ver figura 1).

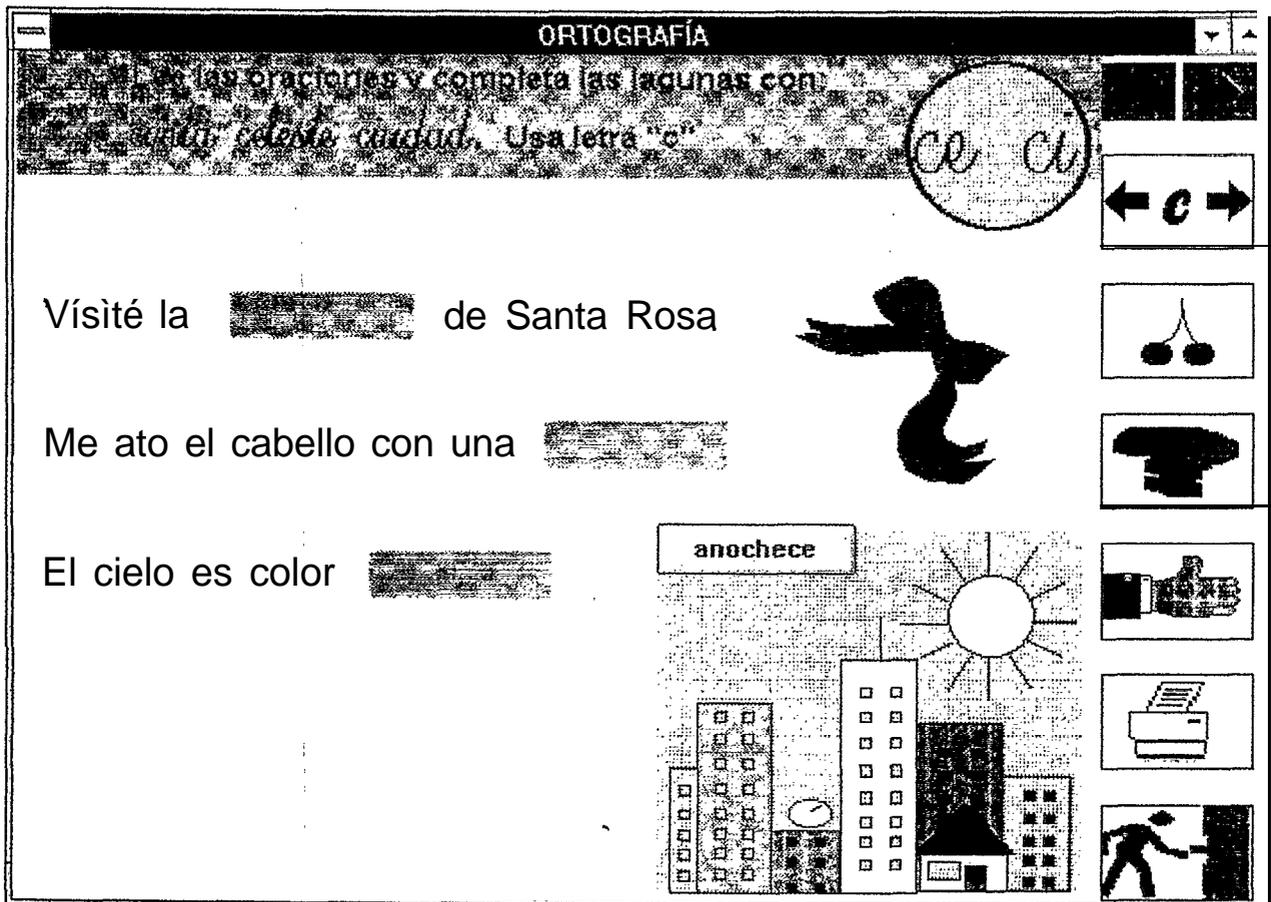


FIGURA 1

El MIC, tiene una estructura cíclica que lo caracteriza: en vez de trabajar una a continuación de otra las dificultades ortográficas como habitualmente se plantea en la escuela (primero se enseña la b, luego la c, a continuación la h, etc.) se presentan grupos ortográficos que se alternan y a los que se retorna una y otra vez. Por ejemplo, la primera unidad presenta simultáneamente actividades con b, c, g, h, j, ll. La segunda Unidad, con q, r, s, v, z x. La tercera Unidad retorna a los grupos iniciales, la cuarta al segundo grupo y así sucesivamente. Entonces si en la Unidad 1, se trabaja con la b; terminaciones aba, en la tercera Unidad se vuelve a la b, pero en terminaciones bunda, por ejemplo, Esta estructura cíclica, con el retorno que implica una y otra vez a los mismos grupos ortográficos, favorece la toma de conciencia ortográfica al facilitar la memoria comprensiva de los usos correctos (la representación gráfica de esta estructura cíclica se puede ver en la figura 5 que a su vez es el menú de acceso que presenta el software a las distintas unidades).

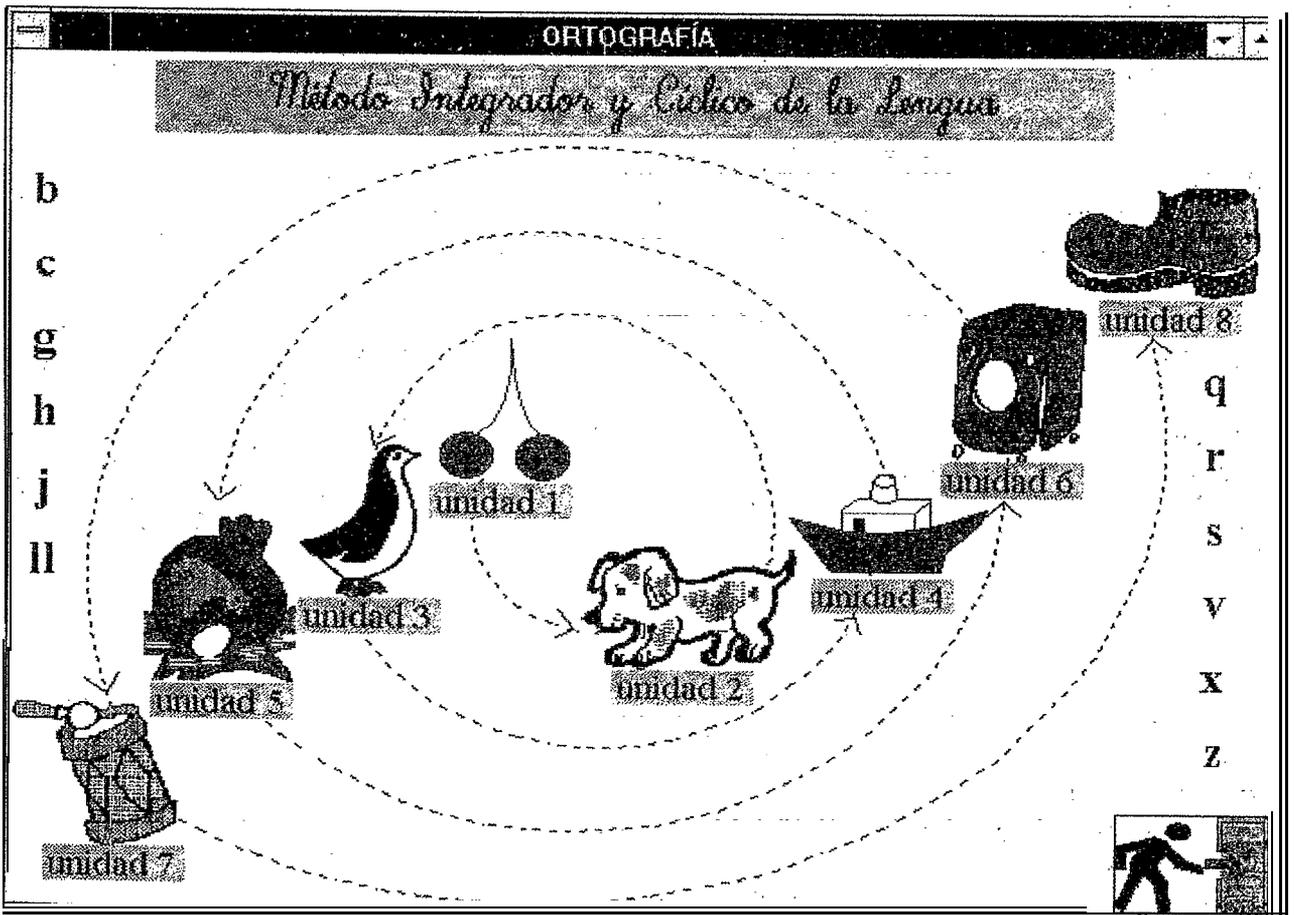


FIGURA 5

4. LA CONCIENCIA ORTOGRÁFICA

La toma de conciencia en el aprendizaje fue claramente descrito por Claparede en la década del 30 y luego retornado por Piaget que cita al autor reiteradamente. Señala Claparede (1932):

Dándose cuenta, participando la conciencia de las faltas que hace el niño al hablar primero la evitará conscientemente, merced a una acción voluntaria y después; tomará la costumbre de no cometer esa incorrección". Ese primer acto de toma de conciencia deja abierto el camino a la posterior práctica inconsciente de los actos. Ya no se necesita "pensar" cómo se dice una palabra., Así sucede con la ortografía: si nos damos, como psicopedagogos o docentes, el tiempo necesario para la toma de conciencia sobre los usos

ortográficos, luego se transforman en práctica inconsciente, ya no demandarán esfuerzo de pensamiento y este quedará liberado exclusivamente para la actividad lingüística productiva. Esta toma de conciencia implica una actividad que implique al pensar para su resolución pero que a la vez no sea tan difícil que desanime a la acción.

La toma de conciencia es un objetivo muy difícil de lograr y el terapeuta o docente advertirá su adquisición cuando el alumno logre detener su escritura ante una duda, o consulte sobre un uso específico. Pero de nada serviría esa toma de conciencia si no se habilitan procedimientos que le permitan al niño la autonomía en la construcción ortográfica. Por eso se trabajan específicamente dos procedimientos de ayuda para encontrar el uso ortográfico correspondiente. Uno de ellos es la de facilitar al alumno la comprensión de la derivación de palabra a partir de una primitiva, la construcción de familias de palabras. Por ejemplo, si ha logrado comprender este proceso a través de las actividades del MIC, al presentársele una palabra de dudosa ortografía como podría ser la palabra deshacer, se detendrá ante la duda y tratará de encontrar la voz primitiva de ese término. El segundo procedimiento al que tiene que recurrir ante la duda ortográfica es la búsqueda de alguna de las reglas aprendidas en el MIC. Se recuerda que en la estructura metodológica se parte de palabras y no de reglas ortográficas, pero luego de la escritura de las mismas se analizan regularidades. Por ejemplo si se pide que escriban en pasado pescar, buscar y tocar, pero agregando aba, la transformación en tocaba, buscaba y pescaba lleva a encontrar una regularidad cuyo uso continuado permitirá su aprendizaje como regla (la generalización).

En síntesis entendemos como **conciencia ortográfica** el interés, la disposición y el deseo del alumno de escribir sin errores y de comprobar la corrección de sus escritos. Es decir que los procedimientos de análisis ortográfico (familias de palabras, reglas o diccionario) constituyen un aporte cognitivo superfluo si no se desarrolla la conciencia ortográfica. Pero, por otra parte su sola existencia no asegura la correcta escritura sino cuando se poseen los procedimientos de análisis. Este doble aprendizaje hace posible finalmente la generalización de 'lo aprendido a otras palabras y contextos comunicacionales.

5. CARACTERÍSTICAS DEL SOFTWARE

El ambiente en el cual se desarrolló el software es Toolbook 1.53 de Asimetrix. Toolbook usa la estructura de libro como metáfora -Cuando se crea una aplicación se crea un libro formado por páginas. El ambiente de programación es manejado por eventos y orientado a objetos, y se estructura como una jerarquía que soporta herencia simple, corre bajo entorno Windows y es una excelente herramienta para la creación de software multimedia reusable y fácilmente adaptable.

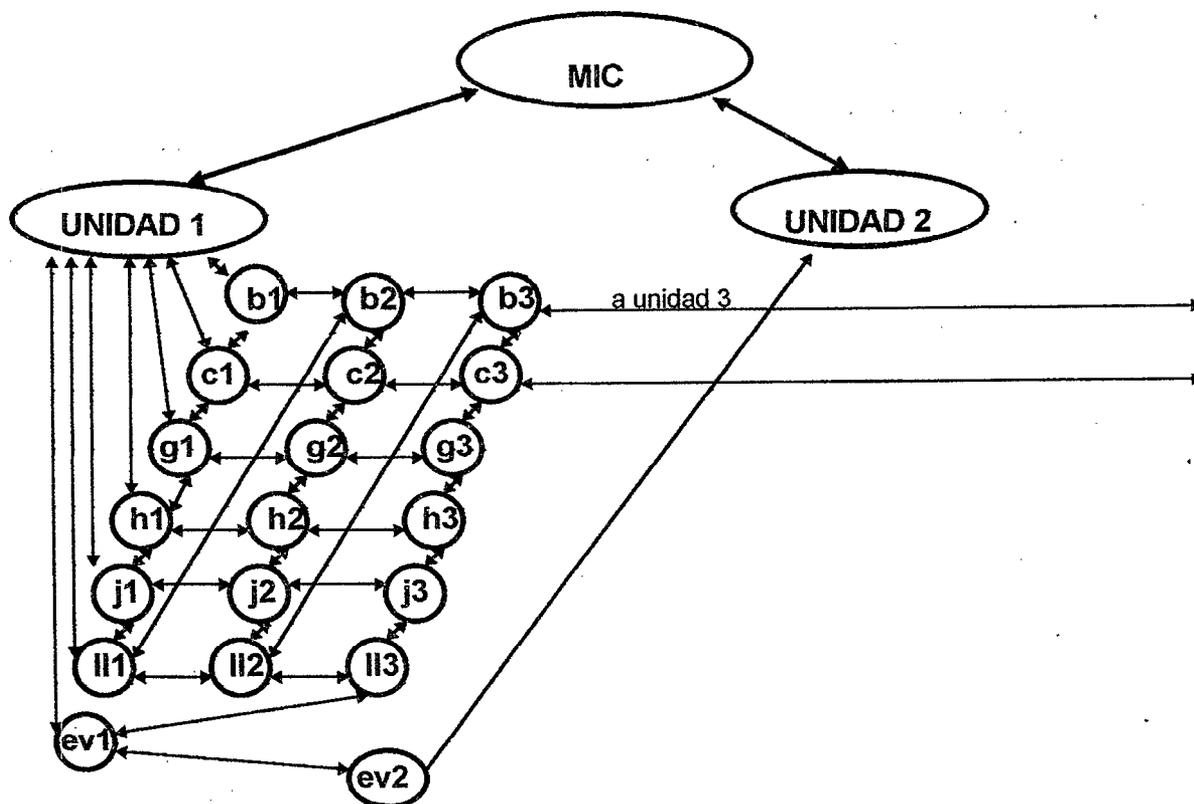
El software es altamente interactivo; con una interface que considera el nivel de desarrollo de los niños de 7 años y es para ellos simple de operar. Presenta diseños de pantallas sencillos: para evitar la dispersión de la atención pero a su vez con ilustraciones y animaciones para que resulte atractivo. Provee un sistema de ayudas sensibles al contexto (al que puede acceder el alumno y el maestro) que proporciona información sobre la consigna que corresponde a esa pantalla y sobre la interface en uso.

El software ORTOGRAFÍA I respeta la estructura cíclica propuesta por el método y la potencia, ya que permite recorrer las actividades de acuerdo a los intereses y necesidades de cada usuario. Se presenta un esquema de navegación que permite en primera instancia elegir la Unidad ortográfica con la que se va a trabajar, mostrando las letras y grupos que se ejercitan en esa unidad (ver figura 5). Cada una de las Unidades abarca 6 (seis) grupos ortográficos, por ejemplo; b, c, g, h, j, ll y está representada por una

imagen de alguno de los usos ortográficos que se presentan. Por ejemplo la Unidad I se representa con la imagen de guindas. Una vez que se accede a la Unidad, se presentan todas las palabras y las letras o grupos que se van a trabajar (ver figura 3).

El usuario puede realizar las actividades siguiendo un recorrido secuencial de las mismas o puede elegir una letra específica con la cual trabajar. Por ejemplo al entrar a la Unidad 1, se puede elegir la ejercitación con “b” en las palabras saltaba, buscaba, pescaba y luego seguir con las actividades programadas para el uso de “c”, tal como está diseñada la Unidad. Pero también puede optar seguir todas las tareas previstas con la “b”. Este recurso es muy valioso para el psicoterapeuta que tiene alumnos, de grados superiores o adolescentes que quieren superar su disortografía, y que pueden optar aprender el uso de las letras o grupos en los que más dificultades presentan.

A continuación se presenta un gráfico reducido de la estructura de navegación del software. Cada óvalo representa un nodo o pantalla del software., Algunos nodos son fundamentalmente para proveer acceso a diferentes actividades, por ejemplo los nodos “MIC”, “unidad 1” “unidad 2”. Los nodos representados por letras y números contienen una actividad para esa letra, por ejemplo “b1” contiene la primer actividad para la letra “b”. Los nodos “ev” contienen actividades de evaluación de la unidad respectiva.



Luego de la última actividad, de evaluación de una unidad se pasa a la pantalla inicial de la unidad siguiente. La unidad 2 trabaja con las letras q, r, s, v, x, z, y la unidad 3 vuelve a trabajar con b, c, g, h, j, II, y así sucesivamente, por eso la navegación de la letra “b” después de las actividades de la unidad 1 continúa en la unidad 3, etc.; la misma organización se repite para las demás letras.

Las páginas de tareas cuentan con una barra de íconos a la derecha y al pasar el puntero del mouse sobre ellos se despliega un cartel con el nombre de la acción que se ejecuta al cliquearlo. Dicha barra está formada por los siguientes íconos (ver figura 4):

Dos flechas: hacia la derecha y hacia la izquierda, que permiten adelantar una página o volver a la anterior, para el recorrido secuencial de las actividades

Dos flechas con una letra en el medio : permite recorrer las actividades relativas a dicha letra o grupo.

Una imagen representativa de Unidad: permite volver a la Unidad desde cualquier actividad. Por ejemplo, la Unidad 1 está representada por guindas, para volver desde alguna ejercitación a la pantalla inicial de la unidad se cliquea sobre las guindas.

Un signo de pregunta: proporciona información de ayuda sensible al contexto, tanto sobre la consigna de la actividad como al uso de la interface. La ayuda se presenta en un cuadro de texto que desaparece al cliquear sobre él.

Una mano: Desencadena el proceso de **evaluación**, después de la evaluación este ícono es reemplazado por una goma de borrar.

La croma de borrar: permite volver la pantalla al estado inicial.

Una impresora: al cliquearla se imprime la actividad realizada.

Una figura humana abriendo una puerta: con ella se abandona el programa, previamente se pide la confirmación o cancelación de la acción.

6.ALGUNAS ACTIVIDADES PREVISTAS EN EL MIC

. Escribir nombres de cosas o acciones a partir de la imagen que los representa.

.Identificar la “madre” en una familia de palabras, y luego escribir las derivadas (ver figura 2).

.Para analizar las variaciones de las palabras derivadas, se trabaja transformando terminaciones (ar, car, etc) Se cliquea la terminación y ésta se desplaza hacia abajo dejando un cuadro de color en el cual se escribe la nueva terminación. Por ejemplo “tocar” se cliquea sobre “car” y se agrega “aba”. Luego se presenta el espacio para escribir la nueva palabra formada.

. Separar palabras en sílabas.

. Completar lagunas en frases incompletas (ver figura 1).

. En múltiple choice, interpretar sentido estricto y figurado identificando la expresión correcta.

. Aparear palabras con sus respectivos significados, etc.

7. CONCLUSIONES

El MIC es una herramienta informática para la recuperación de las disortografías a partir de un minucioso diseño psicodidáctico. Se parte de grupos ortográficos seleccionados entre las claves secundarias y terciarias de escritura, y se los trabaja transformando la palabra misma en núcleo problemático. La metodología presenta un avance cíclico, retornando permanentemente a los grupos ortográficos seleccionados, reemplazando el habitual tratamiento lineal, secuencial de esos contenidos. Las actividades permiten la **mostración** de la dificultad ortográfica, **su consolidación** a fin de facilitar finalmente su **generalización**. Un Sistema de evaluación continua, actúa como ayuda contingente para el aprendizaje, y toma la forma de autoevaluación que implica el pensar en el hacer. Este sistema se estructura en torno a la necesidad de desarrollar la conciencia ortográfica a la vez que se brinda a los alumnos procedimientos de análisis que le permitan reflexionar sobre la escritura.

8. BIBLIOGRAFÍA

- AEBLI, H. 1973. Una didáctica fundada en la Psicología de Jean Piaget. Buenos Aires, Kapeluz.
- AUSUBEL D. 1973. Algunos aspectos psicológicos de La estructura del conocimiento. En La educación y la estructura del conocimiento. Buenos Aires, El Ateneo.
- BRUNER, J. '1969, Hacia una Teoría de la instrucción. México, Uthea.
- CARBONELL DE GROMPONE, M. 1980. "Evolución de la ortografía según la clasificación estructural de los errores ortográficos". Lectura y vida Revista latinoamericana de lectura. Buenos Aires. Año I n°4.
- CLAPAREDE, E. 1932. La educación funcional. Madrid, Espasa Calpe.
- COLL, C. 1990. Aprendizaje escolar y construcción del conocimiento. Buenos Aires, Paidós educador.
- COMENLO, J. 1976.' Didáctica Magna. México, Editorial Porra.
- GAGNE, R. 1977. Los principios básicos del aprendizaje para la instrucción. México, Diana.
- GARDNER, H. 1993. La mente no escolarizada. Barcelona, Paidós.
- MINISTERIO DE CULTURA Y EDUCACIÓN DE LA NACIÓN, CONSEJO FEDERAL DE CULTURA Y EDUCACION 1995. Contenidos Básicos Comunes para ta Educación General Básica.
- MINISTERIO DE CULTURA Y EDUCACIÓN DE LA NACIÓN, 1995. Educación: un proyecto en marcha.
- NIELSEN, J. 1990. "Hipertext and Hipermedia". Academic Press.
- PERRENOUD, PH. 1990. La construcción del éxito y del fracaso escolar. Madrid, Morata.
- PIAGET,. J. 1973. Estudios de Psicología genética. Buenos Aires, Emecé.
- 1984. Psicología de la inteligencia. Buenos Aires, Psique.
- 1985. Psicología y Pedagogía. Buenos Aires, Ariel.
- PIAGET, J. y B. INHELDER. 1975. Psicología del niño. Madrid, Morata.
- QUIROGA, Ana P. de. 1980. Proceso de constitución del mundo interno. Buenos Aires, Ediciones Cinco.
- QUIKOS, J, El lenguaje Lectoescrito y sus problemas. Buenos Aires. Ed. Médica Panamericana.
- 1980 a. Evolución de la ortografía según la clasificación Estructural de los errores Ortográficos . En "Lectura y Vida". Revista Latinoamericana de Lectura, Buenos Aires, Año 1 N4.
- 1986. Dislexia Escolar y Dislexia Experimental. En FERREIRO, E. Nuevas Perspectivas sobre los procesos de lectura y escritura, Buenos Aires, Siglo XXI.
- TALLIS, J. y SOPRANO, A. 1991. Neuropediatría, Neuropsicología y aprendizaje. Buenos Aires, Nueva Visión.
- TORRES SANTOME, J. 1991. El currículum oculto. Morata. Madrid.
- VIGOSTKY, LS. 1984. El desarrollo de los procesos psicológicos superiores. La Pléyacle. Buenos Aires.
- WINNtCOTT, D. 1992. Realidad y juego. Barcelona, Gedisa.

VI. USING MICROCOMPUTERS TO TEACH POLICY °STUDIES SUBSTANCE

A number of articles have been written on how useful microcomputers can be for teaching social science research methods, which generally means teaching statistical analysis. Microcomputers are useful for that purpose because students can quickly obtain output from archived data or inputted data and then experiment with changing the inputs and the procedures to gain insights as to the effects and meaning of various aspects of statistical analysis. This can be done while the instructor lectures, explains, and discusses what is or should be happening. Microcomputers can also be used by students for organizing data, drafting reports, retrieving information, and interacting with other people. It is the purpose of this appendix to describe a way in which microcomputers can be used to teach public policy substance rather than methods or Office practice, especially where the substance involves controversial issues.

A. FOUR IMPORTANT ELEMENTS

The essence of the substantive microcomputer approach is ideally to work with the following procedures:

1. All class sessions are organized in terms of controversial issues.
2. All students have access to shared microcomputers in each class session.
3. All microcomputers have access to data files for each controversial issue.
4. All data files have access to software that is capable of processing the data so as to maximize the learning experience.

On the first point, arranging for a variety of controversial issues can easily be done, given the nature of public policy subject matter. Doing so may be a desirable approach to teaching, regardless of whether one is using microcomputers. There are now textbooks for many courses organized in terms of controversial issues, such as Leslie Lipson, *The Great Issues of Politics: An Introduction to Political Science*; Herbert Levine, *Political Issues Debated: An Introduction to Politics*; and George McKenna and Stanley Feingold (eds.), *Taking Sides: Clashing Views on Controversial Political Issues*. Any course that deals with constitutional law or other court cases inherently deals with controversial issues, as manifested in the majority, concurring, and dissenting opinions. Likewise, any substantive public policy course tends to be inherently organized in terms of controversial policy issues.

On access to hardware, if there is a shortage of microcomputers, then have 3 to 5 students per microcomputer. Each micro can be put on a round table where, 3 to 5 students can each have access to the keyboard. That is not only less expensive, than a one-on-one arrangement, but it may also provide more learning experience because the 3 to-5 students tend to help and reinforce each other. Arranging for hardware can be more easily done now, by virtue of the availability of IBM FC portables at many universities for only approximately \$1,000 with discounts. Thus at

a cost of only \$ 10,000 one can obtain enough hardware to service large classes of 50 students apiece when they are seated 5 students to each large round table.

On the third point of arranging for the data files, a data file in this context means a matrix showing (1) alternatives to choose among on the rows, (2) criteria or goals on the columns, and (3) relations between alternatives and criteria in the cells. Such a matrix can facilitate debating, discussing, and choosing the best alternative, combination, or allocation. There are now almost 200 such data files for issues that relate to political science, public policy, and law available from Decision Aids, Inc., 1720 Parkhaven Drive, Champaign, IL 61820. The data relate to such institutional issues as federalism, separation of powers, judicial review, party systems, democracy versus dictatorship, and socialism versus capitalism. The data files also relate to public policy controversies in the realm of agriculture, civil liberties, criminal justice, defense, economic regulation, education, energy, environmental protection, foreign policy, health, housing, labor, minorities, women, poverty, technology, transportation, and zoning, which covers a to z in policy controversies. The data files also relate to numerous Supreme Court cases individually or in sets. Seeing the format of these data files and using them can stimulate instructors to develop more data files, including ones of special relevance to their courses.

On the fourth point of arranging for software that can process such data, the processing should include

1. Allowing changes in the alternatives, criteria, and relations in order to facilitate experimenting and creativity;
2. Showing what alternative, combination, or allocation is best in light of (1) the alternatives available, (2) the criteria and their relative weights, and (3) the relations between the alternatives and the criteria; and
3. Showing what changes would be necessary to bring a second-place alternative up to first place.

For that kind of software, one can obtain a copy of the Policy/Goal Percentaging software from Decision Aids, Inc., along with the 200 data files for only \$40 as of 1988. That includes a Datafiles Manual, a Program Manual, and summarizing articles. The program is called Policy/Goal Percentaging because it relates policies or alternatives to goals or criteria and uses part/whole percentaging to deal with the criteria being measured in different ways. The software is capable of processing the alternatives, criteria, and relations for any substantive problem in order to indicate the best alternative, combination, or allocation in light of those alternatives, criteria, and relations. The software is also capable of flexibility in allowing additions, subtractions, and other changes in the alternatives, criteria, and relations. It can also indicate what changes are needed in order to bring the second-

place alternative up to first place.

B. BENEFITS AND OBJECTIVES

Microcomputers provide the following benefits for teaching purposes beyond what can be provided by good teaching materials and class sessions:

1. Microcomputers help students to absorb information through interactive questions and answers.
2. They are useful for role playing that helps students to learn better various skills and procedures.
3. They can stimulate creativity on the part of students by allowing them to easily input new data, ideas, and methods in order to see their effects.
4. They can be used along with lectures, discussion, textbooks, and other traditional teaching devices. One does not have to choose between such devices and microcomputers.
5. Relevant hardware is becoming less expensive and much more widely available.
6. There is an increasing quantity of software being developed for a variety of courses in political science, policy studies, public administration, and related fields.
7. Faculty and trainers are becoming more knowledgeable about the potential of microcomputers, and students are becoming more receptive to using them.

The use of microcomputers for teaching either methods or substance in undergraduate or graduate courses can have a significant impact on education. For example, at the 1984 annual meeting of the American Society for Public Administration conducted an all-day workshop dealing with decision analysis in public administration. The participants asked questions and made comments during the day. In their evaluations, however, they complained about the lack of participation. At the 1985 ASPA annual meeting I conducted the same all-day workshop, but this time with microcomputers and hands-on experience. There were virtually no questions or comments during the day. The participants were practically enthralled by the experience of working with the microcomputers. In their oral and written evaluations they commented favorably on the feeling of participation that working with the microcomputers gave them, even though the oral participation was substantially less. In the context of teaching public policy analysis or political science, there are five major learning objectives in using microcomputers to facilitate role playing. The first is to enable the students to process a set of policy goals to be achieved, available alternatives for achieving them, and relations between goals and alternatives in order to choose the best alternative, combination, or allocation. The second is to enable the students to deal with such policy analysis problems as (1) multiple dimensions on multiple goals, (2) multiple missing information, (3) multiple alternatives that are too many for it to be possible to determine the effects of each one, (4) multiple and

possibly conflicting constraints, and (5) the need for simplicity in spite of all this multiplicity.

The third objective is to enable the students to handle sensitivity analysis, whereby one determines how the bottom-line conclusion in a policy analysis is likely to change as a result of changes in the goals, alternatives, relations, weights, constraints, and especially the various ways of handling the analytic problems mentioned in the second objective. As a fourth objective, the courses can enable the students to work with microcomputers, which can facilitate a fast and meaningful handling of the inputs and procedures. The final objective is to enable the students to report the results of these policy analysis activities in a way that is clear and helpful to governmental decision-makers and other political scientists.

C. SPECIFIC PROCEDURES

To further stimulate involvement where there are 5 students per microcomputer, the students can be grouped in terms of their ideological or substantive interests. For example, with 25 students, the five sets might cover relatively strong liberals, mild liberals, neutrals, mild conservatives, and relatively strong conservatives. To facilitate interaction across such groups, students can be encouraged to suggest new alternatives, criteria, and relations for the other groups to try out. That kind of interaction facilitates innovative compromises that may be even better than the best expectations of either side, rather than just better than the worst expectations. Instead of dividing the students into ideological groups, they can be divided in terms of their substantive interest. That can be especially useful if the groups are going to frequently create their own data files.

An important point is that the data files do not require precise quantitative data. One can do meaningful analyses when relations are expressed on 1-2 (or no-yes) scales, or 1-3 1 (or no-maybe-yes) scales, or more precise scales that relate to years, miles, dollars, percentages, or other measurement units. The software is capable of dealing with a variety of measurement units by such methods as converting them all to 1-5 scales or part/whole percentages. The software is also capable of showing that there is no need to argue over whether a relation is scored 60 or 80 because anything above 45 arrives at the same results. The software can handle missing information by determining critical values above which the decision would go one way and below which another way. One then merely has to determine whether an item is above or below that threshold rather than to determine the item's exact score. The software can also handle problems where the number of alternatives is too great for it to be possible to determine the effects of each one,

Policy Evaluation

and where conflicting constraints are present.

An illustrative problem might be how to provide legal services for the poor, as part of a political science course on the judicial process. The class might go through the following steps when it is creating a new data file rather than manipulating an existing one:

1. Start out with some policies and some goals. For example, the alternative policies might be salaried government lawyers versus unpaid volunteers. The goals might be inexpensiveness, accessibility, political feasibility, and competency.
2. Add to the policies and the goals in light of the initial set.
3. Consolidate policies and goals that heavily overlap, as indicated by the fact that they are likely to have the same relation scores.
4. Divide policies and goals when one realizes that the initial wording may have been too gross.
5. Insert relation scores using pluses and minuses. Go down one column or goal at a time in order to promote consistency within each goal.
6. Convert the pluses and minuses into numbers on 1-3 or 1-5 scales. Revise the relation scores if one thinks that finer distinctions are sometimes necessary, such as a score of 2.5.
7. Decide on tentative weights for the goals by giving the least important goal a weight of 1 and expressing the weights of the other goals as multiples of the first goal. Do not weight the goals if they all seem about equally important, or if the weighting seems too difficult. The weighting can be postponed until after one sees the results, because one then often realizes that different weights will not change the results. The results may also help clarify how much difference there has to be in order to change the results.
8. Sum across each policy in order to determine its overall score. If the goals have different weights as multipliers, then sum the products across each policy. Each product involves the relation score multiplied by the weight of the goal to which it pertains.
9. Observe the unweighted or weighted summation scores to see if there is one outstanding policy, a scattered combination of outstanding policies, or an optimum-level pattern where doing too little or too much is undesirable.

Relevant questions for exams and class might relate to the following:

1. What changes in the relative weights of the criteria (or changes in the scores of the alternatives on the criteria) are needed in order to bring a certain second-Place alternative into first place? For example, given the data file on the Supreme court case of *Ballew v. Georgia*, how much and why does the weight of avoiding a conviction have to rise from 1.00 relative to the weight of avoiding a

wrongful acquittal in order for a 12-person jury, a 6-person jury, or a single decision-maker to be optimum?

2. What are the effects of adding or subtracting a criterion or alternative on the bottom-line solution as to which alternative is best? For example, given the data file on the effects of alternative means of resolving criminal cases, how and why will the initial allocation change between trials and plea bargains if expense is added as a criterion to delay reduction and respect for the law, or if diversion is added as an alternative means of resolving cases?
3. What changes in the alternatives, criteria, or relations are needed to have a predictive analysis conform to known reality as to how liberals, conservatives, or others tend to decide a certain controversial issue? For example, what criteria might be added and with what weights in order to understand why liberals are generally opposed to capital punishment and conservatives are generally in favor of it?

D. MISCELLANEOUS POINTERS FOR TEACHING SUBSTANCE WITH MICROCOMPUTERS

1. Provide each student with a copy of the program and the data files at the beginning of the semester. The students can then work with the program and the data more easily outside of class.
2. Before the end of the semester each student should submit a short paper analyzing a data file that he or she has created on a subject of particular interest. The preparation of short memos is even more important in a methods course than a substance course in order to encourage the students to apply the methods.
3. The midterm and final exam can be given in the same classroom where the microcomputers are available. The students may need to take the exam in shifts so that each student will have a microcomputer to use as an aid in developing the answers or analysis to each problem.
4. Questions whose answers involve the use of the microcomputers should be asked in class.
5. At least in the beginning of the semester the instructor should guide the students through the analysis of selected substantive problems by telling them collectively what they should type on the keyboard.
6. The instructor should frequently arrange for different parts of the class to be experimenting with different alternatives, criteria, or relations. For example, the liberal, independent, and conservative groups in the class can each separately clarify their respective positions and observe where the relatively weak elements are in the diverse conclusions. This helps develop mutually satisfying compromises.
7. Give homework assignments in which the students are expected to create and

Policy Evaluation

- manipulate data files. The data files can then be examined and discussed in class.
8. The students can be encouraged to experiment with the data files that are being discussed while the class is being conducted and raise questions and make comments concerning their experiences. It is a sign of interest if all the computers are working while the instructor is talking, assuming that the computers are working on class-related activities.
 9. At the beginning of each class session, announce about two or three data files that will be discussed. Doing so (1) prepares the students for the day's activities, (2) keeps the class more on schedule, (3) informs the students what will be covered at the beginning of the next class period in light of what is left over, and (4) encourages more processing of the data files rather than traditional lecturing.
 10. The students learn better in class if they have a combination of (1) working with microcomputers at their seats and (2) observing the microcomputer display on a monitor or a projector that is under the control of the instructor. The use of a projector can, however, be a distraction from the more useful experience involved in working with the microcomputers.
 11. Data files should occasionally be developed with the class by collectively working out the alternatives, criteria, relations, and analysis. An exciting but risky approach is to do this spontaneously without having worked out the inputs in advance. That way the students get the benefits of participating in the development of new knowledge, rather than just absorbing what has been developed.
 12. In discussing data files, it is helpful to have the monitor or projector show the display that relates the alternatives to the criteria so that the students can see the big picture, rather than focus on just the alternatives or just the criteria.
 13. If pressed for time in discussing a data file, go quickly to the initial bottom-line analysis and to the threshold or break-even analysis, rather than dwell on the preliminary inputs.
 14. There is no need to give a term paper assignment in order to provide the students with experience in using microcomputers for analyzing public policy or political science problems. They will be getting plenty of that in the classroom if microcomputers are used to teach public policy or political science substance.
 15. Learning experience can be enhanced by grouping the students in terms of their subject interests. These interests can be determined at the beginning of the semester. Each group of students can be seated behind the same microcomputer to facilitate their working together. The class can sometimes work on problems where each group applies general principles to its specific subject.
 16. Working with decision matrices enables the students to understand better (1)

how to make decisions, (2) how to explain decisions that other people make, (3) how to influence other people's decisions, and especially (4) to determine what the goals, alternatives, relations, and conclusions might be for any given substantive problem.

E. SOME CONCLUSIONS

The P/G% approach represents a synthesis between inductive and deductive reasoning and between empiricism and rationalism in the following ways:

1. The relations between policies and goals tend to be based on experience, although generally not statistical correlation analysis.
2. The overall analysis is deductive, with a prescriptive conclusion being deduced from premises that relate to goals, alternatives, and relations.
3. The goals and alternatives are often based on surveying decision-makers or their writings, which is a form of empirical analysis.
4. The relations, goals, and alternatives may sometimes be deduced from propositions about other relations, goals, and alternatives.
5. Perhaps most important is the emphasis that P/G% places on sensitivity analysis. That kind of analysis does not tell us what the truth is. It may, however, tell us that there is no difference on the bottom line of what policy to adopt whether the truth is a 64 or an 83 with regard to a controversial relation or goal weight. In other words, we need empiricism to tell us what the reasonable ranges are for the input scores, but deduction tells us at what point within those ranges a score crosses a threshold of changing the results.

By way of (its emphasis on sensitivity analysis, the P/G% microcomputer approach to teaching methods or substance stimulates a student to be constantly thinking about such matters as adding, subtracting, or otherwise changing (1) the alternatives, (2) the minimums and maximums on the alternatives, (3) the goals, (4) the measurement units for the goals, (5) the weights of the goals, (6) minimums and maximums on the goals, (7) the relations, (8) the budget to be allocated if there is one, and (9) any other inputs that are subject to change. That kind of stimulation is conducive to creativity as well as to careful reasoning. Colleges need to do more with regard to encouraging creativity and careful reasoning. If the P/G% approach does have that effect, then it is indeed a useful method for teaching political science and public policy.

The investment of time and money in order to receive the benefits of teaching with microcomputers is low compared to the benefits obtainable. To paraphrase a policy analyst of the 1800s, researchers and teachers have nothing to lose but their less efficient research and teaching methods. They should unite in sharing experience

and software for the good of the public policy profession and the improvement of public policy decisions.

F. REFERENCES

On using microcomputers as a teaching tool, see Kenneth Kraemer, "Curriculum Recommendations for Public Management Education in Computing" 46 *Public Administration Review*, 595-602 (1986); Herbert E Weisberg, "Microcomputers in Political Science," *News for Teachers of Political Science*, Summer 1983, pp. 7, 7; Fiona Chen, "Teaching Computer Application in Public Administration" (Eastern Washington University, School of Public Affairs, 1984); and S. Nagel, "Microcomputers and Public Policy Analysis," in Don Calista (ed.), *Microcomputers and Public Productivity* (Special issue of the *Public Productivity Review*, 7 1985).

On the P/G% software, see S. Nagel, "P/G% Analysis: A Decision-Aiding Program," *Social Science Microcomputer Review* 3 (1985): 243; and S. Nagel, *Evaluation Analysis with Microcomputers* (Westport, Conn.: Greenwood Press, 1988).

Background material exists on teaching from a perspective of societal goals to be achieved, policy alternatives for achieving them, and relations between goals and alternatives in order to choose the best alternative, combination, or allocation. See, for example, the opening chapter in S. Nagel, *Public Policy: Goals, Means, and Methods* (New York: St. Martin's, 1984), and the Instructor's Manual that accompanies that book. Also see S. Nagel, "Using Microcomputers and P/G% for Teaching Policy Analysis and Public Policy," in Peter Bergerson and Brian Nedwek (eds.), *Teaching Public Administration* (St. Louis: St. Louis University, 1985).



PRIMEROS PASOS EN TRANSFORMACIONES LINEALES

Marta G. Caligaris, Georgína 5. Rodríguez y Roberto E. Caligaris*

Universidad Tecnológica Nacional
Facultad Regional San Nicolás
Grupo de Informática Educativa
Colón 332 (2900) San Nicolás
TEL.: (0461) 20620 / 25266
FAX: (0461) 20630
e-mail: recaliga@cablenet.com.ar
rec@utnsn.edu.ar

Cuando enseñamos funciones en una variable, dedicamos mucho tiempo al estudio de la gráfica de la función. Las gráficas de funciones de dos variables son más complicadas y sin embargo seguimos dibujando las superficies resultantes, cuando es posible.

En los cursos de álgebra lineal introducimos un nuevo tipo de función: las transformaciones lineales. Estudiamos las propiedades algebraicas de estas funciones pero no nos dedicamos a las propiedades geométricas.

Presentamos aquí una introducción para este capítulo del álgebra lineal, apoyándonos tanto en la capacidad gráfica como en la de programación que posee **Mathematica**.

Para trabajar en lo que sigue, el alumno no necesita dominar el programa ni, mucho menos, saber programar. Esa tarea es del profesor: definimos funciones en lenguaje habitual con parámetros sencillos para que el alumno utilice, sin analizar cuál es su estructura. Nos interesa especialmente estimular la intuición del alumno para que pueda luego entender los conceptos matemáticos abstractos.

Aclaremos antes de empezar que todo lo que fue realizado en **Mathematica** aparecerá con el tipo de letra Courier New. Lo presentamos de esta manera para distinguirlo en el texto. Los nombres de cada comando, función o variable que definimos utilizan minúsculas para evitar superposiciones con las definiciones del soft. Además, cuando la respuesta dada por **Mathematica** es corta, la mostramos en la misma línea del comando, separada por =>.

INTRODUCCION

Recordemos la definición de transformaciones lineales:

Una función $T: V \rightarrow W$ se dice una transformación lineal si
 $\forall a, b \in V, \forall k \in K$ (cuerpo de escalares se tiene :

$$T(a + b) = T(a) + T(b)$$

$$T(k a) = k T(a)$$

Si $T: V \rightarrow W$ es una transformación lineal, el espacio vectorial V se llama dominio de T y el espacio vectorial W codominio de T .

A partir de la definición analicemos si las siguientes funciones son transformaciones lineales:

$$T_1: \mathbb{R}^2 \rightarrow \mathbb{R}^3 / \forall a \in \mathbb{R}^2 : T_1(a_1, a_2) = (a_1 + a_2, a_1 - a_2, a_2)$$

$$T_2: \mathbb{R}^2 \rightarrow \mathbb{R}^2 / \forall a \in \mathbb{R}^2 : T_2(a_1, a_2) = (-a_2, a_1)$$

$$* T_3: \mathbb{R}^3 \rightarrow \mathbb{R}^2 / \forall a \in \mathbb{R}^3 : T_3(a_1, a_2, a_3) = (a_1 + 2a_2, a_3)$$

$$* T_4: \mathbb{R}^2 \rightarrow \mathbb{R}^2 / \forall a \in \mathbb{R}^2 : T_4(a_1, a_2) = (a_1/2 + a_2, a_1 + 2a_2)$$

* Miembro de la Carrera del Investigador Científico del CONICET

Para trabajar con Mathematica escribiremos las funciones en su lenguaje:

```
Off[General::spell1];(* Para evitar mensajes rojos de error si usamos
palabras similares a otras ya existentes*)

T1 C [a1-, a2-)]-{a1 + a2, a1 - a2, a2);
T2 t [a1-, a2-)]-f-a2, ' a1);
T3 C [a1-, a2-, a3-)]-{a1 + 2, a3);
T4 E {a1-, a2-)]-{a1/2 + a2, a1 + 2 a2);
```

Definimos también dos elementos arbitrarios de \mathbb{R}^2 :

```
b={b1,b2); c={c1,c2);
```

y dos de \mathbb{R}^3 :

```
d={d1,d2,d3); e={e1,e2,e3);
```

para probar la definición de transformaciones lineales. Así, para cada T_i :

```
T1[b]+T1[c]==T1[b+c]           => True
T1[k b]-==Expand[k 8T1[b]]     => True

T2[b]+T2[c]==T2[b+c]           => True
T2[k b]-==Expand[k'T2[b]]      => True

T3Cd]+T3[e]-==Expand[T3[d+e]]  e False
T3[k d]-==Expand[k'T3[d]]      e False

T4[b]+T4[o]-==Expa~d[T4[b+c]]  r3 True
T4[k b]-==Expand[k T4[b]]      e True
```

El alumno puede desarrollar ambos miembros de las igualdades y ver si son o no iguales

Vemos que T_1, T_2 y T_4 son transformaciones lineales y que T_3 no es una transformación lineal.

Podemos ver también que la condición necesaria (no suficiente) para que una transformación sea lineal : $T(0) = O_w$ no se cumple para T_3 .

```
T3C{0,0,0}==Expand[T3[0]]      *      12, 01
```

IMAGEN DE UNA TRANSFORMACION LINEAL

Recordemos que:

La imagen de una transformación lineal $T: V \rightarrow W$ es el subespacio de W :

$Im(T) = \{ y \in W / \exists x \in V : T(x) = y \}$

Para ver el efecto de distintas transformaciones, con dominio y codominio en \mathbb{R}^2 , vamos a 'pintar' el plano con diferentes colores. Para ello, elegimos puntos aleatorios en cada uno de los cuatro cuadrantes del plano $x - y$:

```
primercuadrante=Table[Random[ 1, Rand?mE ||,f200]];
segundocuadrante=Table[Cf-Random[ ],Random[1: 1),{200}];
tercercuadrante=Tz@le[{-Random[ ],-RandomE 1),{200}];
cuartocuadrante=-T+le[Random[ ],-Random[ 1),~200~];
```

Ahora debemos graficar las listas de puntos obtenidas (que no vemos porque escribimos ; al final). Para ello usamos el comando `ListPlot`. Como de aquí en más lo vamos a utilizar muchas veces, cada vez con opciones de color diferente y el resto de las opciones sin cambio, definimos la función:

```
dibujo[conjunto-,colorJ:=ListPlot[conjunto, PlotStyle->{color,PointSize[.025]},
DisplayBuncCion->Identity,AspectRatio->Automatic]
```

Cada vez que usemos esta función debemos especificar qué conjunto de puntos queremos graficar y en qué color.

El color se puede especificar por `RGBColor [a, b, c]`. Para simplificar la tarea del alumno (y hacer que el argumento de la función sea claro para él) definimos los colores que vamos a usar con sus nombres en castellano. Expertos en el uso del programa podrán objetar esta simplificación, dado que existe el paquete `Graphics'Colors'` que al cargarlo permite llamar los colores por su nombre en inglés. El alumno podrá optar por la posibilidad que prefiera.

```
rojo=RGBColor[1,0,0]; verde=RGBColor[0,1,0];
azul=RGBColor[0,0,1]; amarillo=RGBColor[1,1,0];
naranja=RGBColor[1,0.5,0]; violeta=RGBColor[1,0,1];
eeste=RGBColor[0,0.5,1]; negro=RGBColor[0,0,0];
```

Para 'pintar' el dominio, usamos un color para cada cuadrante:

```
c1=dibujo[primercuadrante,rojo]; c2=dibujo[segundocuadrante,verde];
c3=dibujo[tercercuadrante,azul]; c4=dibujo[cuartocuadrante,amarillo];

r2=Show[{c1,c2,c3,c4},PlotLabel->R*]; (*este es el dominio:R**)
```

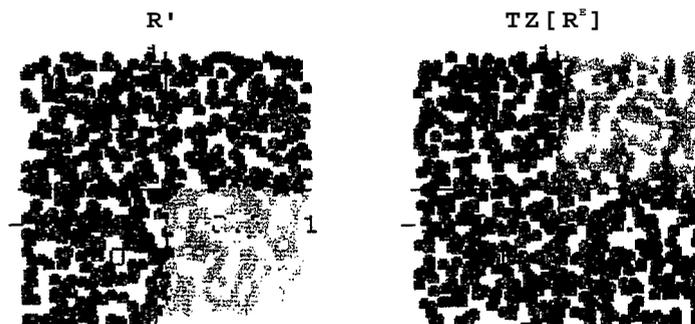
La idea ahora es aplicar las transformaciones lineales definidas previamente a este conjunto de puntos y ver qué ocurre, utilizando el mismo color para cada punto y su transformado. Para ello entonces definimos una nueva función:

```
transf[T-,conjunto ]:=Table[T[conjunto[[i]]],{i,1,Length[conjunto]?1}
```

y para graficar usamos `transf` con la función `dibujo`, previamente definida. Transformamos mediante T_2 :

```
c1transfT2=dibujo[transf[T2,primercuadrante],rojo];
c2transfT2=dibujo[transf[T2,segundocuadrante],verde];
c3transfT2=dibujo[transf[T2,tercercuadrante],azul];
c4transfT2=dibujo[transf[T2,cuartocuadrante],amarillo];

imagenT2=Show[{c1transfT2,c2transfT2,c3transfT2,c4transfT2},PlotLabel->T2 [R^2]];
Show[GraphicsArray[{r2,imagenT2}]]:
```



La imagen de T_2 es, como puede verse en el gráfico, el espacio R^2 .

Esta transformación $[T(x, y) = (-y, x)]$ es una rotación de 90° en sentido antihorario.

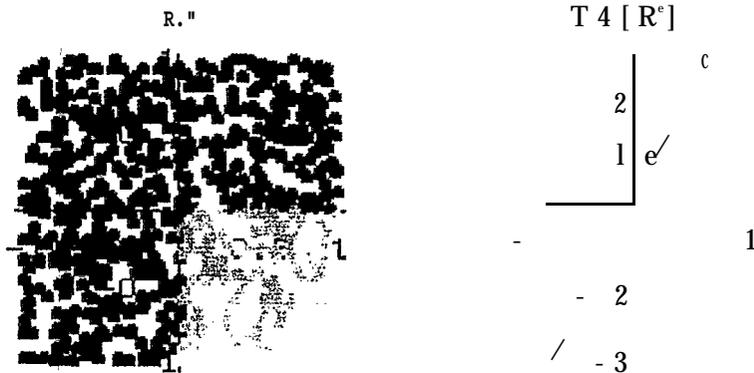
Para T_4 , usamos hs mismas funciones:

```

c1transfT4=dibujotransf[T4,primercuadrante],rojo];
c2transfT4=dibujotransf[T4,segundocuadrante],verde];
c3transfT4=dibujotransf[T4,tercercuadrante],azul];
c4transfT4=dibujotransf[T4,cuartocuadrante],amarillo];

imagenT4=Show[{c1transfT4,c2transfT4,c3transfT4,c4transfT4},PlotLabel->T4 [R^2]];

Show[GraphicsArray[{$r2,imagenT4}]];
    
```



La imagen de esta transformación es una recta. ¿Cómo podemos obtener su ecuación? Sabemos que $(0, 0)$ es un punto de la recta (porque siempre O_w pertenece a la imagen de T); para obtener las coordenadas de otro:

$$T_4[\{1, 1\}] \Rightarrow \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

Entonces la ecuación de la recta es : $y = 2x$, y la imagen: $\text{Im}(T_4) = \{ (x, y) / y = 2x \}$

NUCLEO DE UNA TRANSFORMACION LINEAL

Recordemos que:

El nuclea de una transformación lineal $T: V \rightarrow W$ es el subespacio de V :

$\text{Ker}(T) = \{ X \in V / T(x) = O_w \}$

El primer intento para encontrar el nucleo será gráfico. Analizaremos las transformaciones T_1 y T_4 , cuyos dominio y codominio son \mathbb{R}^2 .

Como el nucleo es un subespacio del dominio, para estas transformaciones las posibilidades son:

$$\begin{aligned}
 &\Leftrightarrow \{(0, 0)\} \\
 &\Rightarrow \text{una recta que pase por el origen} \\
 &\Rightarrow \mathbb{R}^2
 \end{aligned}$$

ya que ningun otro conjunto es subespacio de \mathbb{R}^2 .

Definiremos entonces, un conjunto de rectas que pasan por el origen y analizaremos el efecto de aplicarles T_2 y T_4 .

```

recta1=Table[{0,y},{y,-3,3,0.2}];
recta2=Table[{-y,y/2},{y,-3,3,0.2}];
recta3=Table[{x,0},{x,-3,3,0.2}];
recta4=Table[{-y,2y},{y,-3/2,3/2,0.15}];
recta5=Table[{y,y/2},{y,-3,3,0.2}];
recta6=Table[{y,2y},{y,-3/2,3/2,0.15}];
recta7=Table[{-y,y},{y,-3,3,0.2}];
recta8=Table[{y,y},{y,-3,3,0.2}];

r1=dibujor[recta1,rojo];      r2=dibujor[recta2,azul];
r3=dibujor[recta3,verde];    r4=dibujor[recta4,amarillo];
r5=dibujor[recta5,naranja];  r6=dibujor[recta6,violeta];
r7=dibujor[recta7,celeste];  r8=dibujor[recta8,negro];

```

```

haz=Show[{r1,r2,r3,r4,r5,r6,r7,r8},PlotLabel->"Haz"];

```

```

TPr1=dibujor[transf[T2,recta1],rojo];  T2r2=dibujor[transf[T2,recta2],azul];
T2r3=dibujor[transf[T2,recta3],verde];  T2r4=dibujor[transf[T2,recta4],amarillo];
T2r5=dibujor[transf[T2,recta5],naranja]; T2r6=dibujor[transf[T2,recta6],violeta];
T2r7=dibujor[transf[T2,recta7],celeste]; T2r8=dibujor[transf[T2,recta8],negro];

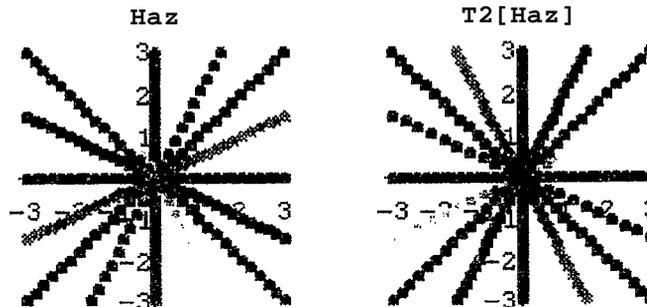
```

Como se puede ver, cada recta y su transformada son del mismo color.

```

imagenT2haz=Show[{T2r1,T2r2,T2r3,T2r4,T2r5,T2r6,T2r7,T2r8},PlotLabel->T2 [Haz]];
Show[GraphicsArray[{haz,imagenT2haz}]]];

```



¿Será que (0, 0) sólo se obtiene como T(0, 0)? Eso parece. Para estar seguros, calculamos analíticamente. Para ello definimos una función *nucleo*, válida sólo para este caso:

```

nucleo[T1]:=Solve[T[{x,y}]=={0,0},{x,y}>(*R2 en R^2*)

```

```

nucleo[T2]      el      {{x -> 0, y -> 0}}

```

Ahora no hay dudas: $\text{Ker}(T_2) = \{(0, 0)\}$. Para T_4 , trabajamos igual:

```

T4r1=dibujor[transf[T4,recta1],rojo];
T4r2=dibujor[transf[T4,recta2],azul];
T4r3=dibujor[transf[T4,recta3],verde];
T4r4=dibujor[transf[T4,recta4],amarillo];
T4r5=dibujor[transf[T4,recta5],naranja];
T4r6=dibujor[transf[T4,recta6],violeta];
T4r7=dibujor[transf[T4,recta7],celeste];
T4r8=dibujor[transf[T4,recta8],negro];

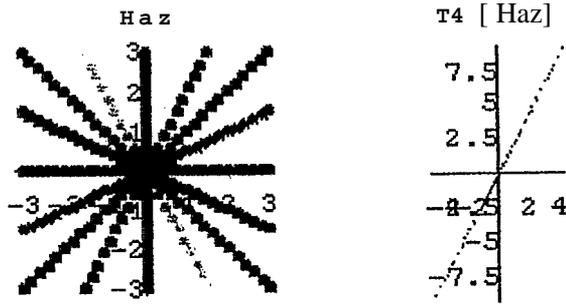
```

```

imagenT4haz=Show[{T4r1,T4r2,T4r3,T4r4,T4r5,T4r6,T4r7,T4r8},PlotLabel->T4 [Haz]];

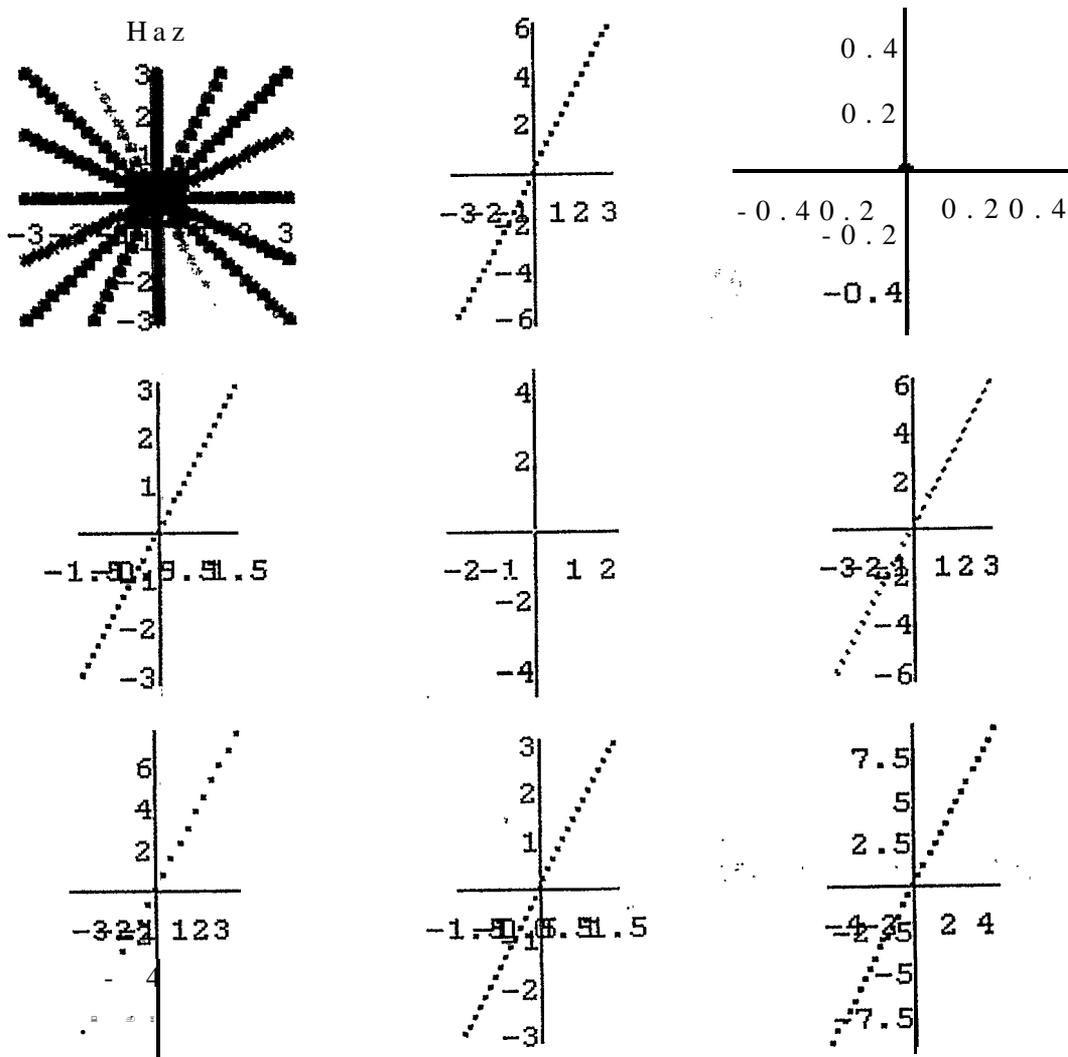
```

```
Show[GraphicsAm&[{haz, imagenT4haz}]];
```



Notamos que ocurre algo diferente con respecto al caso anterior, pero así no nos da mucha información. Probemos de otra manera:

```
Show[GraphicsArray[{{haz, T4r1, T4r2}, {T4r3, T4r4, T4r5}, {T4r6, T4r7, T4r8}}]];
```



Evidentemente la recta2 es el nucleo de T_4 . También podemos obtenerlo analíticamente:

```
nucleo [T4] = > {Ix -> -2 }
```

Entonces $\text{Ker}(T_4) = \{(x, y) / x = -2y\}$

Aunque las gráficas se complican, es muy fácil modificar la función `nucleo` anterior para usarla en transformaciones cuyos dominio o codominio no sean \mathbb{R}^2

```
nucleo2 [T-I :=Solve[T[(x,y)]==(0,0,0),(x,y)]J(*otro codominiof)
```

```
nucleo2 [T1] => {Ix -> 0, y -> 0}
```

El núcleo de T1 es $\text{Ker}(T) = \{(0, 0)\}$.

MATRIZ DE UNA TRANSFORMACION

Recordemos que:

Cualquier transformación lineal $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$ puede representarse mediante una matriz A tal que:

$$T(x) = Ax \quad \forall x \in \mathbb{R}^n$$

La matriz A dependerá de las bases elegidas para \mathbb{R}^n y \mathbb{R}^m

Si bien las transformaciones lineales pueden estudiarse sin hacer referencia alguna a las bases de los espacios dominio y codominio, un cálculo efectivo de las mismas exige el conocimiento de dichas bases. Una transformación lineal $T: V \rightarrow W$ queda determinada cuando se conocen una base ordenada de V, una base ordenada de W, y los transformados de la base V en la base W.

Considerando que la base en cada uno de los espacios, es la base canónica correspondiente, obtendremos la matriz de cada una de las transformaciones lineales dadas al inicio, recordando que la primera columna es el transformado del primer vector de la base V en la base W, la segunda columna es el transformado del segundo vector de la base de V en la base de W, etc. Entonces:

a) $T_1: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ $T_1(a_1, a_2) = (a_1 + a_2, a_1 - a_2)$

```
matrizt1=Transpose[{T1[{1,0}],T1[{0,1}]}];
MatrixForm[matrizt1]
```

```
1  1
1 -1
0  1
```

b) $T_2: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ $T_2(a_1, a_2) = (-a_2, a_1)$

```
matrizt2=Transpose[{T2[{1,0}],T2[{0,1}]}];
MatrixForm[matrizt2]
```

```
0 -1
1  0
```

c) T_3 no es una transformación lineal.

d) $T_4: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ $T_4(a_1, a_2) = (a_1 + 2a_2, a_1 + 2a_2)$

```
matrizt3=TransposrC~T4[:(:1,0)],T4[{0,1}]]];
MatrixForm[matrixzf3]
```

```
1
  1
2
  2
1  2
```

Mostramos para T2, la igualdad $T(x) = A x$

```
T2 Eix,yI1      =>      (-y, x)
matrixt2 . {x,y} =>      (-y, x)
```

ALGUNOS EJEMPLOS INTERESANTES

Presentamos a continuación algunos ejemplos que ayudarán a “descubrir” las propiedades geométricas de las transformaciones lineales de \mathbb{R}^2 en \mathbb{R}^2 . Elegimos como conjunto de partida una circunferencia de radio 1, centrada en el origen. Definiremos esta circunferencia como listas de puntos, así:

```
primerarco=Table[{Cos[t],Sin[t]},{t,0,Pi/2,Pi/30}];
segundoarco=Table[{Cos[t],Sin[t]},{t,Pi/2,Pi,Pi/30}];
tercerarco=Table[{Cos[t],Sin[t]},{t,Pi,3Pi/2,Pi/30}];
cuartoarco=Table[{Cos[t],Sin[t]},{t,3Pi/2,2Pi,Pi/30}];
```

Para ver mejor el efecto de cada transformación, cada cuarto de circunferencia será de un color diferente. Utilizamos la función dibujo:

```
arcol=dibujo[prim+arco,rojo];          arco2=dibujo[segundoarco,azul];
arco3=dibujo[tercfrarco,verde];       arco4=dibujo[auartoarco,amarillo];
cirounf@rsncia=Show[ {araol,~rco2,arco3;arco4},PlotL~l-> "Circunf~rencia"1;
```

Como vamos a trabajar con transformaciones definidas por su correspondiente matriz, la función **transf** que utilizamos antes no nos sirve (trabaja con la ley de la transformación). Definimos, entonces, una nueva función:

```
mtransf [matriz-, conjunto ]:=Table[matriz.aconjunto[[i]],{i.,1,~ngth[conjunto]1}
```

Esta función necesita, como argumento, la matriz de la transformación y el conjunto de partida.

Con las funciones **dibujo** y **mtransf** analizamos las transformaciones lineales cuyas matrices son:

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

+ Ejemplo 1: matriz A

```
a={{Or-11,~lrOI};
```

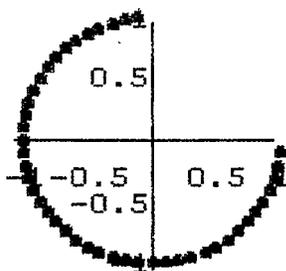
```
arcol.ta=di.bujo[mtransf[a,primerarco],rojol;
arco2ta=dibujo[mtransf[a,segundoarao],azull;
arco3ta--dibjo[mtransf[a,tercerarco],verdel;
arao4ta=di.bujo[mtransf[a,cuartoarco3,arillol;
```

```
imac3nTa=Show[{arcolta,arco2ta,arco3ta,arao4ta},PlotLslbel->~Transformación A];
```

```
Show[GraphicsArrayf(circunferencia,imagenTa)];
```

Circunferencia Transformación A

-0.5 0.5 .1.
-0.5



En este gráfico se ve que el efecto de la transformación A cuya matriz es:

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

es una rotación, de 90°, en sentido antihorario.

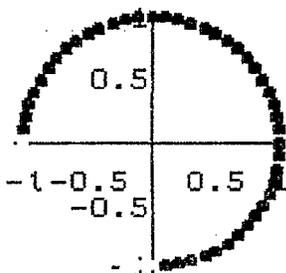
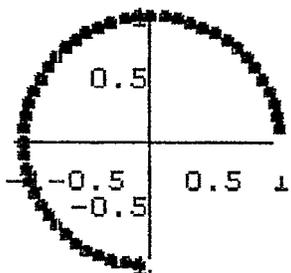
+ Ejemplo 2: matriz B

b={{-1,0},{0,1}};

```
arcoltb=dibujomtransf[b,primerarco],rojo];
arco2tb=dibujomtransf[b,segundoarco],azul];
arco3tb=dibujomtransf[b,tercerarco],verde];
arco4tbdibujomtransf[b,cuartoarco],amarillo];
```

```
imagenfb=Show(arcoltb,arco2tb,arco3tb,arco4tb),PlotLabel->'';
Show[GraphicsArray[{circunferencia,imagenTb}]]
```

Circunferencia Transformación B



En este gráfico se ve que el efecto de la transformación B cuya matriz es:

$$B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

es una reflexión en el eje y.

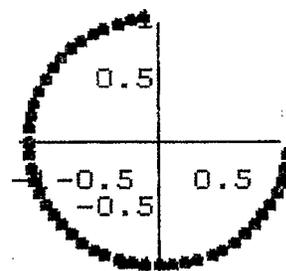
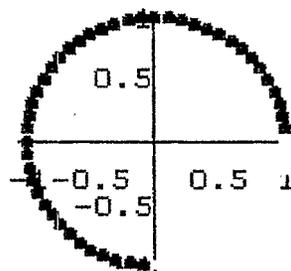
+ Ejemplo 3: matriz C

c={{1,0},{0,-1}};

```
arcoltc=dibujomtransf[c,primerarco],rojo];
arco2tc=dibujomtransf[c,segundoarco],azul];
arco3tc=dibujomtransf[c,tercerarco],verde];
arco4tc=dibujomtransf[c,cuartoarco],amarillo];
```

```
imagenTc=Show(arcoltc,arco2tc,arco3tc,arco4tc),PlotLabel->'';
Show[GraphicsArray[{circunferencia,imagenTc}]];
```

Circunferencia Transformación C



En este gráfico se ve que el efecto de la transformación C cuya matriz es:

$$C = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

es una reflexión en el eje x.

En los gráficos anteriores vemos el conjunto de partida (circunferencia) y el resultado de aplicarle las tres transformaciones propuestas. En cada una de los casos cada punto tiene el mismo color que su imagen.

Ahora, analizaremos las transformaciones cuyas matrices son:

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 1,5 \end{pmatrix}$$

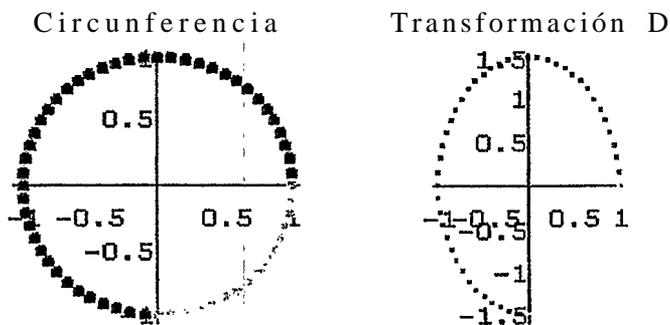
$$E = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$$

+ Ejemplo 4: matriz D

```
d={{1,0},{0,1.5}};
```

```
arcoltd=dibujomtransf[d,primararco],rojo1;
arco2td=dibujomtransfCd,sagundoaraoj,azull;
arao3td=dibujomtransf[d,terosraraol,vl;
arco4td=dzi.bujo [mtransfCd,auartoarco] ,amarill.ol;
~genTd=Show[ (ar~oltd,arco2td,aroc3td,arco4td) ,PlotL~1->~Transformáció~,D"1;
```

```
Show[GraphicsArray[{cxircunferencia,imagenTc}]];
```



En este gráfico se ve que el efecto de la transformación D cuya matriz es:

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 1,5 \end{pmatrix}$$

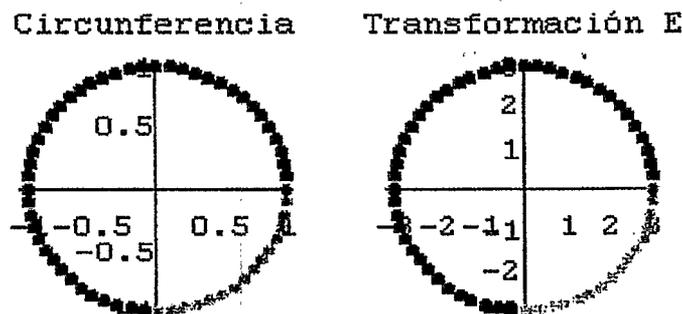
es un estiramiento.

+ Ejemplo 5: matriz E

```
e={{3,0},{0,3}};
```

```
arcolte=dibujomtransf[e,primerarco],rojo;
arco2te=dibujomtransf[e,segundoarco],azul;
arco3te=dibujomtransf [e,tercerarco] ,verde;
arco4te=dibujomtransf~e,cuartoarco],amarillo];
imagenTe=Show[ (arcolte,arco2te,arco3te,arco4te) ,PlotLabel->~Transfo~ción E"1;
```

```
Show[GraphicsArray[{circunferencia,imagenTd}]]
```



Cuidado!! No son iguales. ¿Se ve por qué?

En este gráfico se ve que el efecto de la transformación E cuya matriz es:

$$E = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$$

es un estiramiento.

Los ejemplos anteriores nos permiten confirmar que se cumple la siguiente propiedad:.,

Una transformación lineal preserva la longitud si y sólo si su matriz es ortogonal

OPERACIONES CON TRANSFORMACIONES LINEALES

*, Suma de transformaciones lineales

Dadas dos transformaciones lineales $T_1, T_2 : V \rightarrow W$ se llama transformación suma a la transformación lineal definida por:

$$(T_1 + T_2)(x) = T_1(x) + T_2(x) \quad \forall x \in V$$

Se demuestra que fijadas las bases de V y W , si las matrices de las transformaciones T_1 y T_2 son respectivamente M_1 y M_2 , la matriz de la transformación suma $(T_1 + T_2)$ es la matriz $M_1 + M_2$.

Para obtener la matriz de la transformación $C+D$ basta con sumar las matrices c y d :

`suma=c+d;`

`MatrixForm[suma]`

`2 0`

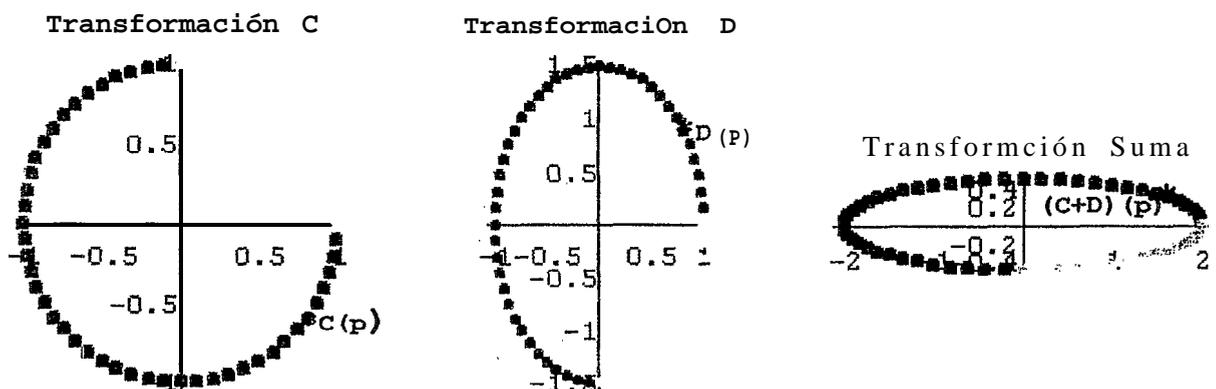
`0 0.5`

Vamos a transformar la circunferencia original mediante $C+D$, para luego mostrar las imágenes mediante C , D y $C+D$. Las dos primeras ya fueron definidas. Los comandos que siguen realizan la última. Como siempre, se mantiene la relación de color entre cada punto y su imagen.

```
arcolsuma=dibujomtransf [suma,primerarco],rojo];
arco2suma=dibujomtransf [suma,segundoarco],azul];
arco3suma=dibujomtransf [suma,tercerarco],verde];
arco4suma=dibujomtransf [suma,cuartoarco],amarillo];

imagenTsumna=Show[{arcolsuma,arco2suma,arco3suma,arco4suma},
PlotLabel->'Transformación Suma' 1];

Show[GraphicsArray[{imagenTc,imagenTd,imagenTsumna} 1];
```



¿Cómo interpretamos este gráfico? Para analizarlo hemos marcado en cada caso la imagen del mismo punto p del dominio con un *. Al sumar las coordenadas de $C(p)$ y $D(p)$ se obtienen las coordenadas de $C+D(p)$.

+ Producto por un escalar

Dada una transformación lineal $T : V \rightarrow W$ y un escalar k se llama producto de k por T a la transformación:

$$(k \cdot T)(x) = k \cdot T(x) \quad \forall x \in V$$

Se demuestra que fijadas las bases de V y W , si la matriz de la transformación es M , la matriz de la transformación producto $(k \cdot T)$ es la matriz $k \cdot M$.

Obtenemos la matriz de la transformación $(k' A)$, utilizando la transformación A ya definida, para los valores de $k = 1.3$ y $k = 0.7$:

```

kmayoruno=1.3 a          +      I{0, -1.31, 11.3, 011
kmsnoruno=0.7 a         +      {I0, -0.71, 10.7, 011

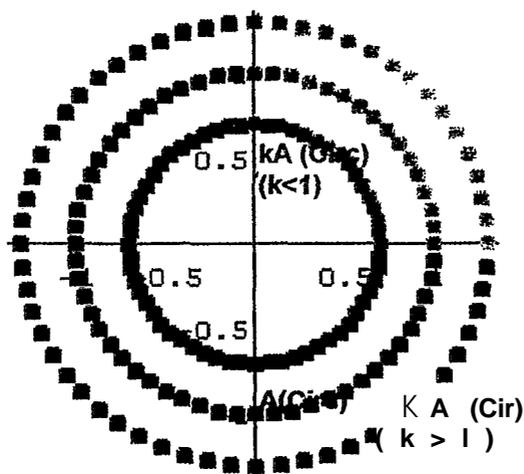
arColmay=dibujo[mtransf[kmayoruno,primerarco],rojo];
arco2may=dibujo[mtransf[kmayoruno,segundoarco],azul];
arco3may=dibujo[mtransf[kmayoruno,tercerarco],verde];
arco4may=dibujo[mtransf[kmayoruno,cuartoarco],amarillo];

imagenTmayor1=Show[arColmay,arco2may,arco3may,arco4may];

arColmen=dibujo[mtransf[kmenoruno,primerarco],rojo];
arco2men=dibujo[mtransf[kmenoruno,segundoarco],azul];
arco3men=dibujo[mtransf[kmenoruno,tercerarco],verde];
arco4men=dibujo[mtransf[kmenoruno,cuartoarco],amarillo];

imagenTmenor1=Show[arColmen,arco2men,arco3men,arco4men];

Show[{imagenTa,imagenTmayor1,imagenTmenor1},PlotLabel->"",
      DisplayFunction->$DisplayFunction];
    
```



+ Composición de transformaciones lineales

Dadas dos transformaciones lineales $T_1: S \rightarrow W$ y $T_2: V \rightarrow S$ se llama composición o producto a la transformación lineal definida por:

$$(T_1 \circ T_2)(x) = T_1[T_2(x)] \quad \forall x \in V$$

Se demuestra que fijadas las bases de V , S y W , si las matrices de las transformaciones T_1 y T_2 son respectivamente M_1 y M_2 , la matriz de la composición $(T_1 \circ T_2)$ es la matriz $M_1 \cdot M_2$.

Mostramos en un ejemplo que la composición no es conmutativa.

```

prodi-a.d          => (( 0, -1.5), (1, 0.))
otroprod=d.a      => {{0, -1}, {1.5, 0.}}

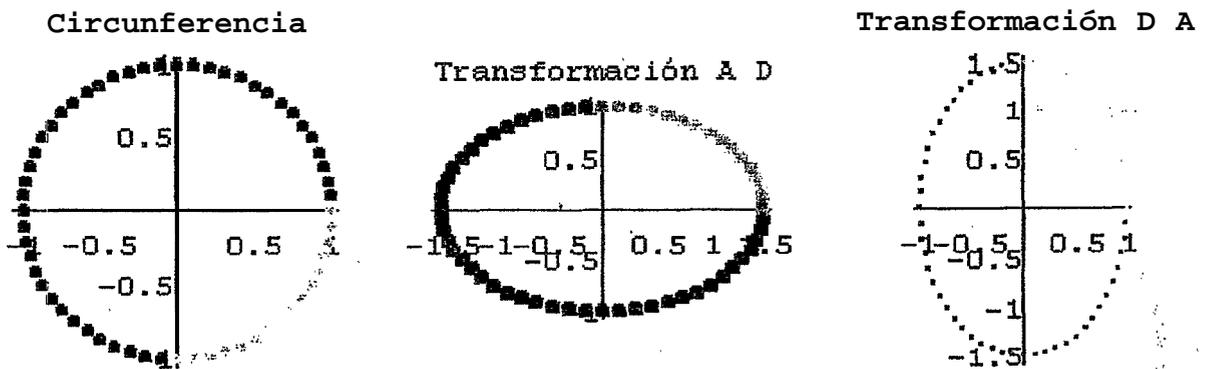
arcolprod=dibujofmtransf [prod,primerarcol,rojo];
arco2prod=dibujomtransf[prod ,segundoarcol,azul];
arco3prod=dibujomtransf[prod ,tercerarcol,verde];
arco4prod=dibujomtransf[prod,cuartoarcol,amarillo];

imagentprod=Show[~arcolprod,arco2prod,arco3prod,arco4prod],
                PlotLabel->'~Transformación A D'];

arcolotroprod=dibujomtransf[otroprod,primerarcol,rojo];
arco2otroprod=dibujomtransf[otroprod,segundoarcol,azul];
arco3otroprod=dibujomtransf[otroprod,tercerarcol,verde];
arco4otroprod=dibujomtransf[otroprod,cuartoarcol,amarillo];

imagentotroprod=Show[arcolotroprod,arco2otroprod,arco3otroprod,arco4otroprod],
                    PlotLabel->'Transformación D A'];

Show[GraphicsArray[~circunferencia,imagenprod,imagenotroprod],
     PlotLabel->'~Transformación A D'];
    
```



Recordemos que la transformación A es una rotación y D un estiramiento en la dirección del eje y . Al realizar el producto $A \circ D$ primero se estira y luego se rota: por eso el resultado final es un estiramiento en la dirección del eje x .

En el caso de $D \circ A$, primero se rota, luego se estira en la dirección del eje y . Entonces el resultado final es un estiramiento en la dirección del eje y , como se puede ver en el gráfico.

* Inversa de una transformación

Dada una transformación lineal $T : V \rightarrow W$ inyectiva se llama inversa de T a la transformación lineal:

$$T^{-1} : \text{Im}(T) \rightarrow V / T \circ T^{-1} = I_{\text{Im}(T)} \quad \text{y} \quad T^{-1} \circ T = I_V$$

Se demuestra que fijadas las bases de V y W , si la matriz de la transformación es M , la matriz de la transformación inversa es la matriz inversa de M .

Trabajamos, como ejemplo, con la transformación A .

```
a      =>      ((0, -1, -1, 0, 1, 1)
inv=Inverse[a]      =>      ((0, 11, 1-1, 0, 11)
```

El conjunto de partida para la inversa de la transformación A es la imagen de la circunferencia original mediante A .

Para definir el nuevo conjunto de partida:

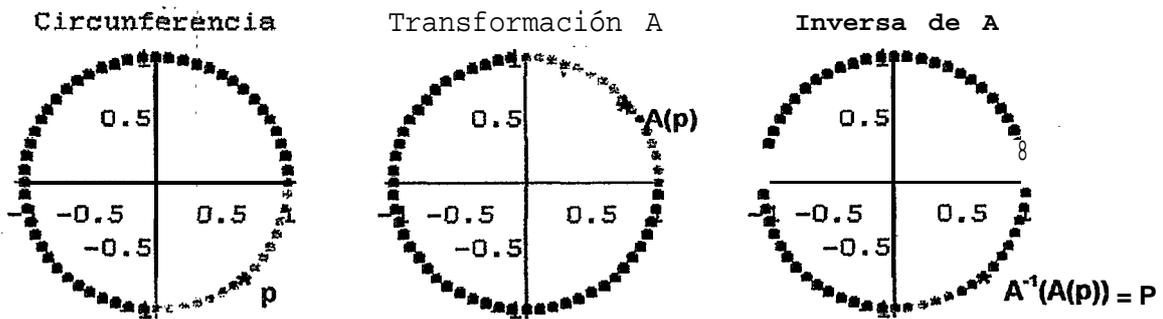
```
nuevoarcol=mtransf[a,primeraruoJ;
nuevoarco2atranpf[a,segundoarcoJ;
nuevoarco3==mtransf[a,terceraraoJ;
nuevoarco4==mtranf[a,cuartoarco];
```

Lo transformamos mediante A^{-1} :

```
arcolinv=dibujomtransf[inv,nuevoarcol],rojo];
arco2inv=dibujomtransf[inv,nuevoarco2],azul];
arco3inv=dibujomtransf[inv,nuevoarco3],verde];
arco4inv=dibujomtransf[inv,nuevoarco4],amarillo];

imageninversa=Show[{arcolinv,arco2inv,arco3inv,arco4inv},
PlotLabel-"YInversa de A"];

Show[GraphicsArray[{circunferencia,imagenTa,imageninversa}]];
```



Como antes, marcamos un punto p de la circunferencia original, su correspondiente transformado mediante A , $A(p)$, y el transformado de este último mediante A^{-1} , $A^{-1}(A(p))$. Volvemos a p !!

CONCLUSIONES

Hemos presentado lo que creemos es una nueva forma de introducir las transformaciones lineales en el aula. Ahora, con un software adecuado (en este caso **Mathematica**), podemos mostrar a los alumnos ejemplos que en el pizarrón **nunca** hubiésemos podido realizar. O alguna vez intentamos calcular y dibujar, punto por punto, la imagen de 800 puntos?

Estos ejemplos son sólo una guía. Cada uno de ellos se convierte, en el aula, en infinitos ejemplos que el alumno puede plantear por sí mismo, simplemente modificando algún parámetro. Así, cada alumno determina su propio ritmo de aprendizaje, "fabricando" la cantidad de ejemplos que necesite, asistido por el profesor.

Se han analizado transformaciones en el plano por el sólo hecho de su fácil visualización ya que queremos estimular la intuición del alumno con ejemplos concretos, para que le resulte más simple entender los conceptos matemáticos abstractos que involucra el tema. Los conceptos adquiridos con estos ejemplos gráficos, son fácilmente aplicables a transformaciones en otros espacios-

Finalmente puede aparecer en el alumno interesado (enhorabuena si son muchos!), la inquietud de fabricar sus propias funciones y/o ejemplos para otros casos. El profesor deberá estar permanentemente atento para auxiliar a sus alumnos y seguir el ritmo de cada uno; reconocemos que no es tarea fácil, pero creemos que es posible.

AGRADECIMIENTOS

Queremos agradecer al Dr. Juan José Manfredí de la Universidad de Pittsburgh, por sus comentarios y sugerencias durante el taller "Cuadernos interactivos de MATHEMATICA", realizado en la Universidad CAECE, Buenos Aires, en julio de 1996.

REFERENCIAS Y BIBLIOGRAFIA

Algebra lineal y sus aplicaciones", G. Strang, Addison-Wesley Iberoamericana.

"Mastering Mathematica. Programming Methods and Applications", J.W. Gray, Academic Press Inc. 1994

"Exploring Mathematics with Mathematica", T.W. Gray and J. Glynn, Addison-Wesley. 1991

"MATHEMATICA. A System for Doing Mathematics by Computer", S. Wolfram, Addison Wesley.

"Guide to Standard Mathematica Packages", Wolfram Research.

"New Tools for Linear Algebra", S. Hughes, E. Koopman, S. Prevost and B. Tesman; *Mathematica in Education*, Summer 1994.

"The Case Against Computer-s in K-13 Math Education", N. Koblitz; 77re *Mathematical Intelligencer*, 18 [1] (1996).

"Some Kinds of Computers for Some Kinds of Learning: A Reply to Koblitz", E. Dubinsky and R. Noss; *The Mathematical Intelligencer*, 18 [1] (1996).

NUEVAS HERRAMIENTAS EN LA ENSEÑANZA DE LA FÍSICA: CIRCUITOS ELÉCTRICOS

Roberto E. Caligaris#, Graciela A. Mansilla y Marta G. Caligaris
Universidad Tecnológica Nacional
Facultad Regional San Nicolás
Grupo Informática Educativa
c.c. 118 - 2900 San Nicolás
e-mail: recaliga@kablenet .com.ar
e-mail: rec@utnsn.edu.ar

Introducción

Existe a nivel internacional una verdadera preocupación por la falta de comprensión por parte de los estudiantes de los problemas que se plantean a nivel de cursos básicos. El trabajo de Sipcic [1] expone claramente esta preocupación y propone el camino para resolverlo mediante la utilización criteriosa de los nuevos programas computacionales

En trabajos previos [2-3] los autores han desarrollado diversos temas de la física elemental enfatizando la necesidad de definir rigurosamente los conceptos involucrados y utilizar programas computacionales para realizar los cálculos, que de otra manera resultarían sumamente tediosos, dejando tiempo y ánimo disponibles para la discusión de los principios fundamentales en estudio. El programa computacional utilizado es el denominado **Mathematica** [4], aún cuando no es el único disponible en la actualidad (como ejemplo, algunas ecuaciones de este trabajo se han resuelto utilizando **Maple**).

En esta presentación se continúa la línea trazada y se muestran los resultados que se obtienen al resolver las ecuaciones diferenciales que describen los circuitos eléctricos, para distintos casos.

Las ventajas de esta manera de presentar la problemática de la enseñanza de la física en los cursos básicos de las carreras de ingeniería y de ciencias son múltiples: a) permite la definición rigurosa de los conceptos desde el comienzo del curso; b) elimina el tedio del cálculo; c) posibilita incorporar un mayor número de temas en el mismo tiempo que antes se dedicaba a los cálculos; d) se puede, en consecuencia, incorporar y desarrollar temas que en la actualidad están reservados a los cursos de especialidad nada más que por cuestiones de tiempo.

Para concretar el objetivo de este trabajo se presenta inicialmente un caso simple y luego se estudia un circuito RLC en sus distintas alternativas, se plantea la ecuación diferencial que en cada caso corresponda y se pide al **Mathematica** que la resuelva. Finalmente se discute el resultado. Los autores desean enfatizar que se presentan puntos que normalmente pueden formar parte del programa de estudios del curso de física básica pero raramente alcanzan a ser dictados a lo largo del

Miembro de la Carrera del Investigador Científico del CONICET

período lectivo y que con el uso racional de los actuales programas computacionales se podría hacer.

También se puede preguntar el lector por qué circuitos eléctricos? Existen varias respuestas: Una es que las ecuaciones diferenciales que describen las diversas posibilidades que se pueden presentar no siempre se resuelven en los cursos básicos; el programa computacional que se utiliza permite resolver cualquiera de ellas, sin excepciones, y sacar el jugo a su contenido físico. Otra se toma del libro de Nilsson [5] cuando en el prefacio indica como motivación el deseo de asegurar mejor apoyatura a los estilos de aprendizaje de los estudiantes de la década de 1990. Se pretende mejorar esta afirmación diciendo que con el proyecto que los autores están encarando se apoyará mejor el estilo de aprendizaje de la segunda mitad de la década de 1990 y se salvarán las omisiones que los anteriores introducían.

Un ejemplo simple

Con el objeto de iniciar suavemente, y dentro del espacio disponible, se propone resolver el siguiente problema (tomado de Nilsson [5], ejemplo 7.2, pag.245):

Se aplica un pulso de voltaje $v(t) = 10 t \text{ Exp}(- 10 t)$ a un inductor de 1 00 mH. Calcular la respuesta del circuito y graficar $i(t)$ y $v(t)$. La condición inicial es $i(0) = 0$.

Se resuelve este problema indicando solamente los comandos que se utilizan, que más adelante se describirán en detalle, resaltando que siempre se utilizará el sistema internacional (SI) de unidades que el estudiante deberá tener en cuenta.

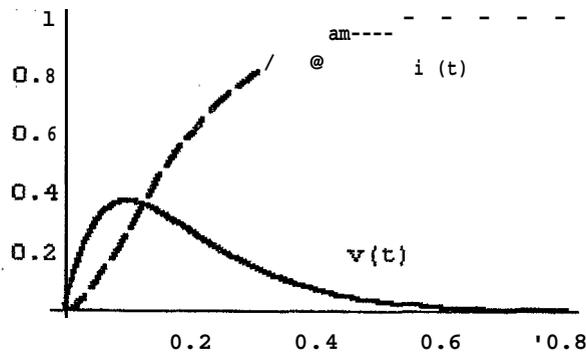
```
L=0.1; v[-tz-] = 10 t Exp(- 10 t);,
```

```
DSolve[{i'[b] - (1/L) v[t] == 0, i[0]==0},i[t],t]
```

Su solución y la representación gráfica solicitada siguen 8 continu&&n con los detalles dei comando utilizado:

```
{i[t] -> I1. + -1. - 10. t
10 t
E
```

```
Plot[{i[t]/.%,v[t]}, {t,0,0.83,
PlotStyle->{{Text["i(t)",{0.6,0.8531,Thickness[.0061,
Dashrig[.05,.02313,
{Text["v(t)",{0..5,0.23],ThicknessE.0061))1;
```



Las curva de rayas muestra la respuesta del circuito estudiado al voltaje aplicado, el que se muestra en línea llena

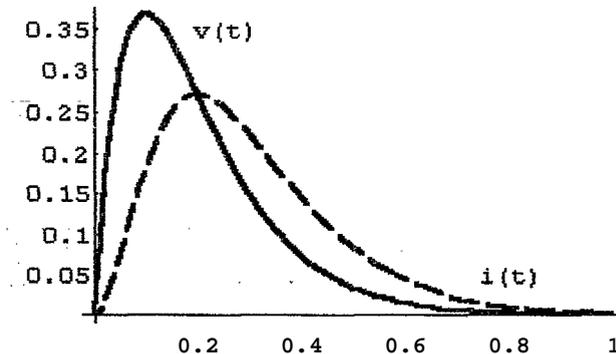
Epa! el voltaje se hace cero y la intensidad de corriente vale 1 Amp indefinidamente. ¿Habremos creado el móvil perpetuo de primera especie? ¿Qué opina el lector?

Lo ayudamos a contestar. Agregue una resistencia de 1 R en serie. La ley de Ohm nos dice cuál es la ecuación diferencial a resolver. Se la resuelve siguiendo los mismos pasos que en caso previo y se los indica sin mayores detalles, dejando al lector comparar con el mismo:

```
DSolve[(i'[t] + (R/L) i[t] - (1/L) v[t] = 0,
i WI=0), i[t], t]
```

```
{(i[t] ->
      2
      50: t
      -----)}
      10. t
E
```

```
Plot[CiCtI/.%,v[t]3, It,0,13,
PlotStyle->( { Text E^w i (t) I_i^v {0.8,0.053},Thickness[.0061,
Dashing[ (.05,.02313,
I Text I W (t) 11 , {0,25,0.3531,Thickness~.00~1331;
```



Qué desilusión. No era un móvil perpetuo!

La posibilidad de usar sólo un comando para resolver la ecuación diferencial y un comando para graficar dejó tiempo y ganas para pensar. Con un estudiante motivado y un maestro orientador la herramienta que se posee brinda posibilidades insospechadas.

Circuito RLC serie

En un circuito RLC serie la física del problema está contenida en la regla de Kirchoff que corresponde a una malla ya que en este caso no existen nodos; Alonso y Finn prefieren llamarla directamente Ley de Ohm ([6], pag.665). Esta ley lleva a la ecuación diferencial de segundo orden en la intensidad de corriente i siguiente:

$$\left(\frac{\partial^2}{\partial t^2} i(t) \right) + \frac{R}{L} \left(\frac{\partial}{\partial t} i(t) \right) + \frac{i(t)}{LC} = 0$$

y el problema se limita a resolverla para el caso que al estudiante le interese, según sean las condiciones iniciales del problema.

La ventaja del **Mathematica** reside en que permite resolver este tipo de ecuaciones diferenciales con sólo un comando, denominado "DSolve". La sintaxis es la siguiente:

$$\text{DSolve} [i''[t] + (R/L) i'[t] + 1/(L C) i[t] = 0, i[t], t] \quad (1)$$

y la solución general dada por el programa es:

$$1 \{ i[t] \rightarrow E^{r_1 t} \left(\frac{(-RC + \sqrt{4LC^2 - R^2})}{2LC} t \right) + E^{r_2 t} \left(\frac{(-RC - \sqrt{4LC^2 - R^2})}{2LC} t \right) \}$$

También **Maple** posee comandos similares en este aspecto y la solución de la misma ecuación diferencial la muestra de la siguiente manera:

$$i(t) = -CI e^{\frac{(-RC + \sqrt{R^2 - 4LC})}{2LC} t} + CZ e^{\frac{(-RC - \sqrt{R^2 - 4LC})}{2LC} t}$$

Naturalmente la solución general tiene dos constantes indeterminadas. A partir de aquí se pueden obtener las soluciones particulares que puedan interesar según el problema planteado. Aún corriendo el riesgo de ser redundantes, se debe insistir en que el estudiante no debe saber programación para resolver las ecuaciones diferenciales que se presenten. Basta con conocer el comando correspondiente pero fundamentalmente saber la física involucrada en el problema que está estudiando!

Es posible generar una primera discusión del problema. En los textos tradicionales se suelen definir distintas formas de comportamiento de la solución del problema: sobreamortiguado, subamortiguado y críticamente amortiguado. Los autores entienden que ésta es una complicación adicional que debe soportar el estudiante sin ningún rédito. Es suficiente analizar la solución general presentada y a partir de los valores de R, L y C se determinará si la función solución es ondulatoria o exponencial, convergente o divergente, etc. La capacidad gráfica del **Mathematica** verificará la bondad del análisis efectuado por el estudiante.

Veamos algunos ejemplos que grafican lo dicho: Sea un circuito serie conformado por un condensador de 0,1 pF, un inductor de 100 mH y una resistencia de 560 R. Se carga el condensador hasta que la diferencia de potencial sea de 100 V. Se cierra el circuito; calcular la intensidad de corriente y la diferencia de potencial entre las placas del condensador. Las condiciones iniciales son $i(0)=0$, $V(0)=100$. Este problema se tomó del ejemplo 9.11 del libro de Nilsson [5], pero las variantes son propias de los autores.

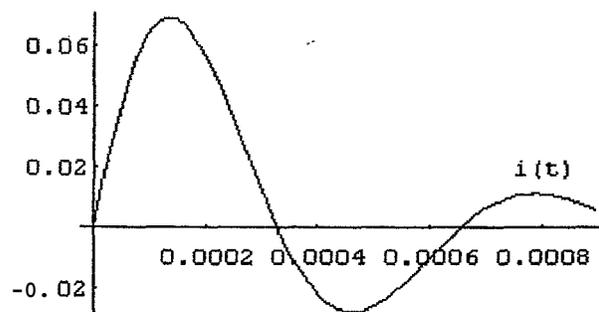
Ya se ha presentado el DSolve, simplemente se calcula y se indican los resultados.

```
R=560; L=0.1; C= 0.1 10-(6) ; Vo=100;
```

```
DSolve[(i''[t]+(R/L) i'[t]+1/(L C) i[t]==0,i[0]==0,
i'[0]==Vo/L3,i[t1],t1
```

(2)

```
{{j[t] -> .-----}}
0.104167 Sin[9600. t1]
2800. t
E
```

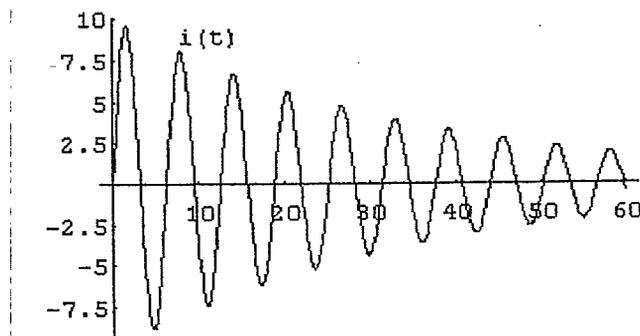


Dependerá de la habilidad del profesor sacarle más jugo a este ejemplo y a todos los posibles.

Cambiando los valores de R, L y C se pueden analizar soluciones muy diferentes. Por ejemplo sea $R = 0.560 \text{ Q}$, $L = 10 \text{ H}$ y $C = 0.1 \text{ F}$. La ecuación diferencial es la misma y las condiciones iniciales no se modificaron. La solución de la ecuación diferencial es ahora, siguiendo el mismo procedimiento con DSolve, la siguiente:

$$\{i[t] \rightarrow \frac{10.0039 \sin[0.999608 t]}{0.028 t}\}$$

y su representación gráfica:



Es posible discutir el tema y extraer diversas conclusiones. Se debe observar la enorme diferencia de tiempo para que la intensidad de corriente se haga cero en uno y otro caso. Este punto nos induce al siguiente comentario: Muchas veces en las clases y en los textos se proponen ejercicios con números que no necesariamente representan casos reales y que pasan inadvertidos ante la falta de herramientas para visualizarlos. Las técnicas computacionales actuales nos brindan otras posibilidades que nos están obligando a ser muy cuidadosos en la formulación de problemas.

Ya que hemos comprobado que este tipo de problemas se resuelve muy fácilmente, seguimos adelante incorporando en el circuito una fuerza electromotriz. En este caso la ecuación a resolver es:

$$\left(\frac{\partial^2}{\partial t^2} i(t)\right) + \frac{R}{L} \left(\frac{\partial}{\partial t} i(t)\right) + \frac{i(t)}{LC} = fem(t)$$

Para comparar con los ejemplos anteriores tomamos los mismos valores ya utilizados y agregamos como fem la función $V_e \sin(\omega t)$ para los dos casos:

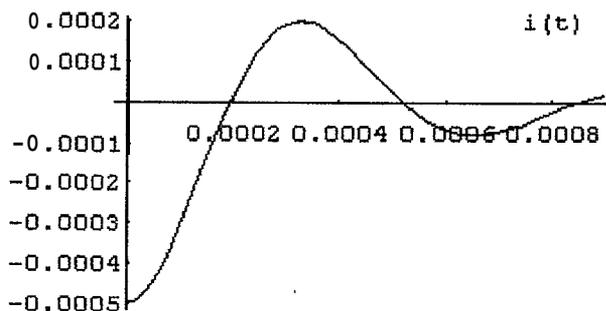
$$R=560; \quad L=0.1; \quad c= 0.1 \cdot 10^{-6}; \quad \omega=50; \quad V_e=100;$$

$$\text{DSolve} [(i''[t] + (R/L) i'[t] + 1/(L c) i[t] == \text{orne} (V_e(L) \text{Cos}[\omega t], i[0]==0, i'[0]==0), i[t], t] \text{Ch~P} \quad (3)$$

cuya solución matemática es:

$$((i(t) \rightarrow \left. \begin{array}{l} \frac{-0.0005 \cos[9600. t]}{2800. t} - \frac{0.000146 \sin[9600. t]}{2800. t} \end{array} \right\})$$

La representación gráfica que nos permite el Mathematica es la siguiente:



Es importante enfatizar que en ningún curso elemental de física se hacen estas cuentas y por lo general se le impone al estudiante alguna fórmula sacada de la galera para algún caso muy especial, y que por supuesto jamás recordará (ni utilizará).

Vayamos al ejemplo restante:

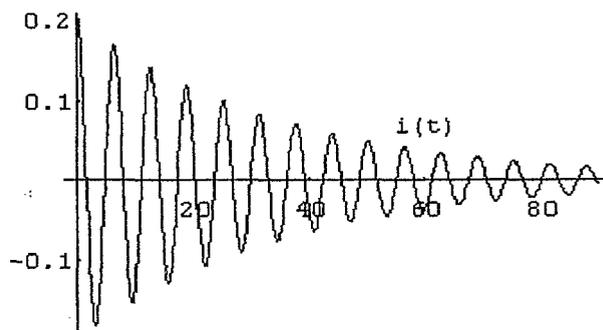
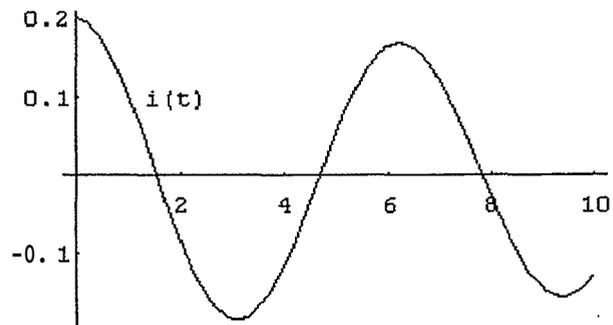
$$R=0.560; L= 10; c= 0.1; \omega=50; V_e=100;$$

$$DSolve[{\{i''[t]+(R/L)i'[t]+1/(L c) i[t]=\omega (V_e/L) \cos [\omega t], \\ - i[0]=0, i'[0]=0\}, i[t], t] // Chop$$

También escribimos su solución y luego lo representamos gráficamente:

$$\{ (i[t] \rightarrow \left. \begin{array}{l} \frac{0.2 \cos[0.9-99608 t]}{(7 t)/250} - \frac{0.00561 \sin[0.999608 t]}{(7 t)/250} \end{array} \right\})$$

Quando se la representa gráficamente y a simple vista parece que la amplitud se mantiene constante. El análisis de la expresión matemática seguramente permitirá un más profundo estudio de la respuesta del circuito a esta fuerza electromotriz externa. Además de ello (que aquí no se hace porque excede el marco del objetivo del trabajo) es posible tener una idea graficando tal respuesta para un tiempo mayor, digamos de 90 segundos. La representación de estos tiempos diferentes sigue a continuación:



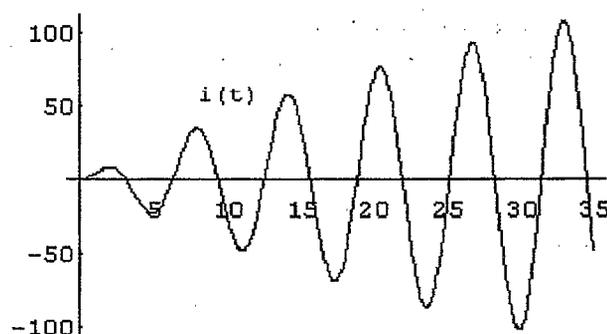
El ejemplo discutido, con diversas variantes propias se tomó inicialmente, como se dijo, del libro de Nilsson, ejemplo 9.11, pag. 370 y totalmente al azar. Sin embargo permitió estudiar distintos casos que sirven como orientación, y se espera que como motivación, para el desarrollo de todas las enormes posibilidades que el programa computacional brinda. Este es el objetivo.

El próximo paso es naturalmente estudiar qué es lo que sucede cuando la frecuencia externa corresponde a la frecuencia de resonancia del sistema. Obviamente se supone que el lector conoce cuándo ello ocurre. Como en los casos previos se escriben los valores de las magnitudes en juego, la expresión DSolve de la ecuación diferencial que describe el problema físico, la solución que brinda el **Mathematica** y la representación gráfica correspondiente.

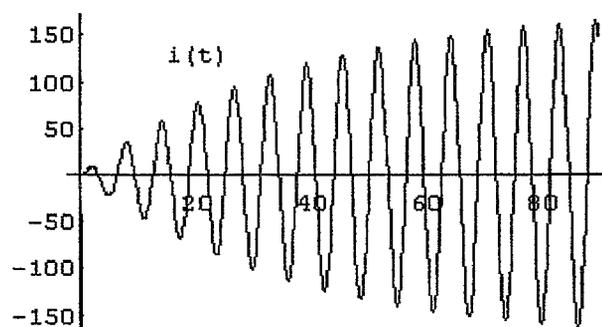
```
Rs0.560; L= 10; c= 0.1; ome=1/Sqrt[L c]; ve=100;
```

```
DSolve [ { 'i'' [t]+(R/L)'i'[t]+1/(L c) i[t]=  
ome (V+L) Cos[ome t] , i [0]==0, iv [0]=-0), i[t], t // Chop
```

```
{ { i[t] -1  $\frac{-1178.641 \sin[0.999608 t]}{0.028 t} + 178.571 \sin[1. f-1]$  } }
```



y pareciera que la intensidad de corriente crece indefinidamente. Sin embargo, el estudiante avisado se dará cuenta, al revisar la expresión de $i(t)$ que ello no es así. ¿Se equivocó el **Mathematica**? Obviamente no parece probable. Y resulta claro si se toma un intervalo mayor de tiempos. Veamos:



Queda claro que la amplitud no crece indefinidamente.

Otros casos de interés con aplicaciones prácticas se presentan cuando las fuerzas electromotrices que se aplican son discontinuas, por ejemplo un escalón cuadrado, un diente de sierra, etc. Se discutirán algunos de estos casos como ilustración de las posibilidades que nos brinda la herramienta computacional.

Consideremos en primer término una fuerza electromotriz escalón, cuya ecuación viene dada por:

$$\begin{aligned} f - e - 1 &:= 0 \quad /; \quad t \leq 0 \\ f - E t - 1 &:= 50 \quad /; \quad t > 0 \end{aligned}$$

La ecuación diferencial que se debe resolver es la misma ya estudiada e identificada como (3) con los cambios que devienen del ejemplo elegido ahora:

$$R=560; \quad L=0.1; \quad c= 0.1 \quad 10^{(-6)};$$

debemos indicar que si el lector intenta resolver la ecuación diferencial utilizando el comando DSolve generará el enojo del **Mathematica** con resultados imprevisibles.

En tales casos se debe solicitar al programa que resuelva la ecuación diferencial en forma numérica y para ello basta con escribir una N delante de DSolve y un pequeño cambio en las instrucciones como se indica a continuación:

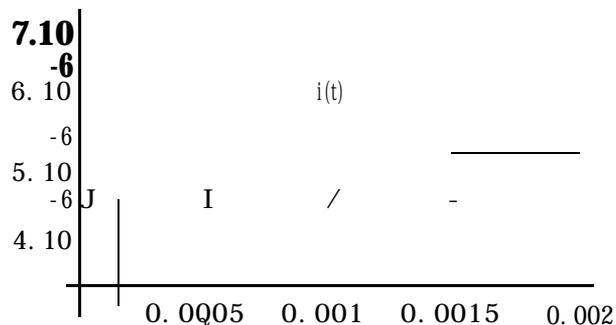
```
NDSolve[{i V[t]+(R/L)i'[t]+1/(L c3 i[t]==
  fern[t]/L,i[0]==0,i'[0]==03,i,{t,0,4.31}
```

y **Mathematica** nos indica que ha efectuado las cuentas en forma numérica con el mensaje:

```
{{i -> InterpolatingFunction[{0., 4.}, 0]}}
```

El intervalo de tiempo pedido al programa fue **(0,4)** y para cada caso es el resultado o de un análisis cualitativo del problema, o, lo que es más común, por el infalible sistema de prueba y error. Si el lector quiere dibujar la respuesta entre los valores de t 0 seg y 4 seg es libre de hacerlo. Se lo advertimos. Nosotros mostramos el siguiente dibujo:

```
Plot[Evaluate[i[t]/.%],{t,0,0.002},PlotStyle->
  {Text["i (t) 11,, {0.001,6 lo"}{-633131;
```



Observe el lector los valores de i y de t.

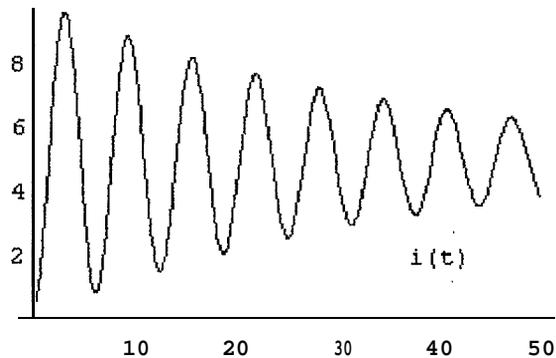
El otro ejemplo discutido siguiendo con los valores ya conocidos es:

```
R=0.560; L=10; c= 0.1;
```

los comandos que se utilizan y la curva que corresponde a la intensidad de corriente en el circuito es:

```
NDSolve[(i "[t]+(R/L)i'[t]+1/(L c) i[t]==
  fern[t]/L,i[0]==0,i'[0]==03,i,{t,0,5031
```

```
Plot[Evaluate[i[t]/.%],{t,0,50},PlotStyle->
  {Text["i (t) ",{40,2}]]];
```



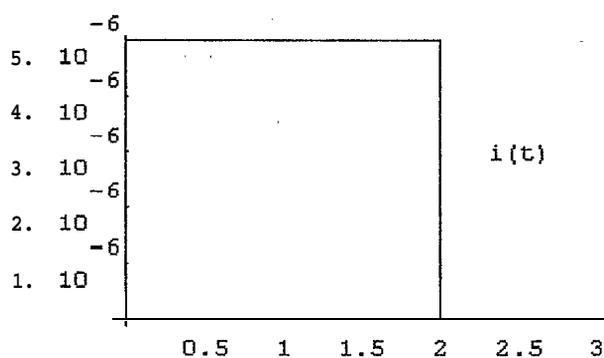
El lector puede extraer sus conclusiones, pero además, y esto es lo más importante, puede intentar repetir los cálculos con todas las posibilidades que se le puedan ocurrir. Advertimos, como ya se dijo, que el programa es muy quisquilloso por lo que recomendamos que se guarde continuamente la tarea realizada ya que deberá recurrir más de una vez al conocido Ctrl+Alt+Del aunque no le guste, ya que no siempre responde el comando para abortar (Alt+.).

Como otra aplicación, manteniéndonos en la misma línea de ejemplos, cortamos el escalón de manera que nos quede un pulso cuadrado. La fuerza electromotriz en este caso es:

```
fem [t-1 := 0 /; t<=0
f-Sm c t-1 := 50 /; (t>=0)&&(t<=z)
f@m Et-1 := 0 /; t>2
```

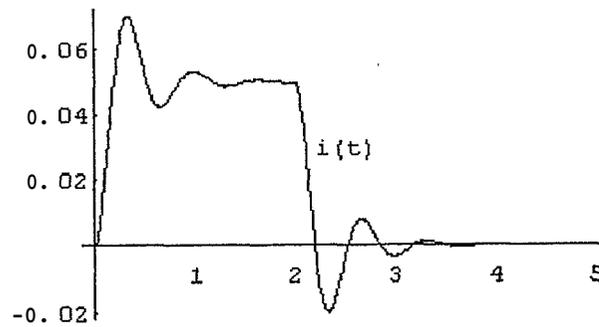
y los resultados sin entrar en mayores detalles son los siguientes para cada caso que se indica:

```
a=560; L=0.1; c= 0.1 10"(-6);
```



La respuesta es un pulso cuadrado. Si modificamos algo los valores de R, L y C se obtiene un pulso "casi" cuadrado:

```
R=56; L=10; c= 0.001;
```



Proponemos a continuación, considerar como fuerza electromotriz externa una sucesión de pulsos cuadrados, como el que terminamos de estudiar. Las ecuaciones no sufren ninguna modificación y la forma de calcularlas tampoco. De ello se encarga el **Mathematica**.

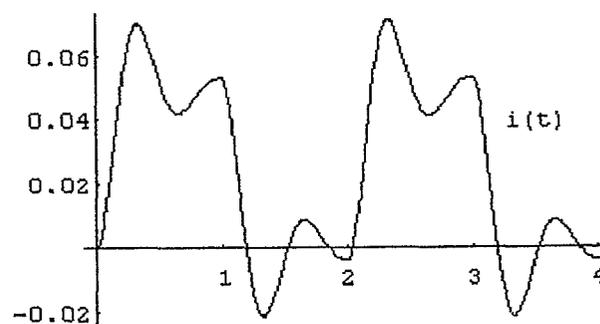
Consideremos la fuerza electromotriz definida de la siguiente manera:

```
femCt-1 := 0 ); t<=0
fem [ t-1 := 50' /; (t>=0)&&(t<1)
fem[tJ :- 0 ! /; (Iz>=1)&&(H2)
femlt-] := 50 /; (t>=2)&&(3)
£ e d t J := o /; t > 3
```

y utilizando el mismo **NDSolve**:

```
NDSolve[{i'[t]+(R/L)i'[t]-I-1/(L c) iCt]==
fem4[t]/L,i[0]==0,i~0l==0I,i,~t,0,4]l
```

llegamos a la representación de la intensidad de corriente en función del tiempo:



Circuito RLC paralelo

En un circuito RLC paralelo la ecuación diferencial que gobierna el sistema es la siguiente:

$$\left(\frac{\partial^2}{\partial t^2} v(t)\right) + \frac{\partial}{\partial t} v(t) + \frac{v(t)}{LC} = 0$$

que es similar a la correspondiente para la intensidad de corriente en el circuito serie. Entendemos que no es necesario hacer mayores aclaraciones. La solución general de esta ecuación diferencial es:

$$v(t) = -C d e^{\left(\frac{-1 \pm \sqrt{1 - 4LC}}{2LC}\right)t} + \frac{v(0) + \frac{1}{C} \int_0^t v(\tau) d\tau}{LC}$$

Para ver un ejemplo solicitamos la diferencia de potencial entre las placas del condensador para un circuito conformado por una resistencia de 4000 Q, un inductor de 10 mH y un condensador de 1 pF. El voltaje inicial en el condensador es de 10 V y su variación inicial en el inductor -240 000 V/s.

En el lenguaje de **Mathematica** la solución de esta ecuación diferencial se pide de la siguiente manera:

`R = 4000; L = 0.01; c = 10^(-6);`

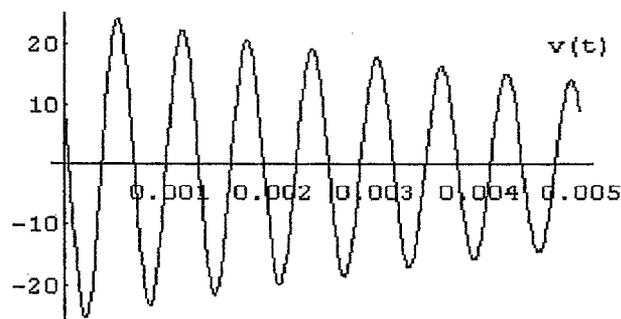
`DSolve[(v''[t]+1/(R c) v'[t]+1/(L c) v[t]==0,v[0]=10,
v'[0]==- 240000},v[t],t1//Chop`

y la solución particular es:

$$\left\{ \left\{ v[t] \rightarrow \frac{10. \text{Cos}[9999.22 t]}{125. t} - \frac{23.8769 \text{Sin}[9999.22 t]}{125. t} \right\} \right\}$$

siendo su representación gráfica la siguiente:

`Plot[Evaluate[v[t]],{t,0,0.005},PlotStyle->
{Text["v(t)",{0.005,20.}]}];`



Se ha mostrado mediante pocos ejemplos la capacidad de uno de los programas de computación que más ayuda puede brindar para eliminar el tedio de los cálculos de las soluciones de las ecuaciones diferenciales que aparecen cuando se estudian problemas, aún elementales, de circuitos eléctricos. Nuestro énfasis está puesto en los cursos básicos de física y la necesidad de un replanteo de los mecanismos de enseñanza entendiendo que el uso de la herramienta computacional actual puede ser de suma importancia para que el profesor pueda desplegar toda su inventiva sin depender de la complejidad de las ecuaciones que se puedan presentar.

Conclusiones

Los autores sostienen que la aparición de programas computacionales con capacidades simbólicas, numéricas, gráficas y de texto como el **Mathematica** o **Maple** (aunque no son los únicos) brindan la oportunidad, en algunos casos muy ansiada, de modificar toda la estructura de enseñanza de la física, de la misma manera que ya se está haciendo con la enseñanza de la matemática en el exterior (6).

Tradicionalmente estamos enseñando física, en los cursos básicos, a partir de simplificaciones en las definiciones de los conceptos fundamentales debido a la incapacidad matemática del receptor de la enseñanza. Las simplificaciones que se mencionan encierran muchas veces errores que luego son muy difíciles de subsanar ya que lo que primero se aprende es lo que más arraigado permanece [1].

La propuesta que estamos haciendo apunta a que desde el comienzo enseñemos al estudiante las definiciones correctas utilizando la herramienta que brindan mancomunadamente el análisis matemático y los programas disponibles (**Mathematica**, **Maple** o el que los mejore si así sucediera).

De esta manera un curso de física debería dejar de ser un tedioso compendio de definiciones: que el alumno no termina de entender y de problemas que el estudiante no alcanza a interpretar. Es bien conocido que el comentario común a la inmensa mayoría de los estudiantes es el siguiente: “entiendo (o en su defecto sé) la teoría pero no puedo resolver los problemas”.

En este trabajo se ha presentado sólo un tema que nunca llega a plantearse en los cursos básicos como aquí sí se ha hecho, Pero en esencia lo que se pretende es mostrar que de esta manera se debe reemplazar (o si no se quiere ser tan drástico complementar fuertemente) el libro de texto por el uso de una sucesión de “notebooks” (los cuadernos o folios del **Mathematica** o del **Maple**) electrónicamente vivos y en los cuales cada uno de los ejercicios puede ser cambiado y reejecutado tantas veces como el estudiante lo desee. Esto hace que cada estudiante pueda plantear su propio ritmo de aprendizaje y profundización del mismo, presentando en sus propios ejemplos todas las dudas que se le vayan generando y así transformar sus notas, que de otra manera serían material inerte, en un proceso interactivo sin precedentes.

Aún a riesgo de ser reiterativos también se debe mencionar que los pocos gráficos que aquí se han presentado sirven para contrastar con lo que no se puede hacer con la metodología habitual del pizarrón y la tiza.

Pero no debemos abandonar el pizarrón y la tiza sino darles otro destino que entendemos debe ser el de plasmar el razonamiento riguroso, dejando las cuentas y los gráficos para que los haga la computadora asistida por el mejor programa computacional que exista en ese momento. Tomando en forma libre expresiones escuchadas en seminarios dictados por el Dr. Carlos E. D'Attellis podemos sintetizar: las ideas en el pizarrón, las cuentas en la computadora [8].

Referencias

- [1] SIPCIC, Slobodan R., "What is Wrong in the Way We Teach Mechanics and How to Fix It", *Math. in Ed. & Res.*, 4, (4), 5 (1995).
- [2] CALIGARIS, Roberto E. y CALIGARIS, Marta G., "Ab Initio: Mathematica en la enseñanza de la física. 1. Mecánica del punto", *Infocom96*, Buenos Aires, junio 1996.
- [3] CALIGARIS, Roberto E. y CALIGARIS, Marta G., "Mathematica en la enseñanza de la física. Introducción a la mecánica cuántica", *Primer Congreso Argentino de Enseñanza de la Ingeniería*, Río Cuarto, Córdoba, octubre 1996.
- [4] WOLFRAM, Stephen, "MATHEMATICA. A System for Doing Mathematics by Computer". Addison-Wesley, 1991.
- [5] NILSSON, James W., "Circuitos Eléctricos", cuarta edición, Addison-Wesley Iberoamericana, 1995.
- [6] DAVIS, Bill; PORTA, Horacio and UHL, Jerry, "Calculus & *Matemática*", Addison Wesley Publishing Company, 1994
- [7] ALONSO, Marcelo y FINN, Edward, J., "Física" Vol. II: Campos y ondas. Fondo Educativo Interamericano S.A. Barcelona 1970.
- [8] DATTELLIS, Carlos E., *Primeras Jornadas de integración y vinculación regional*. Universidad Tecnológica Nacional, Facultad Regional Rosario, Rosario, Santa Fe, septiembre 1996.

MODALIDADES DE APLICACION DE LA INFORMATICA

Prof. Jarmila M. Havlik

Alguien me dijo "la computadora en la escuela debe ayudarle al docente a hacer mejor, lo que mejor sabe hacer, que es enseñar".

Con esta exposición quiero mostrar que hay más que eso, y que otra postura, no opuesta pero si diferente, se presta mejor para Interpretar la actividad: "proponiendo una planificación conciente de un proyecto pedagógico en el cual el ordenador pueda jugar un papel que no puede ser delegado a los propios profesores".

RELACION DE LA ENSEÑANZA DE COMPUTACION CON LAS TEORIAS DEL APRENDIZAJE.

Partimos de un hecho indudable: los chicos aprenden, y mucho, durante su paso por la escuela. Interpretar este hecho requiere de una teoria, significa observar, registrar, evaluar, en función de alguna Idea previa sobre que es aprender, que cambios deseamos que se produzcan, etc. Como siempre, lo que encontremos depende en gran parte de la teoria con la que vayamos a buscar los datos, pero en este fenómeno particular que es el aprendizaje, no sólo la Interpretación sino el proceso mismo de producción depende de las teorías. Porque un docente enseña siempre desde alguna postura, más o menos explicita, que le da pautas sobre como explicar, presentar actividades, corregir, evaluar, etc.

Cómo se aplica esto al análisis de lo que sucede en la enseñanza de computación?

En toda situación de aprendizaje escolar hay tres polos, la llamada triada pedagógica:

S

A D

El sujeto de conocimiento -alumno- suele adoptar una actitud particularmente activa, Intellectualmente hablando, sobre todo por 'la dinámica grupal que se acostumbra en esa clase.

El polo del saber requiere una caracterización en tanto objeto computacional, puesto que es un "observable" en el sentido piagetiano pero además un objeto lógico que contesta como un interlocutor, a quien interactúa con él.

El polo del docente es muy particular, en tanto mediador en esa dinámica Interactiva, entre los saberes que quiere comunicar y los que aportan los mismos alumnos.

He usado varias veces la palabra Interacción, pero cabe aclarar que esta no implica construcción, una interacción simplemente describe un ida y vuelta entre el S y el O, pero puede no ser constructiva. Trataremos de mostrar con los ejemplos que' puede ser serio en muchos casos, sencillos pero interesantes.

La posibilidad de construir el aprendizaje en computación estará en gran medida condicionado por el diseño didáctico que realice el docente, y en ese diseño tiene que tener en cuenta los tres polos.

El objeto de conocimiento: computación o curricular?

En la escuela los objetos de conocimiento son los contenidos curriculares. Por eso vamos a simplificar la pregunta, y reducirla por ahora a: contenido "computación" o contenido curricular? o es una disyunción inclusiva?

Actualmente hay mucha variedad de propuestas, pero no vamos a analizar esta multiplicidad, sino su sentido general, que reúne dos vertientes: el aprendizaje del uso de la herramienta misma y los contenidos curriculares, ultimamente revalorizado. En computación no ocurre lo mismo que con lápiz y papel. El cuaderno y el lápiz son medios para hacer una cuenta, y no son estudiados en sí mismos por el alumno, aunque su utilización requiera cierto aprendizaje (cuidado, prolijidad, secuencia de hojas, etc.). Con la computadora, también se aprende algo de su manejo al usarla, pero cuando un niño tiene "computación en la escuela" los padres esperan que aprenda mucho de "computación". De todos modos parece un falso dilema, sobre todo cuando se trabaja con utilitarios y/o programación.

Además, cuando se parte del propósito de establecer relaciones curriculares, no hay que perder de vista que el alumno es la única sede del establecimiento de dichas relaciones curriculares; no el escritorio de los especialistas ni siquiera la planificación de los docentes, por bien intencionada que esté. No se garantiza la relación curricular porque ésta se proponga en los papeles, sino cuando un chico dice: Ah!, esto es como lo que hicimos en ...matemática. o lengua, etc.

Tampoco vale la pena -con las costosas máquinas y el escaso tiempo que se dispone-, hacer una relación meramente externa: el mismo tema que se da en clase, pero en una pantalla, por más Interactiva que parezca, si no se piensa en un vinculo intrínseco a la disciplina de que se trate.

Podríamos decir que hay modos de usar la computadora que comportan una pérdida de contenido y otras una conservación y hasta una ganancia.

Ejemplos: El procesador no tiene pérdida para el tratamiento de la lengua escrita, así como el lenguaje Logo no la tiene para la geometría, la lógica, etc. Aunque en todos los casos, mucho depende de la propuesta, estas herramientas brindan mucho saber con ideas sencillas.

Si bien no fue creado para educación, el procesador de textos resulta "transparente" para permitir ver la manera de pensar de los chicos, aprender que interesa desde el Brea de lengua y paralelamente aprender computación". Este utilitario no oculta, más bien revela, el nivel de conocimientos -y problemas- de los alumnos, pues todo lo de la lengua esta ahí, potencializado por las virtudes del programa.

Respecto al Logo, cuál es la gran diferencia de este objeto, tal que explique el esfuerzo que se realiza por justificarlo?

Creemos que el hecho de ser un objeto especial, programable, permite -y obliga- interacciones especiales, y modificaciones en varios aspectos del aprendizaje escolar:

progreso en la autonomía, al tener respuestas inmediatas y poder avanzar reflexionando sin depender de una tardía corrección.

disponer de un instrumento de verificación abierto a las preguntas, para confrontar las llamadas "ideas previas".

- contar con la memoria y la posibilidad de programación

Cómo se interpretan todas estas particularidades, desde las teorías del aprendizaje?

Volviendo al principio, decimos que las teorías quieren dar cuenta del proceso por el cual los niños aprenden, y la verdad es que los niños aprenden aunque uno no sepa que teoría está utilizando.

Por ejemplo, un docente dice:

"Los chicos con la máquina aprenden en forma constructiva; por ensayo y error".

Tomado al pie de la letra;esto es una contradicción, puesto que la teoría que postula el ensayo y error como método de aprendizaje es justamente a la que se opone el constructivismo de Piaget, con la teoría de la asimilación.

Si el esquema es

Ensayo -- Error

Ensayo ---- Error

Ensayo -- Acierto -- Refuerzo (aprendizaje)

Pero el docente que afirma lo primero, no dice algo descabellado, pues es cierto que los chicos prueban, sin temor, "ensayan" soluciones, se equivocan, vuelven a probar. Pero en ese proceso hay esquemas previos, reflexiones antes de los siguientes ensayos, y confirmación/disconfirmación de hipótesis, aunque no las puedan explicitar.

La teoría pura del ensayo y error, en cambio, plantea ensayos ciegos, como los de la rata en el laberinto.

En esa perspectiva, el error es lo que debe ser rápidamente descartado, y lo es porque la respuesta no es reforzada. En educación esto se traduce en el temor de los docentes de que los errores "queden fijados". Por eso tratan de dar pautas muy claras y orientaciones precisas que los eviten lo más posible.

Con la computadora, los chicos se equivocan mucho, corrigen mucho y por eso se habla de "ensayo y error". Pero en realidad lo productivo de esto es que los errores sí les sirven para pensar, y las rectificaciones para que no se vuelvan a dar son verdaderos cambios que sirven para otras oportunidades en las que hay algún problema. Es cierto que a veces extienden la solución adonde ya no corresponde, pero eso será fuente de nuevas reflexiones y diversificación de las respuestas.

Con esto hemos criticado la teoría conductista, que se basa en una buena estimulación (garantizada por el atractivo de pantallas cada vez más hermosas) y en respuestas que pueden controlarse. Para el conductismo, el sujeto es pasivo, el aprendizaje no es una cualidad intrínseca al sujeto, necesita ser impulsado desde afuera.

En el aprendizaje reproductivo se busca el éxito y se aprende sobre todo de los aciertos; en el aprendizaje comprensivo son los fracasos los que resultan más informativos.

Tal vez centrándonos en el tema de los errores, que tienen un sentido específico, se pueda expresar parte de la concepción que sustenta la teoría psicogenética. Esta no se originó en un interés sobre el aprendizaje, y sin embargo tiene tantos aspectos que se vinculan con el mismo que sería muy difícil resumirlo aquí.

EL SENTIDO DEL ERROR:

Para el conductismo el error debe ser descartado, pues se corre el riesgo de que quede "fijado".

Para el constructivismo el error es significativo de muchas maneras.

La teoría psicogenética se ha ocupado de los errores en diferentes aspectos:

En cuanto a/ problema epistemológico:

Como llamado de atención sobre una lógica diferente a la del adulto (cosa que hoy nos parece natural, pero que fue revolucionarlo en su momento).

- Como tema interesante para ser investigado, en tanto da información sobre cómo piensan los chicos en las cuestiones clásicas de las ciencias (conservaciones, reversibilidad, transitividad), por medio del método clínico-crítico.
- Como eje para investigar nuevos temas: de qué manera encontrar conocimientos investigables psicogenéticamente? será detectando errores sistemáticos, que revelan un proceso constructivo.

Con referencia al aprendizaje en general:

- Como camino para investigar de qué manera proponer un problema significativo que provoque diferentes respuestas, en el marco de situaciones tales que hagan que se ponga de manifiesto el error y con ello se de lugar a la posibilidad de una perturbación para el sujeto, que lleve a un conflicto cognitivo y permita su superación.
- Como derivación didáctica, preguntándose acerca de qué significan los errores en clase, cuáles son los diferentes tipos de error, cuáles son los que resultan útiles para provocar conflictos cognitivos y/o socio-cognitivos.

El docente no es un Investigador, el docente quiere enseñar.

Ciertamente, hay una gran diferencia entre investigar y enseñar, el docente quiere lograr aciertos, aprendizajes tales que superen los errores, y no errores interesantes para seguir investigando.

Pero una máquina tan especial, una inteligencia artificial, hay que aprovecharla como tal al máximo, y entonces trabajar con ella en la forma más directa posible permite poner a la vista muchos errores, y creemos que

La Interacción con una Inteligencia artificial permite poner de manifiesto muchos errores interesantes.
La Interacción con algunos programas educativos permite ocultar errores interesantes.

Entonces: algunos programas los ocultan en la medida en que saltean etapas de construcción.

Los chicos aplican sus esquemas (de acción, de conocimiento) y cuando se equivocan tienen distintas conductas: a veces salen del programa, otras repiten la misma acción - operación, probando, y otras veces se preguntan qué pasó, y hasta leen los mensajes de la máquina. Pero para esto tiene que haber: a) cierto interés, que está dado por la meta que quieren lograr, b) posibilidad de

comprender el problema y por último c) la certeza de que en el contexto de esa situación está la solución o las herramientas para encontrarla.

Algunos ejemplos:

ERRORES 'QUE PERMITEN SU CORRECCION POR VERIFICACION:

Para que la corrección derive de una verificación, ésta tiene que estar bien orientada por el problema, o sea delimitado el margen del error, y además se debe disponer en el entorno de las posibilidades de efectuar la constatación; o sea que el error sea revisable con los medios del sistema mismo. En caso contrario, es difícil garantizar que se mantenga su utilidad, porque si el alumno va a buscar información a algo alejado, a veces ni se acuerda para qué iba.

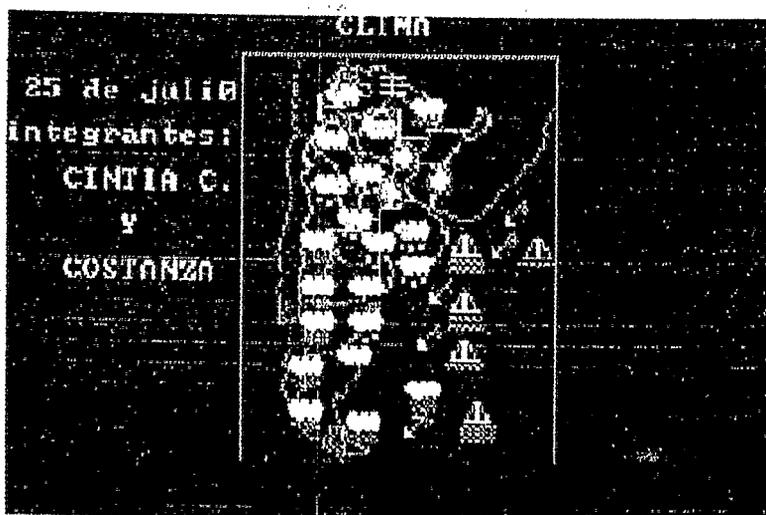
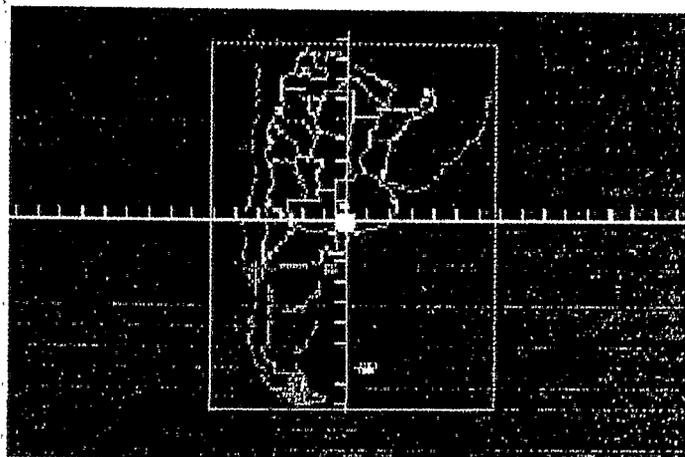
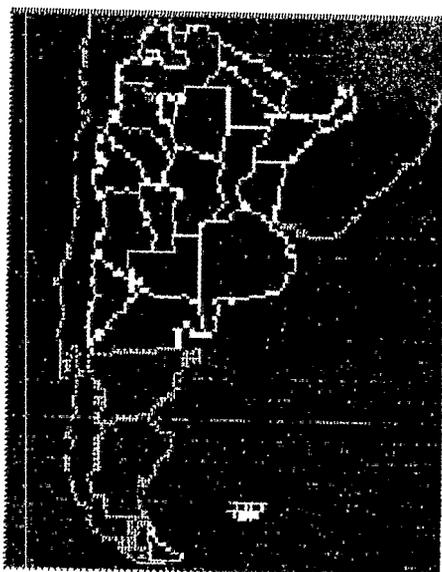
Por ejemplo, trabajando con ejes cartesianos, en Lenguaje Logo (en este caso Logo Writer) éstos se cruzan en el centro de la pantalla (00). En el caso de querer ubicar figuras en la pantalla mediante la orden FPOS (fijar posición), las previsiones de los chicos a veces son erróneas, ya sea por pensar primero en la ubicación en Y y luego en X, o por los sectores de números negativos. Propusimos un trabajo para ser contestado con lápiz y papel, en el que hay que analizar proposiciones y determinar su valor de verdad, con contenido relativo a esta problemática .

Esta tarea puso de manifiesto muchas dudas de los chicos, sobre todo porque implicó un trabajo de reflexión sobre una práctica que se estaba llevando a cabo con la computadora, al programar el mapa del clima .

Algunas de esas proposiciones eran:

- Los números negativos de la Coordenada X están del 0 para la izquierda.
- Los números negativos de la Coordenada Y están del 0 para arriba.
- El orden de los números al indicar coordenadas es siempre el mismo: primero el de X, segundo el de Y, y no se puede cambiar.
- La X y la Y no pueden tener el mismo número en la ubicación de la tortuga.
- ... y muchas más

Pero lo interesante es que, con la hoja en la mano, pudieron preguntar" a la computadora acerca de cada una de las dudas y los errores detectados, y corregirlos. Esta es una interacción no dirigida desde la máquina, como puede ser un ser educativo, sino dirigida por el alumno.



ERROR CONSTRUCTIVO: ejemplo de circunferencia y la mitad para el diámetro.

Para el mismo proyecto de clima, deben representar en gráfico de sectores ciertas temperaturas máximas y mínimas. Se propone la realización de la circunferencia, y el problema aparece cuando hay que llegar al centro de la misma.

Un diálogo muy común que se suscita es como el siguiente:

Alumno - ya lo puse en el medio.

Docente - Cómo hiciste?

A - A ojo, lo puse justo.

d - Como sabés que es justo? hay una cuenta que puedes hacer para estar seguro, y si no es exacto todo va a salir mal. Para el centro tenés que saber el

radio, y tal vez te sea mas facil empezando por el diámetro. Sabés calcular el diámetro?

A - (aventura' fórmulas) Pi por radio al cuadrado, etc.

D - Sabes cuánto mide la circunferencia en total, su longitud?

A - Sí, 360 (pasos)

D - Cuánto medirá el diámetro

A - La mitad, 180

D - 180 qué?

A - grados, no ves que es un "llano"?

D - te pregunto de longitud

A- también 180, porque es la mitad

D - Recorre (con el lápiz, tiza o lo que corresponda) una mitad del contorno, y luego el diámetro, preguntando si ambos pueden ser lo mismo: la mitad.

A - No, esta es más corta, debe ser 90 (el diámetro)

D - Cómo averiguas la longitud, sabiendo el diámetro?

A - Pi por diámetro

D - Ahora sabés la longitud, como seria la cuenta?

A - Al reves, 'ah! ya sé, dividido.

D - Pedile a la computadora que te haga la cuenta: (en Logo se puede hacer directamente, da el resultado en la parte de comandos)

A - Ese es el diámetro, la mitad... hace la cuenta . . .

En fin, muy resumidamente es una situación muy común, de reflexión sobre: a) una constante en un sistema de transformaciones: cambia la circunferencia y proporcionalmente cambia el diametro; etc. y b) la aplicación de una fórmula usada en forma automática, y acá vista en funcionamiento efectivo, con un resultado evidente y dinámico, que es el desplazamiento de la tortuga según medidas precisas.

ERROR DE INFORMACION PERO QUE SIRVE PARA PENSAR: poner un objeto en Fpos (-30 70), luego en [30 -70) , pensando en [-30 -70), y creyendo que el menos abarca a los dos, en el primer caso porque está al principio, en el segundo porque se reparte. No son ideas tontas, en matem. al pp. a veces abarca todo, según un paréntesis, y el del medio es por sentido común. Corregirlo significó pensar qué es el signo para un número negativo (lo d e t e r m i n a :

Hablamos del contenido, y del sujeto en tanto activo constructor. También es importante cómo influye todo esto en el rol del docente.

Sin duda la presencia de la computadora provoca modificaciones en este punto. La tríada pedagógica D - S - A; se hace muy especial cuando el saber puede ser lo que este puesto en la máquina, y el docente no es el depositario del mismo sino un mediador para que ese saber llegue a ser asimilado.

Para analizar esta dinámica podemos apelar a la teoría de vigotsky que, al poner el énfasis en la internalización de la cultura, valoriza especialmente el rol

del adulto como intermediario y posibilitador de los aprendizajes de los niños. La Intervención crea la llamada Zona de Desarrollo Próximo, en la que se promueve el desarrollo hacia lo que todavía no puede hacer sólo pero que hace con ayuda. En cierto modo, uno podría pensar que, dado que Vigotsky desarrolla sus Ideas en un mundo sin computadoras, hoy una parte de la Información que necesariamente brindaba el adulto al niño se la puede dar la maquina; aparte de los programas realizados especialmente para informar, de hecho en la interacción las respuestas de la misma dan datos sobre los cuales pensar (mensajes de error, opciones, etc.). Pero esto requiere un análisis más profundo, porque la experiencia indica que los chicos siguen requiriendo el diálogo con el docente y con sus compañeros, aún para Interpretar la comunicación con la computadora.

Por eso, entre los extremos de analfabetismo y telemática, sugiero observar al niño real, que es el que se equivoca en las mismas cosas, con o sin computadoras, y brindarle oportunidades de generar sus propios productos, por ejemplo con el Logo, aunque no sean tan espectaculares, poniendo en juego criterios propios.

CURRICULUM VITAE

Datos personales

Apellido y nombres: HAVLIX, JarmilaMaría

Domicilio: Canalejas 266 - (1609) BOULOGNE -Pcia. Bs. As.

Teléfono: 765-5120 Fax. 737 - 1052 DNI: 10.620.201

E-mail sofia@overnet.com.ar

TITULO: iprofesora de Enseñanza Media, Normal y Especial en Filosofía
Otorgado por: FACULTAD DE FILOSOFIA Y LETRAS - Universidad
Nacional de Buenos Aires

Actividades profesionales actuales:

Docente de Psicología Genética - Cátedra de Psicología Genética- Carrera de
Ciencias de la Educación - UBA.

Investigadora de UBACyT (Ciencia y Técnica de la UBA) sobre “La
Adquisición de la noción de amplitud del ángulo”, psicogénesis y
derivaciones didácticas (diferenciando grupos con y sin computación).

Coordinadora y docente de informática en instituciones educativas de
EGB.

Docente de computación en educación especial -discapacidad mental-.

Colaboradora estable en la Revista Novedades Educativas, sección
Informática educativa

AN EXPERIENCE IN A DISTANCE GRADUATE COURSE ON SOFTWARE ENGINEERING

García Martínez, R., Rossi, B. & Gramajo, E.

**CAPIS. Graduate School. Buenos Aires Institute of Technology
Madero 399. (1106) Buenos Aires. Argentina**

rgm@itba.edu.ar

A B S T R A C T

This paper describes a latin-ameritan experience on the teaching-learning process in a distance graduate course on software engineering, the didactical aspects are discussed, the curricula of the course is considered, some statistics of the project are presented and provisory conclusions outlined.

1. INTRODUCTION

The development of the modern communication methods is producing new work modalities among those which starts to emerge the “distance work” concept where a person, without be moved of his household and through the means of communication, accomplishes his labor for the company in the one whi ch works, through a flow “of informatión”.

Obviousiy these changes, burst upon the field of the teaching being coined terms, already of common use, such as “virtual classroom “. As in the case of the mentioned employee, the student, without be moved his place, “lives a similar reality” to the one which are accustomed with our classic teaching methods where the professor exchanges knowledge with students that are found contiguous to him.

We said contiguous students to the professor in the sense of ,the fact that, SO^o much they as the professor, are found few meters one of the other.

AS a previous step to the “virtual classroom” exist already, what is known as Open Universities, where the student studies his career without his physical presence in the studies center [García Martínez et al., 1991; García Martínez & Marsiglio, 1991; Garcia Martínez & Perichinsky, 1996a].

The knowledge exchange is produced through use of all the means of communication known, such as the telephone, the data nets and other. In the societies that they have implemented this study modality is being producing a massive increase in the level of knowledge transmission, preparing them for the immediate future ñ the one which already is perceived that will be gone producing deep modifications in the relative levels among the countries. Those

of them that accumulate more and better “brain matter” will be those which will have priority in the international arrangement.

With this methodology students do not attend regularly to the University. Receive their lessons in the form of chapters (didactic units in the tutorial terminology) carefully designed so much academically as pedagogically. Each one of them, once studied and understood, is evaluated through a questionnaire prepared specifically and that once answered, is returned to the University.

Obviously, it can seem that who studies with this modality is found “orphan” of the didactic support and of the explanations that can be accomplish by the professor.

It is here where the means of communication play the decisive role, since each student is assigned to a monitor-professor who can be consulted every day, personally or by phone, in a wide schedule along the day, or in any moment by mail, fax or electronic mail. This way of communication form a direct relationship among each student and his monitor-professor, that improves the traditional education, since is totally personalized.

Upon completing all the corresponding units to a matter (a module in the tutorial terminology) is fixed a date of an examination, which necessarily must be yielded in this University.

2. DIDACTICAL ASPECTS

Basically the tutorial programs present aspects that permit, by a part, to conjugate the individual characteristics of the student to encompass his speed of knowledge acquisition and by the other, to achieve a personalized dealing among the professor and the student.

As consequence of this original implementation modality is achieved, also, the optimum utilization of the time by the student. Summarizing through the tutorial modality is procured:

- The student is the person who rules his progress speed in the course.
- There exists, a personalized relationship among the monitor-professor and the student.

The utilization of the time is optimized.

This modality consists in abandoning the classic form of contiguous teaching that implies all / the time of the teaching with the physical presence of the professor, by the discontinuous, that permits the student to rule his apprenticeship, adapting it to his possibilities and availabilities of place and

opportunity, dispensed of the obligation of the physical presence of the professor, replacing it by the presence, through documents perfectly planned for the self learning process by the the monitor-professor.

The traditional professor that imparts knowledge is replaced by the monitor-professor that guides, amends and evaluates. The setup of the program has been made in independent modules and these are divided in didactical units.

The tutorial methodology presupposes a self assisted regime with support of monitor-professors and designed by the discontinuous modality. There are combined the theoretical classes with practical exercises.

The tutorial methodology has made possible that removed persons from specialization centers or with difficulties to attend, yet when are near, they could carry out their education, update or development in his profession. The difficulties of the travelling and the subordination to rigid schedules has been solved by the flexible tutorial methodology, permitting the student to make compatible his training with the availability of time and place. This modality causes that the own student will be the one which regulate the advance of his studies.

The developed methodology use the figure of the tutorial-professor, as power element of the system, the one which directs and supports to the student.

The monitor-professor is available for personal interviews or by telephone every hour every day and regularly by letter, telefax, or electronic mail.

The evaluations and follow-up of the student are accomplished through the controls that are proposed in each Didactical Unit. These are transmitted to the monitor-professor and after amended, they will be returned.

3. CURRICULA OF THE COURSE

We accept as a premise that the Software Engineering concept that were used until a little time ago, it is not sufficient to provide a data processing response to any type of problem. It were thought that the aim of Software Engineering was solving problems of the aigorithmic type and that the aim of Knowledge Engineering was occupied of heuristic problems.

Today this concept has disappeared. Nowadays, exist a unified view of the Software Engineering and who develop it should know both fields of the data processing [Feldgen et al, 1996, Garcia Martinez & Perichinsky, 1996b]. The software engineering students should know profoundly the problems that are presented in the software as a rule and they should have a clear vision that the systems present a techniques duality to apply. A professional of the data processing specialized in Knowledge Engineering of will not be able to dispense of the knowledge, technical and traditional software methodologies (as a way of calling them), neither conversely.

Even though the Software Engineering and the Knowledge Engineering were born as separated branches within the data processing, as time goes by both branches have gane being approached close without any doubt.

The Software Engineering provides to the Knowledge Engineering the techniques and methods that permit to improve the software production and, therefore, to accomplish a final product that it will be more correct, responsible, verifiable, updated.

On the other hand, the Knowledge Engineering permits to improve the traditional software production through different tools, as for example intelligent CASE tools, the rehusé based on knowledge bases and other.

For this reason, among others, it is necessary that the software engineers as well as the knodlege engineers know in addition to the tools, technical and methodologies of each discipline, the joint approaches that they are being accomplishing.

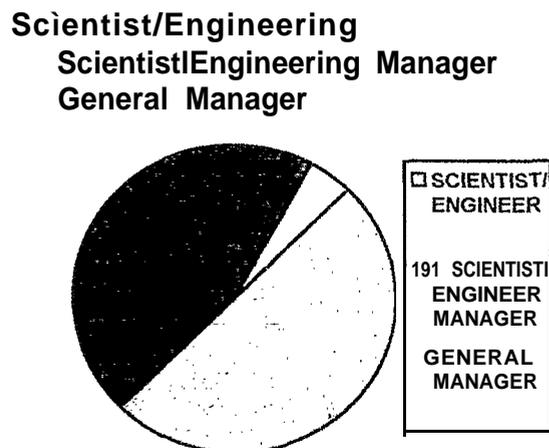
4. SOME STATISTICS

4.1. STUDENTS CHARACTERIZATION

In this section, the population of the students that are involved in this educational project, is caracterized in two ways: by the responsibility level that have as engineers in the enterprise were they work, and the distance from his home to the educational institution.

4.1 .I. RESPONSABILITY LEVEL

The responsibility level that the students have as engineers in the enterprise were they work have been divided in:

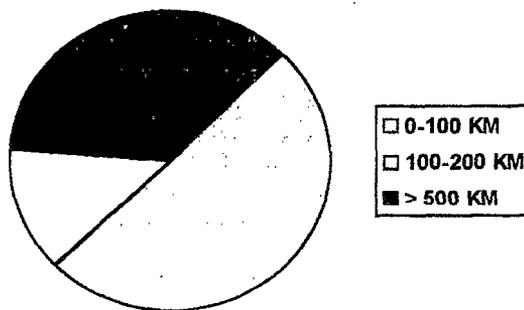


Is remarkable that near the 50% of the students have medium or high managing responsibilities in the enterprises were they work.

4.1.2. DISTANCE

The distance from the students home to the educational institution have been divided in:

- A distance of 0-100 km from the educational institution to the students home.
- A distance of 100-200 km from the educational institution to the students home.
- A distance longer than 500 km from the educational institution to the students home.



Is remarkable that this methodology supports the teaching-learning process of near the 50% of the students who can not be physically contiguous to the monitor-professor.

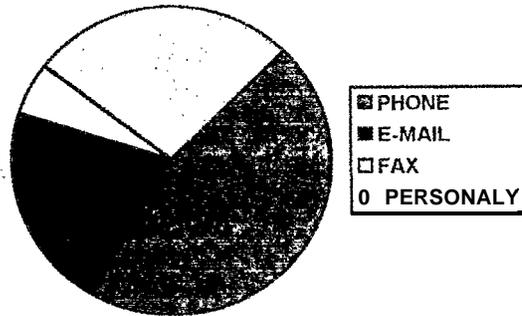
4.2. TEACHING-LEARNING PROCESS

In this section, the teaching-learning process involved in this educational project, is characterized in three ways: by the way of student/monitor-professor communication takes place, the amount of consults that a student make per week and the amount of weeks between two student examinations.

4.2.1. WAY OF STUDENT/MONITOR-PROFESSOR COMMUNICATION

The way of student/monitor-professor communication takes place have been divided in:

- Phone
- E-mail
- Fax
- Personally



Is remarkable that near the 75% of the students, prefer to comunicate with the professor ón-line' (by phone or personally) than in a differed way (near 25% by fax or e-mail).

4.2.2. AMOUNT OF CONSULTS PER WEEK

The amount of consults than in average a student make is :

4.5 consults per week

Each consult takes in average 20 minutes, this makes 1.5 hours of consult per week per student.

4.2.3. EXAMINATIONS.

The amount of weeks between two student examinations is :

AVERAGE. 1 examination per 3 weeks
 MINIMUM: 1 examination per 1.5 weeks
 MAXIMUM: 1 examination per 13.5 weeks

Each examination follows the conclusion of a didactical unit, as the course is composed by 35 didactical units the expected time used in the teaching-learning process: per student in average is 105 weeks (near two years).

5. PROVISORY CONCLUSIONS

As the teaching-learning process developed for this project is been modified on-line during the project development, by force, the conclusions are provisory.

Near the 50% of the students involved in the project have medium or high managing responsibilities in the enterprises where they work and can not be physically contiguous to the monitor-professor during the teaching-learning process.

Around the 75% of the students prefer to communicate with the professor on-line' (by phone or personally) than in a differed way (near 25% by fax or e-mail). Each consult takes in average 20 minutes, this makes 1.5 hours of consult per week per student.

The expected time that is going to be used in the teaching-learning process per student is 105 weeks (near two years).

The authors think that the teaching technology used in this educational project is different from that ones that have been used up to now. The latin-american educational market has been traditionally opened to new educational experiences, and the one described in this paper allows to close the cultural gap between Latin-America and developed countries.

The authors wish to thank to Prof. Pazos who encouraged them to carry this project, and to the Authorities of the School of Computer Science of The Politechnical University of Madrid who make available the support of the didactical and technical aspects of the project.

6. REFERENCES

- García Martínez, R., Peri, J., Marsiglio, A. 1991. *Proyecto de Currícula de Maestría en Ingeniería del Conocimiento*. Actas del Primer Congreso Iberoamericano de Educación Superior en Computación. Páginas 140-145. Santiago. Octubre 1991. Chile.
- García Martínez, R. y Marsiglio, A. 1991. *Ingeniería del Conocimiento. Una Experiencia Docente*. Actas del Primer Congreso Iberoamericano de Educación Superior en Computación. Páginas 202-208. Santiago. Octubre 1991. Chile.
- Feldgen, M., Cha, O., García Martínez, R. & Perichinsky, G. *On the Development of a Curricula in Informatics Engineering*. Proceedings of the XIV International Conference on Applied Informatics. Páginas 161-163. Innsbruck. Austria. 1996.
- García Martínez, R. y Perichinsky, G. 1996a. *Consideraciones sobre la Capacidad Investigativa en Unidades Académicas Universitarias*. Proceedings del II Congreso Internacional de Informática y Telecomunicaciones. INFOCOM96. Páginas 171-175. Buenos Aires. 1996.
- García Martínez, R. y Perichinsky, G. 1996b. *Un Proyecto de Maestría en Ingeniería de Sistemas Inteligentes de Producción*. Actas del V Congreso Iberoamericano de Educación Superior en Computación. (en prensa) Mexico D.C.. Octubre 1996. Mexico.

EL TELETRABAJO

por Martha Alicia Alles

Teletrabajo es un fenómeno actual producto de la sociedad laboral en la era de la tecnología.

Para darse teletrabajo se tienen que dar en forma conjunta estos tres elementos : **trabajo a distancia**, mediante la utilización de las **telecomunicaciones** y **por cuenta ajena**. Por lo tanto no entran en esta figura los trabajadores independientes que realicen trabajos por su cuenta como micro emprendimientos.

La legislación argentina no lo contempla y no es utilizado en nuestro país con excepción de muy pocos casos.

Para soportar este trabajo se realizó en el mes de enero de 1997 una encuesta a 50 principales empresas de nuestro medio y en sólo una se aplicaba esta metodología.

El teletrabajo presenta pros y contras tanto para,, la empresa como para el empleado

INTRODUCCION

El teletrabajo es un fenómeno de nuestros días producto de la sociedad laboral en la era de la tecnología. Al menos así subtitula su libro Et Teletrabajo, Francisco Ortiz Chaparro'

Este autor llama a nuestra sociedad la sociedad de **la información, y es** absolutamente real. Esta es la principal característica que diferencia esta época de otras anteriores y si bien ha sido un proceso gradual ha tenido una fuerte explosión en los últimos años a partir de la generalización de las : computadoras personales de uso hogareño y lo, que es mas interesante aún, la masificación he esta herramienta en las empresas que las han visto florecer como hongos con la sola conexión a una red que las comunica entre sí. Y el segundo fenómeno que nos invade en estos días es la Internet.

Continuando con el autor mencionado, en **esta nueva sociedad la información sustituye a los antiguos factores de producción y de creación de riqueza. El trabajo manual cede su puesto al trabajo intelectual. El poder se basa cada vez menos en parámetros físicos y materiales (el territorio, el poder militar) y se encarna en la capacidad para almacenar, gestionar, distribuir y crear información.**

Nuestro referente se refiere a la **automatización**, diciendo que la **mayor parte de las tareas laborales están automatizadas y se definen no en términos de fabricación, sino en términos de recogida de información, solución de problemas, producción de ideas creativas y capacidad de responder flexiblemente a situaciones nuevas o de actuar flexiblemente cuando se interactúa con otros. El trabajo puramente mecánico lo hacen las máquinas.**

La globalización. Para Ortiz Chaparro una sociedad de la información tiene una vocación de sociedad global. Esta globalización es fácilmente medible, antes sólo se podía comprar cosas en el mercado local, y hoy con un teléfono y con una tarjeta de crédito se puede comprar cualquier cosa de cualquier parte del planeta, Las comunicaciones han borrado las fronteras de todo tipo, geográficas, étnicas o religiosas.

¿ QUE ES EL TELETRABAJO ?

El teletrabajo **es trabajo** (entendiendo por trabajo aquel que se hace para un empleador), parece una cosa obvia pero es la mejor definición pues se confunde habitualmente teletrabajo con trabajos **free lance** u otras variantes en esa dirección. Pero el segundo elemento que caracteriza el teletrabajo es que

' El teletrabajo. una nueva sociedad laboral en la era de la tecnología. Serie McGraw-Hill de Management. España. 1996

es un trabajo a distancia, y el tercer elemento es que utiliza las telecomunicaciones.

Es decir que implica necesariamente la transmisión del producto del trabajo por un medio de comunicación. También en ocasiones implica la transmisión por parte del empleador de los datos e informaciones útiles para el trabajo.

No importa cuales sean esos medios de comunicación, puede ser un teléfono o un fax o medios más sofisticados, como redes locales o satélites, correo electrónico, video conferencias o cualquier otra cosa no conocida aún.

Las tecnologías de información no sólo presuponen información como un enlace con su empleador, sino un elemento fundamental en la tarea a realizar, como consultar base de datos, navegar por la información, relacionarse con personas de cualquier parte del mundo y de un mismo equipo de trabajo o compañía.

Como ya habrán deducido este factor de la telecomunicación está íntimamente ligado a la distancia.

El trabajo a distancia sin que medie las telecomunicaciones no difiere del trabajo a domicilio tradicional que ha existido siempre. Por ejemplo las costureras a façon de cualquier fábrica de ropa. Pero el concepto es más amplio aún, porque no hay teletrabajo aunque exista la informática si no existe la telecomunicación.

Un resumen de lo dicho hasta aquí sería :

Teletrabajo = trabajo a distancia + telecomunicaciones + por cuenta ajena

EJEMPLOS DE TAREAS FACTIBLES DE HACERSE A TRAVES DEL TELETRABAJO

Veamos primero qué cosas deben darse para caracterizar este tipo de tareas :

- => La tarea a realizar debe ser factible de cuantificar e identificar claramente.
- => Espacio físico reducido (habitat)
- => Cierta rutina.
- => Individual y de bajo nivel de comunicación con otros.
- => Poco herramental y de bajo costo.
- => Plazos, fechas ciertas o fáciles de determinar.

=> Factible de descentralizarse (desde el punto de vista del empleador)

Ejemplos prácticos donde se puede aplicar teletrabajo :

=> Ventas de todo tipo : servicios, venta directa y otras.

=> Utilización de base de datos de una oficina central para cualquier tipo de actividades derivadas

=> *Data entry* (ingreso de datos desde la fuente en soporte generalmente de p a p e l)

0 Programación de computadoras

0 Trabajos relacionados con periodismo, redacción, edición de libros y revistas y tareas conexas.

0 Producción de programas periodísticos de cualquier medio : televisión, radio.

0 Distintas especialidades en relación con procesadores de textos y otros trabajos de secretaría administrativa.

0 Research (investigación) de cualquier tipo de cosas

0 Diversas actividades profesionales : traductores, abogados, arquitectos, psicólogos, sociólogos, economistas, consultores de diferentes extracciones; contadores, entre otros.

0 Capacitación de adultos o específica en diversos temas.

COMO COMENZAR UNA EXPERIENCIA DE TELETRABAJO ?

Como tantas otras cosas con una decisión consensuada en la máxima dirección de la compañía. A partir de allí aplicar los pasos de implementación :

Elegir las tareas apropiadas, y luego - y muy importante - plantearse quienes pueden ser los empleados a seleccionar para asignarles el teletrabajo. No todos los empleados sirven para teletrabajar y no a todos les puede gustar hacer la experiencia. La fuerza de voluntad del teletrabajador es la llave del éxito de este tipo de programas, por ello la correcta elección de los que la llevarán a cabo es de vital importancia.

MARCO NECESARIO PARA UNA EXITOSA EXPERIENCIA DE TELETRABAJO

LOS RECURSOS HUMANOS

Deben converger varias cosas al mismo tiempo y la mayoría de ellas giran en torno a los recursos humanos intervinientes.

Según Ortiz Chaparro alguna de las **características necesarias para serie buen teletrabajador** son :

1. Responsabilidad y madurez
2. Capacidad para organizarse
3. Capacidad como emprendedores
4. Confianza en el trabajador, su inteligencia y su profesionalidad
5. Capacidad de comunicación
6. Capacidad de adaptación
7. Con un ambiente familiar y domicilio adecuados

A su vez para que un teletrabajador pueda trabajar a gusto, continúa Ortiz Chaparro es condición sine qua *non* :

- a) Espacio
- b) Potencia eléctrica
- c) Calefacción, frío y ventilación
- d) La iluminación
- e) Tranquilidad

¿ Qué características personales debe tener un teletrabajador ?

- 1.. Capacidad de organizar su, tiempo '
2. Capacidad de planificación
3. Autodisciplina
4. Capacidad para soportar el aislamiento
5. Capacidad para seguir un horario
6. Capacidad para separar vida familiar y trabajo
7. Capacidad de comunicación por teléfono
8. Poca necesidad de contactos sociales
9. Capacidad para combinar trabajo y ocio

Y no requiere en especial :

1. Capacidad de supervisión
2. Capacidad de trabajo en equipo

Es decir que requiere hasta un diferente enfoque del tradicional para evaluar las potencialidades de un trabajador.

A su vez es clave de un proyecto de teletrabajo la correcta elección del jefe de este tipo de empleados.

LA RELACION LABORAL

Aspectos especiales a tener en cuenta en una relación laboral dentro del teletrabajo :

El lugar de trabajo : una parte de la vivienda debe ser reservada para la actividad profesional que el trabajador se compromete a mantener limpia y en condiciones de visitabilidad como si se tratara de un despacho de una oficina. Y todo cambio de lugar de trabajo debe ser comunicado al empleador.

Equipo y útiles de trabajo : los equipos y materiales necesarios para el ejercicio de la actividad de teletrabajo serán suministrados por la empresa. Estos continúan siendo propiedad de la empresa, El reemplazo de los mismos cuando sea necesario está a cargo de la empresa así como también el mantenimiento y traslado de los mismos.

Desplazamientos : el trabajador participará regularmente en las reuniones de información y de trabajo exigidas por el cumplimiento de su tarea. Se estima una vez por mes.

Indemnización por gastos de teletrabajo.

Duración de la relación de teletrabajo : si bien es variable en cada caso debería contemplar la vuelta del trabajador al esquema anterior. Porque el trabajador llegue a la conclusión que no le conviene o interesa la experiencia o bien, porque la empresa llega a la conclusión que el trabajador no es apto para trabajar en su casa.

Gastos extras ocasionados por el teletrabajo : por ejemplo más gastos de luz, de teléfono, de calefacción deben ser abonados por la empresa, la que podrá pedir un rendimiento de estos gastos con los recibos correspondientes.

El salario : en general no cambia. Hay compañías que en vez de pedir a sus empleados que rindan los gastos extras originados en el teletrabajo prefieren abonar un plus que compense estos gastos. Hay casos donde por el contrario se le abona al empleado un salario menor, porque el teletrabajo es voluntario y se considera en realidad un premio. Otro factor que esgrimen los que abogan por pagar menos es que el teletrabajador tiene menos gastos en transporte ropa y comidas.

No debería significar una pérdida de las seguridades sociales que el empleado ya tenía.

El control de teletrabajador : no es posible llevar a cabo ningún control horario sobre el empleado, no puede establecerle un horario ni un marco temporal. El empleado es a su vez responsable de su propio equipo y herramientas de trabajo.

La seguridad y protección de los datos : esto es vital para la empresa. El teletrabajador debe comprometerse a resguardar la información.

EXPERIENCIAS EN ARGENTINA

A nivel de la calle hay mucha confusión sobre qué es teletrabajo, se confunde con trabajo *free lance*, micro emprendimiento o bien trabajar desde la propia casa - por cuenta propia - utilizando o no las telecomunicaciones. La realidad - por otra parte - es que la legislación argentina no contempla al teletrabajo: *Con lo que podría asimilarse es con el trabajo a façon, como el de costureras o similares, perfectamente medible, . . . hay jurisprudencia al respecto.. -esto me respondió el Doctor Ernesto Martorell cuando lo consulté por la legislación existente sobre el tema, confirmando lo que yo misma recordaba sobre el particular.*

Nuestro *feeling* hasta aquí era que en Argentina es casi inexistente la aplicación del teletrabajo, pero para no basarnos en nuestras impresiones realizamos en enero de 1997 una encuesta² sobre un universo de alrededor de 50 de las principales empresas industriales, comerciales y de servicios, la única respuesta positiva se corresponde con el caso que relataremos más adelante. Esta empresa tiene un caso de teletrabajador que como ellos mismos dicen lo han implementado *a la argentina*,

UNA EXPERIENCIA AISLADA

Las experiencias en Argentina son muy pero muy pocas. Voy a relatar una que además de ser en nuestro país servirá para ilustrar una experiencia real de teletrabajo.

A partir de aquí voy a contarles la historia de María; que fue una de las *jóvenes top* entrevistada en mi libro *Las Puertas del Trabajo*, antes de ser una teletrabajadora

Hagamos un poco de historia para entender mejor el caso planteado. María trabajaba en una empresa localizada en el cordón industrial del Gran Buenos Aires que tiene con varias fábricas en distintas ciudades del interior del país. Ella era allí Responsable de Desarrollo de Recursos Humanos. Por una situación familiar debe trasladarse a una ciudad del interior. Y afrontar así la enorme disyuntiva entre su carrera y la situación familiar planteada: Frente a

²Top Management, Buenos Aires, enero de 1997. Se enviaron 57 formularios, 7 de las empresas no habían contestado a la fecha de la realización de este trabajo, 49 contestaron en forma negativa, 1 en forma positiva. De los que respondieron negativamente : 1 desconocía por completo la existencia del teletrabajo, 1 está pensando en implementarlo, 5 contestaron que habían investigado y no está contemplado en sus respectivos convenios colectivos. La mayoría pidió copia del resultado de la encuesta porque estaban muy interesados.

esta opción se comienza a pergeñar la posibilidad de un trabajo virtual y lo diseñan en conjunto, María y sus jefes.

Ella prepara en uno de los cuartos de su casa una oficina, compra una PC, un fax y se instala. De ello resulta que, a distancia, María trabaja por proyectos: planes de capacitación, programas de inducción, programas para jóvenes profesionales y otros temas similares. El único compromiso que debe asumir frente a sus empleadores - además de entregar su trabajo en tiempo y forma - es una semana por mes debe estar físicamente en las oficinas de la empresa.

¿Cómo es la adaptación? Confiesa que muy difícil. Fue difícil armar la rutina horaria, similar a un horario preestablecido, que ella modifica de acuerdo a sus necesidades pero que de algún modo se impone a si misma para sentir la presión de la responsabilidad. Fue difícil darse vuelta y no tener a nadie para hacer un comentario sobre cualquier tema, desde ya estaba el teléfono y todos sus compañeros detrás... pero a *miles o cientos de kilómetros de distancia*, según el caso.:

Hoy ya pasaron varios meses y la adaptación está superada. Ella más que feliz porque está equilibrando diferentes aspectos de su vida. No fue necesario posponer cosas y desde su casa sigue conectada con su profesión. Para ella esto fue la solución a sus problemas.

¿Cómo funciona en este caso el teletrabajo? Lo explicaré con un ejemplo: confección de **un plan de capacitación**. En una primera instancia se hace un relevamiento de necesidades, esto puede hacerse por escrito o por teléfono, luego se confecciona el mismo, en esta etapa se trabaja a solas con una PC y luego la presentación se puede hacer también por algún medio escrito vía Fax o módem o una carpeta por correo. ¿Qué participación se requiere en forma **perSOnal y no virtual?** En ocasiones, por ejemplo: debe viajar a reuniones a las diferentes fabricas cuando se inicia la implantación de algún programa.

¿Cómo se ve esta relación laboral según el ángulo del cual se mira?

Para la empresa: no existe una diferencia notoria en el rendimiento de este profesional con el de otro, haciendo la misma tarea, pero en las oficinas centrales.

Para el interesado: le permitió resolver un tema familiar que es quizás transitorio y de ese modo darle continuidad a su carrera laboral.

EN EL MERCADO EDITORIAL

Pero continuando con mi investigación tomé contacto con una de las principales editoriales del país. Mi interlocutor, el gerente de recursos humanos de la empresa me comenta que el Estatuto del Periodista Profesional contempla tres categorías de trabajadores :

1. Profesional en relación de dependencia : trabaja en las oficinas y cumple horario
2. Colaborador : tiene relación de dependencia pero no asiste a las oficinas y no cumple horario. En esta empresa por una interpretación del mencionado estatuto estos trabajadores no tienen vacaciones. La explicación es que no tienen vacaciones porque no cumplen horario. Hasta aquí esta figura *pareciera* que estaría inserta en la modalidad de teletrabajo pero no es así porque no lo hacen a través de las telecomunicaciones. Ellos deben llevar los trabajos personalmente para discutir con los, respectivos jefes el contenido de las notas. Nuestro entrevistado dice creo que en *poco tiempo esta figura del colaborador ingresará a la de teletrabajo.*
3. Colaborador *free lance* : hace menos de 24 colaboraciones, al año, no tiene relación de dependencia y factura honorarios por cada nota.

Nuestro entrevistado comenta *tenemos un caso en España, nuestro corresponsal allí es un empleado nuestro y envía su información por fax o utilizando un modem. Pero internamente no lo consideramos teletrabajo. Dentro del estatuto se incluye en la categoría 2.*

VENTAJAS Y DESVENTAJAS DEL TELETRABAJO PARA LA EMPRESA Y PARA EL TRABAJADOR

Como en cualquier otra cosa el teletrabajo tiene pros y contras, no es la solución de todos los problemas ni el trabajo del futuro. Debemos realizar sobre este aspecto de la vida moderna el análisis más objetivo que sea posible.

VENTAJAS

Para la empresa

· Ahorros económicos de espacio físico y todo lo que de allí se deriva : luz, teléfono, etc

· Disminuyen los problemas de ausentismo

· Realiza una gestión por resultados.

· Aumento de la productividad.

Para el empleado

· Soluciona ciertos problemas de la vida da familia y permite un contacto más directo con ella

Si el trabajador tiene una buena productividad puede tener más tiempo libre. Más libertad, real y sensación de...

Tiene la posibilidad de tener menos problemas laborales producto de la interacción con otros compañeros de labor.

Menor gastos : de ropa, de transporte, etc. Esto será en función del salario pactado.

INCONVENIENTES

Para la empresa

En la etapa inicial, el costo de los equipos y otras inversiones.

Imposibilidad de un control por presencia del empleado.

Mayor dificultad del trabajo en equipo.

Pérdida paulatina - gradual - de la identificación; del empleado con la organización.

Para el empleado

Aislamiento y sensación de pérdida de *status*.

Pérdida de uno de los principales beneficios del trabajo : la socialización

Pérdida de la guía en la carrera laboral (del marco de referencia)

Afrontar la primer etapa de cambio a teletrabajador

Falta de apoyo de compañeros y referentes (no tiene a quién preguntarle una duda)

Desvinculación paulatina de la empresa (pérdida de identificación, de la *camiseta*)

En un artículo que me publicara el diario EL Cronista titulado *Beneficios del trabajo en la era del trabajo virtual* comento que *comenzaron a verse los problemas sociales que el mismo implica, en un principio - y como sucede habitualmente - sólo se veían los beneficios.*

El señor o señora en cómodos pijamas o pantuflas, si así lo desean. La tarea es la misma, solo que las instrucciones previas y el trabajo terminado, todo va y viene magnéticamente.

En el último terremoto de Los Angeles. trajo sus beneficios pues” estas terminales remotas, en los domicilios de los distintos empleados, permitieron seguir operando aunque las autopistas estuviesen rotas.

Pero todo no termina allí, porque el trabajo trae beneficios de tipo social al trabajador. Todos habremos vivido la experiencia de cuando ingresa una chica mona a un trabajo y, además, bien vestida, después de las lógicas envidias que pueda provocar en algunas, esto trae una nivelación para arriba, pues lo natural es que todas las demás chicas se esfuercen por mejorar su estilo. Esto es el resultado positivo de la competencia.

Dentro de los beneficios sociales está en primer lugar el mencionado un poco graciosamente más arriba. La autoestima se ve fortalecida por el trabajo grupal. En el caso específico de las mujeres hay una enorme diferencia a favor de las que trabajan en cuanto a mantenerse más jóvenes, más delgadas, más actuales que la señora que se queda en su casa aunque ésta tenga la oportunidad de llevar una vida sin tanto *stress*.

La socialización en sí misma es un valor pues una persona socialmente contenida puede manejar mejor todos los eventuales problemas con que deba enfrentarse.

La interacción con pares, inferiores y superiores, es enriquecedora para la gestión diaria tanto en lo laboral como en lo atinente a otros elementos. En contraposición una persona que trabaja en forma aislada tiende a contraer su carácter, a desestimularse. La falta de competencia es desmotivante pues si bien en una cantidad excesiva es mala la ausencia total de ella lo es también.

Pero porqué si es tan beneficioso el trabajo, en un lugar de trabajo surge la idea del trabajo virtual y además se le ven o vieron beneficios? Pues como todas las cosas tienen sus puntos a favor y en contra. A favor y en función del empleado : resuelve el problema de aquellas personas que “deban cuidar a otra, por ejemplo una mamá con niños pequeños o alguien que deba cuidar a una persona mayor o a un perro. Resuelve los problemas de distancia y otros derivados de un traslado. A favor y en función del empleador: no necesita prever el puesto de trabajo, no ” consume ” ni café ni té, no se distrae con otras personas ni se sindicaliza, no es factible que forme grupos - negativamente - con otras personas.

³El Cronista. 27 de junio de 1995

Las contras son también para unos y otros. En contra y en función del empleado: la más importante, pierde marco de referencia, pierde la pertenencia a un grupo y a una organización, además, y de alguna manera ya mencionada, la falta de incentivo personal derivada de la sana competencia que plantean los grupos humanos. En contra y en función del empleador: algunas - aunque parezca un contrasentido - son las mismas que para el empleado, por ejemplo pertenencia a la organización y marco de referencia. La falta de identificación trae como consecuencia - no necesariamente, pero puede ser - falta de fidelidad o compromiso, cierto "mercenarismo", quiero decir con esto que a la persona le da lo mismo trabajar para la empresa A o B, pues no se siente parte de ella.

Lo interesante que supone - al menos es mi opinión - el análisis de estos temas es que las tendencias que ya fueron probadas en otros países deben ser analizadas en todas sus facetas, y no importarlas a paquete cerrado.

En otro artículo, que también me publicara el matutino *El Cronista* y que el mismo diario titulara *Al abordaje de la Red*⁴ e ilustrara con una imagen de la película por mí mencionada en la nota comento nuevamente los beneficios y las contras del trabajo virtual en un momento en que se comenzó a hablar mucho del tema y que en revistas empresarias se publicaron notas confundiendo teletrabajo o trabajo virtual con micro emprendimiento y trabajos *free lance*. Algunos párrafos de ese artículo incluyo a continuación :

Cuando ví la película *La Red* - donde una activa Sandra Bullock interpreta a Angela Bennett, una joven **que casi pierde su identidad por tener un trabajo virtual**, más otras cosas que pasan en la película, pero no las voy contar aquí porque me alejo del motivo central - reflexioné sobre este tipo de inconveniente que tiene el teletrabajo.

Angela tiene un trabajo muy interesante, muy técnico y ella es considerada una número uno en su especialidad. Pero desde el punto de vista personal y *Social* ella es una ermitaña, solo se relaciona con computadoras manejadas por gente que no se conoce entre sí, se comunica con la pizzería a través de una PC y cuando va a la playa lo hace en compañía de *su notebook*. Vive permanentemente conectada a una red y ese es todo su mundo. El trabajo, virtual la ha transformado en una solitaria con muchos problemas para relacionarse no sólo con el sexo opuesto sino para relacionarse en general. Hasta aquí todo es una película...

SÍNTESIS

⁴El Cronista, Suplemento Multimedia, 20 de enero de 1996

Guy Sorman ⁵ da una interesante definición del progreso. *El progreso no significa felicidad, a pesar de que algunos filósofos y economistas liberales norteamericanos sientan la tentación de fusionar ambos conceptos. El progreso es material y cuantificable ; la felicidad es incuantificable e incognoscible. Aún así, podemos evitar tender puentes entre uno y otra. Sabemos que una de las medidas menos cuestionables del progreso es la constante disminución de la mortalidad infantil. Y siempre he pensado que para una madre, poseer la certidumbre de que la casi totalidad de sus hijos sobrevivirá al parto tal vez no sea la felicidad en sí, pero ciertamente es condición de una mayor felicidad.*

De manera arbitraria, defino entonces el progreso como cuantificable y como uno de los componentes de la felicidad individual.

Conocer el progreso, medirlo sin tener que creer en él, implica su reversibilidad.

¿Porqué cito aquí a Guy Sorman? Porque pienso que no hay que entusiasmarse con todo lo que significa progreso en el concepto neto de la palabra y hay que tener en cuenta el ser humano como centro de la preocupación social y no la *máquina* como centro.

Por ello creo que es de vital importancia analizar objetivamente los pro y contra que expusimos más arriba y sacar a partir de allí sus propias conclusiones.

El teletrabajo es la solución del futuro?

Creo que no. Como tantas otras cosas debe administrarse cuidadosamente. Por un lado trae muchas ventajas para unos y otros, pero esto puede ser así sólo por períodos. Tiene altos costos personales derivados del aislamiento. Como le pasaba a Angela, en la película, que se había convertido en una persona totalmente desvinculada de los afectos y, aún sin llegar a esos extremos, siempre tiene para el trabajador virtual un costo social derivado del hecho de no tener el marco de referencia que representa el contexto laboral.

Por lo tanto el tema debería encararse desde la empresa tomando grupos móviles de trabajo virtual, es decir personas que alternen períodos de trabajo virtual con períodos de trabajo tradicional.

Y para las personas vemos el tema del mismo modo, es decir, recurrir al trabajo virtual en determinados momentos de la existencia, como el caso relatado de María y no transformarlo en un sistema natural de vida.

MAA, 31-1 -97

⁵ La singularidad francesa. Guy Sorman. Editorial Andrés Bello, 1996

Ambiente basado en Hipermedios como soporte al Aprendizaje de la Tecnología de Objetos

BERTONE, E. ; LAFUENTE, G; NICOLAU, S; OLSINA, L.
Grupo de I&D en Ingeniería de Software.
Departamento de Computación. Facultad de Ingeniería. UNLPam.
e-mail [olsinal, snicolau, lafuente,ebettone]@unlpin.edu.ar. TelFax 0302 24711

Resumen

En esta comunicación describimos las características esenciales de un ambiente basado en hipermedios que puede ser utilizado como soporte en el aprendizaje interactivo de la Tecnología de Objetos. Intentamos mejorar los mecanismos alternativos de transmisión de conocimientos en el área, explotando los atributos de interactividad, didáctica y atracción mediante el uso racional de multimedios y navegación hipermedial.

Además en este proyecto, como en otros anteriores, estamos experimentando un Modelo de Proceso Flexible y estrategias innovadoras para la especificación, diseño e implementación de aplicaciones de autoría y en Internet.

Áreas Claves: Educación, Hipermedios, Ingeniería de Software.

1. Introducción

El uso de hipermedios para el desarrollo de aplicaciones de diversa índole está ganando cada vez mayor espacio en el mundo de la informática y el área de la Educación se puede favorecer por el empleo oportuno y racional de estas tecnologías innovadoras de manejo de información.

Por otra parte la incorporación de los Sistemas basados en la Tecnología de Objetos ha producido un cambio paradigmático evolutivo comparado con el enfoque estructurado y debe, por todos los medios, ser entendido. No sólo implica la comprensión de un conjunto de conceptos como descomposición objetual del dominio del problema, abstracción, encapsulamiento, herencia, polimorfismo, colaboraciones, etc., sino también la comprensión de ambientes de desarrollo, metodologías de análisis y diseño y modelos de proceso que soporten todo el ciclo de desarrollo y extensión. Sin duda que la administración de proyectos OO ha dejado en evidencia que se requiere mayor experticia en este campo [Sadr96].

Pensamos que el trabajo de los docentes en la enseñanza de los conceptos, ambientes, metodologías y estrategias de proceso OO se puede complementar mediante el uso de un producto educativo integrado, interactivo; que soporte la navegación libre o asistida en una red de conceptos racionalmente estructurados y presentados en múltiples medios. Nuestro proyecto, en fase de desarrollo, tiene primordialmente una doble fundamentación de relevancia: tecnológica y educativa. Desde el punto de vista tecnológico y apoyándose en principios y modelos científicos, nos propusimos encarar un área temática de actualidad y no explotado en su totalidad

que consiste en comprender los ítems antes citados necesarios para llevar a cabo un proyecto de desarrollo de software bajo el paradigma de Objetos. Desde el punto de vista educativo y didáctico pensamos que para transmitir los conceptos y constructores de proceso es conveniente usar tecnología hipermedial ya que esta favorece, según nuestra hipótesis; al proceso de aprendizaje y comunicación de un modo dinámico y atractivo. No es nuestra intención reemplazar el rol del profesor en dicho proceso sino que el ambiente debe ser utilizado como soporte al mismo.

A seguir organizamos este trabajo en dos partes principales. En la sección dos expondremos los contextos temáticos principales de la aplicación educativa y un conjunto de estructuras de acceso reusables y en la sección tres discutiremos aspectos del modelo de proceso hipermedial utilizado para el desarrollo de productos de software de la citada aplicación. En la cuarta parte concluiremos con algunas reflexiones y futuros avances.

2. Componente de la Aplicación Educativa.

La aplicación está estructurada en seis contextos temáticos fundamentales. Un contexto temático es un contexto de navegación el cual es una primitiva de diseño, hipermedial. Un contexto de navegación está compuesto esencialmente por clases navegacionales como nodos, enlaces, anchors y otros contextos. Representa una unidad o espacio de información semánticamente cohesiva [Schwabe96] y navegable de un modo no secuencial.

Por otra parte la aplicación educativa incorpora una colección de estructuras de acceso, como índices globales a todos los contextos y locales a un contexto, diccionario y referencias y otras facilidades como cuestionarios, anotadores y ayudas. Como se aprecia en la figura la hemos agrupado los siguientes contextos de navegación a un mismo nivel (describiremos asimismo algún contexto anidado), a saber:

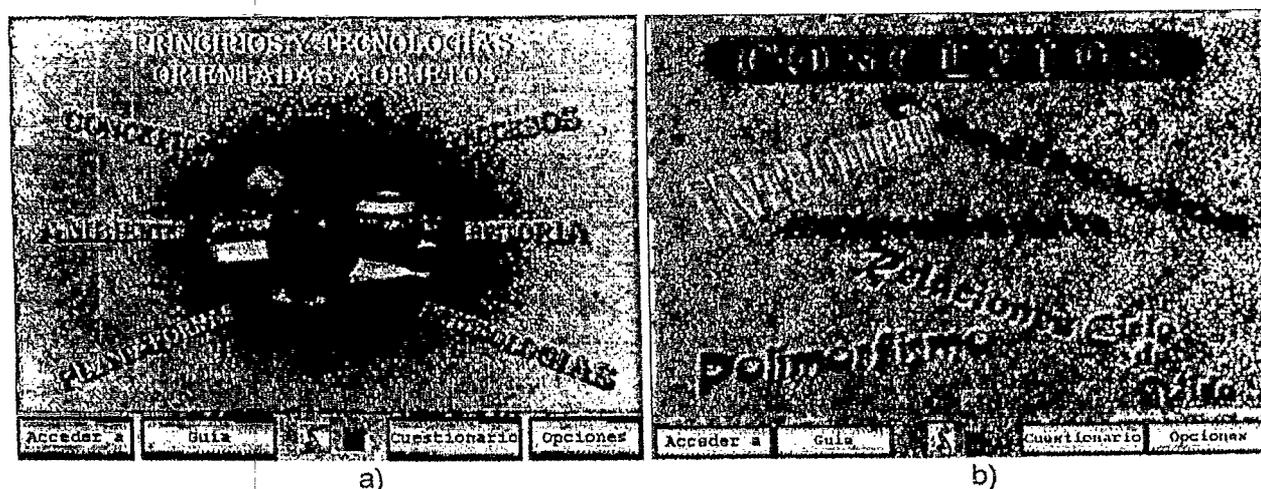


Fig. 1. a) Nodo superior en la jerarquía de la aplicación hipermedial; b) Pantalla que aparece luego de clicar el botón "Conceptos" del nodo anterior

- . Conceptos: contiene los principios y contenidos básicos de la tecnología de objetos, en donde el usuario puede navegar en modalidad asistida o libre y está presentado de manera independiente de lenguajes o entornos de trabajo.
- *Ambientes de Desarrollo*: involucra las características esenciales de los siguientes ambientes integrados: C++, Java, Smalltalk, Toolbook. Se incorpora características sintácticas de cada lenguaje para conceptos como clase, mecanismos de herencia, etc.; y una tabla comparativa de características (por ej. si el lenguaje soporta asociación dinámica, si es tipado, si soporta herencia múltiple, recolección de residuos, clases parametrizadas, etc.). Por otra parte, se presenta una demo para cada ambiente y el alumno puede introducirse en la herramienta, y, en colaboración con el docente, puede realizar los ejercicios propuestos.
- *Plataformas*: incluye el estado del arte de los Sistemas Operativos OO y capas de soporte para ambientes distribuidos (como CORBA, etc) .
- . *Modelos de Proceso*: definimos a un modelo de proceso de desarrollo de software como a la representación abstracta de las fases, tareas y actividades; el modo en que las mismas se interrelacionan, se ejecutan, se controlan y se les asignan recursos, con el fin de crear y evolucionar los artefactos de software para satisfacer necesidades de los usuarios.
- *Historia*: representa una línea de tiempo respecto de los diferentes constructores del paradigma (como lenguajes, metodologías, etc.)
- . *Metodologías*: incluye material sobre la notación y la semántica de las metodologías mas importantes como la metodología de Booch [Booch94], Rumbaugh [Rumbaugh 91] y el Modelo de Objetos Unificado.

En la figura 1 b vemos el primer nodo destino del enlace cuyo punto de partida es el de "Conceptos". Este nodo contiene contextos de navegación que representan a los principios y contenidos básicos de la orientación a objetos, en donde el usuario puede navegar en modalidad asistida o libre y está presentado de manera independiente de lenguajes o entornos de trabajo [Amandi94].

Entre las unidades navegables que el estudiante puede recorrer se encuentran: "Introducción", "Clasificación", "Encapsulamiento", "Relaciones", "Polimorfismo", "ciclo de vida de un Objeto". Al ingresar al nodo de "Conceptos", una voz (activable o desactivable por medio del icono visible en la parte inferior de la fig. 1b) nos sugiere que el recorrido se puede efectuar de un modo libre o asistido. Seamos la opción, y, si elegimos por ejemplo el modo asistido, el ambiente no nos permite estudiar acerca del concepto de "Encapsulamiento" si previamente no pasamos por la unidad "introducción". Lo cual es lógico que se aprenda el concepto de clase antes que el concepto de representación interna de los miembros de la misma; con más fuerza, para conceptos como relaciones de herencia y agregación o conceptos como polimorfismo. (El estudiante opcionalmente puede establecer niveles de aprendizaje y Pasar a la siguiente unidad si solamente obtuvo cierto puntaje en el ítem previo).

En la figura 2 podemos apreciar un nodo del contexto de navegación denominado "Introducción" a partir del cual realizaremos un conjunto de observaciones válidas para otras unidades de información. En primer lugar disponemos de dos

mecanismos de navegación dentro del contexto: por medio de los controles de navegación direccional ubicados en la parte inferior-central (los mismos permiten ir al menú principal, ir al nodo inmediato superior, recorrer con un comportamiento paso-a-paso el nodo anterior o el nodo siguiente y, por último, recorrer todos los nodos visitados); el otro mecanismo para recorrer los nodos es por medio de un acceso directo que en la figura está indicado como “índice Temático del Contexto Introducción”. Este mecanismo, además de ofrecer un comportamiento de acceso directo, facilita la orientación del usuario al indicar jerárquicamente el camino recorrido [Thüring95] (por ejemplo, se indica que el estudiante está en “Conceptos” e “Introducción” y en el tema “Objetos”).

Por otra parte contamos con cinco controles, como se aprecia en la parte inferior. El botón “Acceder a”, el icono que maximiza o minimiza el Índice previamente descrito, el icono de voz, del cual se obtiene sugerencias, el icono del anotador contextual, y el botón de “Opciones” del cual se puede activar servicios de ayuda, de impresión de información, etc,



Fig. 2. Un nodo del contexto navegacionat “introducción” dentro del contexto “Conceptos”.

El botón ‘(Acceder a’ está visible en todo momento y en todos los nodos de la aplicación, al igual que el botón de “Opciones”. Hemos diseñado e implementado un conjunto de estructuras reusables como son: índice General, Referencias, Diccionario o Glosario, Anotador, Cuestionario y otras. En la figura 3 apreciamos el nodo de referencias y en la figura 4 el visor del Anotador contextual.

Además en la figura 2 observamos, en el campo de texto que describe conceptos sobre el tópico “Objetos”, a un conjunto de iconos por medio de los cuales se pueden activar sonidos, animaciones, videos, llamar a otros entornos de trabajo o aplicaciones demostrativas. Como resultado de haber presionado un botón apareció la foto de la izquierda con una animación en la que se va indicando los distintos objetos

del mundo real. En tanto se despliega una animación el usuario puede encontrarse con nuevas opciones, que eventualmente puede explorar.

Al visor de Referencias (fig. 3) se puede llegar a través del botón "Acceder a", o a través de una referencia bibliográfica (o de otro tipo) en el texto del tema que el estudiante está visitando. En el primer caso aparece el nodo y el estudiante debe apretar el botón de la letra. En la lista deslizable verticalmente ubicada a la derecha, aparecen todas las referencias para la letra seleccionada y la información asociada a la elección (por ej- para Booch94) que se muestra en los campos de texto superiores. En el caso en que el estudiante presione el ancho de referencia en el hipertexto, se muestra el nodo directamente con la información asociada,



Fig. 3. Visor de la herramienta para acceder a las referencias globales.

Por último, queremos describir algunas características del Anotador contextual. En la figura 4 podemos observar al visor del anotador conteniendo tantas paginas por contexto temático con el fin de recordar, en tanto se esta en proceso de aprendizaje, aquellas cuestiones que serán de consideración posterior,

Decimos que el anotador es contextual porque al ingresar en un contexto temático, y si se activa esta herramienta, aparecer8 las anotaciones anteriores para ese contexto. Es más, como los contextos pueden ser anidados como vimos previamente con "Conceptos", y si el estudiante se encontrara en un momento dado en el tema "Introducción", al activar el anotador en este punto, aparecerá la información sobre "Conceptos-Introducción". Además, el anotador contiene un conjunto de herramientas para cumplir funciones básicas como copiar, cortar, pegar, establecer tipo de letra, guardar, buscar una palabra, etc. Muchas de estas funciones se pueden ejecutar a través de las opciones de la barra de menú, de la barra de herramientas y,

de un modo rápido, usando el menú desplegable por medio del botón derecho del mouse.

Uno de los objetivos tenidos en cuenta en el diseño e implementación de estas herramientas, como anotador, índices, referencias, etc., fue el criterio de reusabilidad. Hemos construido un conjunto de componentes reusables.

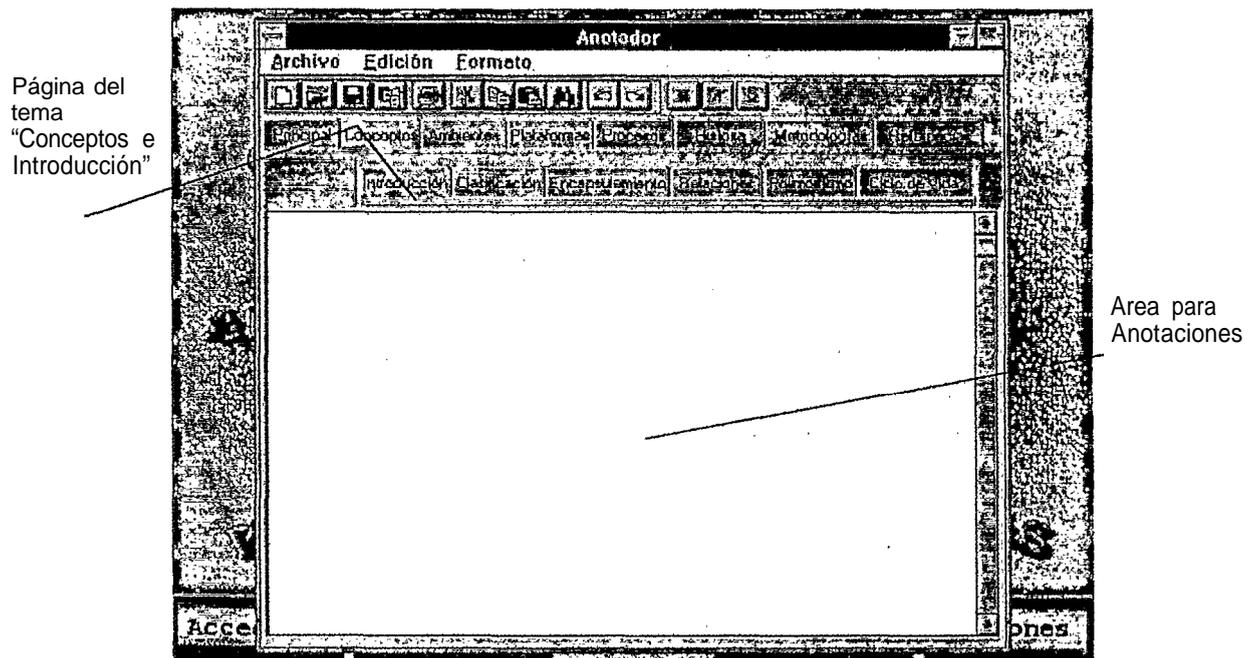


Fig. 4. Visor de la herramienta para acceder a las anotaciones contextuales

3. Proceso de Desarrollo Hipermedial utilizado.

El modelo; de desarrollo utilizado está sustentado en un modelo de proceso flexible (MPF) que soporta todas las tareas y actividades en el ciclo de construcción de artefactos hipermediales [Olsina96b]. En todo momento se está creando o actualizando a algún modelo. Los modelos físicos se construyen principalmente con la estrategia de prototipación flexible orientada a objetos (PFOO). Los modelos lógicos empleados se apoyan en los modelos del método de diseño de hipermedios orientado a objetos (OOHDM) [Rossi95, Schwabe96,] y en el modelo de requerimientos (una adaptación de casos de uso según Jacobson). Asimismo se emplea un modelo de plan, un modelo de aseguramiento de calidad, entre otros.

Este enfoque de modelo de proceso, desde el punto de vista de la fase de desarrollo, permite actividades iterativas, concurrentes y particionamiento de un problema en subproblemas pudiéndose atacar incrementalmente porciones de menor complejidad.

Particularmente estamos empleando este proceso de desarrollo hipermedial en la construcción de los distintos artefactos de esta aplicación educativa (al igual que en el proyecto ya concluido, denominado “*Facultad de Ingeniería*” [Olsina96a]). Por lo tanto, dentro de las especificaciones lógicas nos apoyamos en el modelo de casos de uso para requerimientos, y en los modelos conceptual, navegacional y de interfaces abstractas de OOHDM para análisis del dominio y diseño. Como soporte a las especificaciones y a la experimentación como así también a la etapa de construcción de versiones, empleamos la estrategia de prototipación, flexible.

En la figura 5 mostramos un diagrama reducido de las distintas clases de modelos que emplea el MPF, y, seguidamente describiremos algunas características salientes de los mismos.

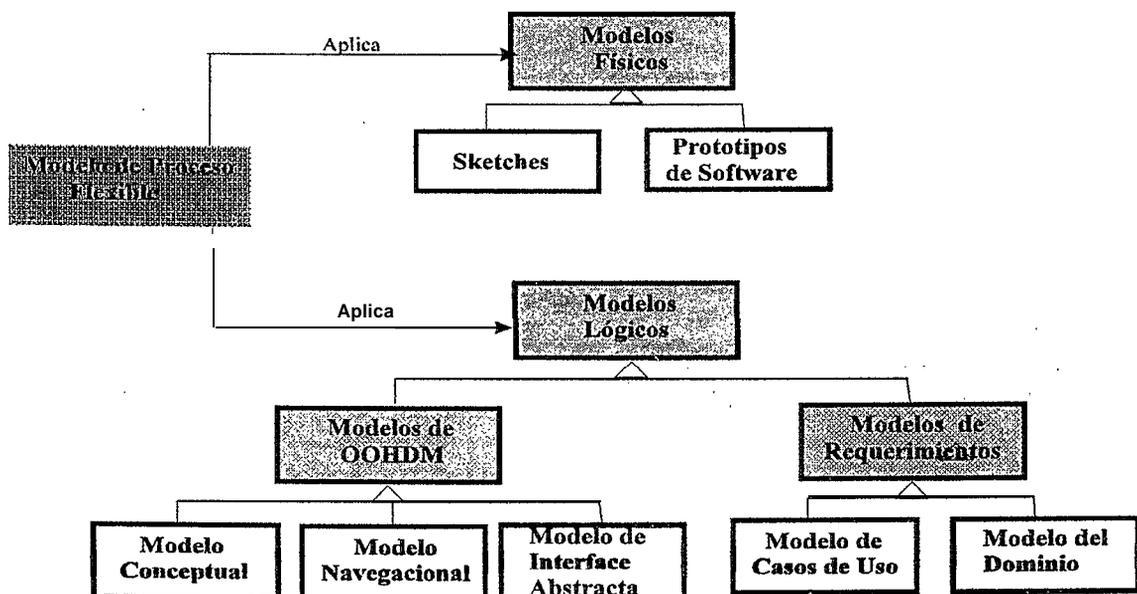


Fig. 5. Diagrama reducido de modelos físicos y lógicos utilizados por el MPF

Modelo de Requerimientos: el modelado de requerimientos tiene dos responsabilidades fundamentales [Jacobson94]:

- *Capturar requerimientos funcionales* [IEEE93]: el modelo de casos de uso define un comportamiento detallado del sistema aunque comprensible por usuarios y desarrolladores. Una propiedad relevante es la de usar un lenguaje de comunicación comprensible por usuarios en general y expertos del dominio. El contexto o entorno se define por la descripción de los diferentes actores. Consecuentemente el modelo de casos de uso no reemplaza al modelo conceptual sino que representa una visión externa del sistema (semejante a la visión de caja negra), en cambio el modelo conceptual representa una visión interna del mismo.

- *Estructurar al modelo de clases y objetos (modelo conceptual) y a otros modelos del sistema en visiones de complejidad manejable.* Podemos encontrar por ejemplo todas las responsabilidades de, una clase analizando a todos los casos de uso en donde esa clase desempeña un rol por otra parte una vez implementado un componente o subsistema ofrece un mecanismo de verificación del modelo físico con el modelo de requerimientos.

Un modelo de casos de uso es un grafo con dos tipos de nodos [Jacobson92, 94]: el nodo actor y el nodo caso de uso. Además existen arcos de comunicación entre ambos nodos. A cada nodo se le asocia una clase. Un nodo actor está conectado al menos a un nodo caso de uso, y este, a su vez, está conectado al menos a un nodo actor a través de un arco de comunicación. Una instancia de una clase actor puede crear instancias de una clase de caso de uso. Un arco de comunicación entre un nodo actor y un nodo caso de uso significa que se puede enviar un estímulo entre instancias de la clase actor e instancias de la clase casos de uso.

Los actores son objetos que se encuentran fuera del sistema a modelar en tanto que los casos de uso son objetos que se encuentran dentro del sistema. Los actores representan entes externos que tienen necesidad de intercambiar información con el sistema. Es importante resaltar que los actores son instancias de clases mientras que los usuarios son un tipo de recurso que implementa estas instancias.

Cuando un actor usa al sistema el mismo realiza un caso de uso. La colección de casos de uso es la funcionalidad completa del sistema. Por lo tanto una instancia de caso de uso es una secuencia de transacciones realizadas por el sistema, la cual produce un resultado medible para un actor en particular.

Por otra parte, dentro del modelo de dominio, usamos principalmente al glosario. Un glosario sirve para especificar y comunicar en etapas tempranas del desarrollo de software las palabras claves del dominio del problema. Las palabras claves y sus conceptos relacionados se escriben en lenguaje natural, en una lista clasificada. Esta lista de conceptos claves sirve para alimentar al modelo conceptual como medio para encontrar clases, y asimismo sirve para alimentar al modelo navegacional, como medio para encontrar nodos o contextos navegacionales.

Modelo Conceptual: en el modelaje conceptual se especifican y documentan las clases y objetos, los atributos (con sus posibles tipos), las relaciones entre clases y los subsistemas. El objetivo fundamental de esta tarea consiste 'en descubrir y especificar las responsabilidades esenciales y refinar los objetos encontrados. En esta tarea se construye esencialmente diagramas de clases y objetos, y se pueden especificar textualmente en tarjetas semejantes a CRC', las partes salientes de los subsistemas; clases, objetos y relaciones. Así tenemos especificaciones gráficas, visuales y especificaciones textuales.

La diferencia principal entre clases del modelo conceptual y casos de uso del modelo de requerimientos está en que las clases se relacionan con otras clases dentro

CRC se traduce en Clases, Responsabilidades y Colaboraciones.

del mismo sistema, mientras que un caso de uso no se puede comunicar con otros casos de uso sino solamente con actores, fuera del sistema.

Modelo Navegacional: la noción de navegación es una propiedad distintiva de las aplicaciones hipermediales. En 'este proceso de diseño navegacional se debe considerar criterios cognitivos como se apunta en [Nanard95, Schwabe96, Thüring95]; los perfiles de los futuros usuarios; las unidades cohesivas de información a diseñar (contextos navegacionales) considerando las posibilidades y estrategias para recorrerlos.

En la tarea de modelaje navegacional se utilizan primitivas de especificación como nodos, enlaces, contextos y transformaciones navegacionales, estructuras de acceso como índices y visitas guiadas (el lector puede ver en la fig. 1 dos botones con la descripción "Guía" que permiten ingresar a visita guiadas; una visita es global a varios contextos y la otra es local al contexto de "Conceptos"). También se pueden construir tarjetas para especificar a las primitivas semejantes a las descritas previamente.

Modelo de Interfaces Abstractas: en el modelaje de interfaces abstractas nos permite construir diferentes interfaces para un mismo modelo navegacional [Rossi95]. Los aspectos de diseño a tener en cuenta reside en definir qué eventos intervendrán en el lenguaje de acción y qué objetos de interface percibirá el usuario (asociados a alguna metáfora); qué transformaciones de interface y de objetos navegacionales sucederán; como serán sincronizados los objetos multimediales de interface, principalmente los que incorporan audio y video.

Modelos Físicos: en el desarrollo de aplicaciones hipermediales la estrategia de prototipación flexible [Olsina97] soporta tanto el descubrimiento de requerimientos y el aprendizaje conjunto de usuarios y desarrolladores como la exploración de aspectos de arquitecturas de diseño y la construcción de prototipos que pueden evolucionar hasta lograr su funcionalidad bien implementada y validada.

La PFOO hereda características de las estrategias de prototipación rápida-funcional (PRF) y/o de la prototipación evolutiva (PE) como mecanismo para crear y evolucionar modelos físicos.

La estrategia de PRF es altamente iterativa y se aplica para requerimientos funcionales y aspectos de diseño pobremente entendidos o cuestiones comprendidas pero que necesitan validación con el usuario; se puede sacrificar la completitud pero no la correctitud de los requerimientos ni la estética de las interfaces. El prototipo debe ser rápido y fácil de modificar y extender.

En cambio en la estrategia de PE se evolucionan los requerimientos funcionales bien comprendidos y se tiene en cuenta requerimientos no-funcionales como pruebas de performance, de seguridad, etc. La estrategia es menos iterativa que la PRF basada también, en el esquema de iteración/retroalimentación experimental desarrollador-usuario. Por cada ciclo se incrementa al prototipo y se actualizan concurrentemente las especificaciones de los modelos lógicos.

4. Consideraciones Finales.

En el presente trabajo hemos discutido brevemente algunas características esenciales de un ambiente basado en hipermedios a ser utilizado como soporte en el aprendizaje interactivo de la Tecnología de Objetos, y, además, expusimos algunos aspectos del Modelo de Proceso Flexible útil durante el desarrollo de artefactos hipermediales.

Es oportuno realizar un conjunto de consideraciones.

Con respecto al ambiente educativo, que a pesar de estar en fase de construcción, lo hemos utilizado en el aula, esencialmente en lo que se refiere al aprendizaje de los contextos temáticos “*Conceptos*” y “*Ambientes*”, para comprobar el grado de atracción y eficiencia de la herramienta como soporte a la enseñanza. Algunas conclusiones preliminares al trabajar con grupos de estudiantes elegidos al azar, son:

- * favorece el aprendizaje de los conceptos básicos gracias a los atributos de interactividad y atracción que ofrece multimedios y la navegación hipermedial. Es preciso que la aplicación esté racional y cuidadosamente estructurada
- incentiva la curiosidad cognitiva y el aprendizaje exploratorio
- en algunos tópicos (ejemplificación a través del uso de ambientes de implementación) requiere el soporte de los docentes.
- la aplicación educativa hipermedial no reemplaza al docente sino que puede ser un complemento beneficioso
- hicieron poco uso del anotador. En cambio los cuestionarios con niveles de dificultad y preguntas al azar dentro de un contexto eran de su interés.

Por último, y relacionado al proceso de desarrollo de aplicaciones hipermediales podemos concluir:

- cubre todas las fases y tareas esenciales de un proyecto hipermedial
- establece un equilibrio favorable mediante la aplicación sistemática de modelaje físico y lógico
- la clara división de tareas facilita la planificación y el control
- en los proyectos realizados hasta la fecha, el modelo propuesto ha dado pruebas de favorecer el reuso, la evolución de versiones y la calidad
- sólo es posible automatizar algunas tareas y actividades

Referencias

- [Amandi94] **Amandi A.; Prieto M.; Lopez J.; Leonardi G.; Rossí G., 1994, “Learning Object-Oriented concepts using Multimedia Technology”** , ACM, OOPS Messenger (March 94).
- [Booch94] Booch, G. ,1994, “*Object-Oriented Analysis and Design with Application*”, Benjamin/Cummings, 2nd ed.

- [IEEE 93] IEEE Recommended Practice for Software Requirements Specifications, 830-1993 Standard
- [Jacobson92] Jacobson, I.; Christerson, M.; Jonsson, P.; Overgaard, G., 1992, "Object-Oriented Software Engineering: a use case driven approach", Addison-Wesley
- [Jacobson94] Jacobson, I., 1994, "Scenario-based Design", J. Carroll Ed. ACM Press, Ch 12 : pp. 309-336.
- [Nanard95] Nanard, J.; Nanard, M., 1995, "Nypertext Design Enviroment and the Hypertext Design Process", Comm. ACM 38,8 (Aug 95) pp. 49-56
- [Olsina96a] Olsina, L. ; Salto, C. ; Nicolau, S. ; Irastorza, J. ; Bertone, E., 1996, "Estrategias y Criterios de Diseño e Implementación Hípermediales en el proyecto facultad de Ingeniería", Anales de INFOCOM 96, p.p. 510-519 Bs As.
- [Olsina96b] Olsina, L., 1996, "Visión de Modelo de Proceso para el desarrolla de Hipermedios", Proceed. del IV Encuentro Chileno de Ciencias de la Computación. Valdivia. Chile.
- [Olsina 97] Olsina, L., "Object-Oriented Prototyping Strategy fo support Hypermedia Flexible Process Mode/". Submitted paper, 1997.
- [Rossi95] Rossi, G. ; Schwabe, D.; Lucena C.J.P. ; Cowan, D.D. , 1995, "An Object-Oríented design Model for Designing the Human-Computer Interface of Hypermedia Aplicación"; Proc. of the International Workshop on Hypermedia Design (IWH95), Springer Verlag .
- [Rumbaugh91] Rumbaugh, J; Blaha, M; Premerlani, W; Eddy, F; Lorensen, W. ,1991, "Object-Oriented Modeling and Design", Prentice Hall.
- [Sadr96] Sadr, B. ; Dousette, P., 1996, 'An OO Project Management Strategy", IEEE Computer 29, 9 (Set 96) pp.33-38
- [Schwabe96] Schwabe, D.; Rossi, G. Barbosa, S , 1996, "Systemafíc Hypermedia Applicaton Design with OOHDM", Hypertext 96, US.
- [Thüring95] Thüring, M.; Hannemann, J.; Haake, J. , 1995, "Hypermedia and Cognition: Designing for Comprehension", Comm. ACM 38,8 (Aug 95) pp. 57-66

PROCESAMIENTO DE IMAGENES / IMAGE PROCESSING

TITULO:	TECNICAS ACELERADAS DE VISUALIZACION DE VOLUMENES
AUTOR:	GIULIETTI, ALEJANDRO ; DELRIEUX, CLAUDIO.
INSTITUCION:	DEPARTAMENTO DE INGENIERIA ELECTRICA. INSTITUTO DE CIENCIAS E INGENIERIA DE COMPUTACION. UNIVERSIDAD NACIONAL DEL SUR
DIRECCION:	C.C. 127 - (8000) - BAHIA BLANCA - BUENOS AIRES

TITULO:	EXTRACCION DE PARAMETROS EN IMAGENES FORENSES MEDIANTE MORFOMETRIA ANALITICA
AUTOR:	GONZALEZ, MANUEL; OLMO, DARIO; MOLER, EMILCE; PESSANA, FRANCO ; BALLARIN, VIRGINIA.
INSTITUCION:	LABORATORIO DE PROCESOS Y MEDICIONES DE SEÑALES. FACULTAD DE INGENIERIA. UNIVERSIDAD NACIONAL DE MAR DEL PLATA.
DIRECCION:	J. B. JUSTO 4302 - (7600) - MAR DEL PLATA - BUENOS AIRES

Técnicas Aceleradas de Visualización de Volúmenes

Alejandro Giuliatti, Claudio Delrieux
Departamento de Ingeniería Eléctrica
Instituto de Ciencias e Ingeniería de Computación
Universidad Nacional del Sur
Alem 1253 - Bahía Blanca - ARGENTINA
claudio@acm.org

Resumen

La visualización científica constituye actualmente una verdadera revolución en la metodología de investigación científica, tanto básica como aplicada. Dentro de la investigación en visualización científica, la representación de datos volumétricos se destaca por las dificultades computacionales que plantea, pero al mismo tiempo concentra la mayor atención en la investigación actual. Esto es así porque el rendering de datos volumétricos constituye uno de los procesos computacionalmente más caros.

En este trabajo se desarrollan técnicas y presentan detalles de implementación de un sistema de visualización de volúmenes basado en una modificación de los métodos tradicionales. El sistema está basado en el procesamiento *cell by cell*, y a través de simplificaciones en los distintos procesos intermedios, se logran mucho menores tiempos de procesamiento.

1 Introducción

La visualización científica constituye actualmente una verdadera revolución en la metodología de investigación científica, tanto básica como aplicada, comparable al invento del microscopio o el telescopio. Se utilizan los sistemas computacionales no para simular sino para *representar* enormes conjuntos de datos, apelando a la vasta capacidad de interpretación visual del cerebro. En este sentido, la visualización científica representa la culminación de las actuales posibilidades de los sistemas de computación gráfica. Estos datos pueden provenir de sensores, como en el caso de tomógrafos o de satélites, o bien pueden provenir de tareas computacionales anteriores, como por ejemplo de simulaciones o de análisis por elemento finito. El resultado gráfico que se espera de la visualización de estos datos no es meramente cuantitativo -no se busca necesariamente la representación fiel de valores- sino *cualitativo* -se busca un entendimiento global de determinadas propiedades de los datos [6, 21].

La visualización involucra el empleo de técnicas derivadas de la computación gráfica utilizadas para la representación de datos científicos de diverso tipo [1]. Dentro de la investigación en visualización científica, la representación de datos volumétricos se destaca por las dificultades computacionales que plantea, pero al mismo tiempo concentra la mayor atención en la investigación actual, dado que constituye una de las dificultades más importantes en los sistemas de visualización científica, por la complejidad del cálculo y el manejo de memoria asociado. La implementación de un sistema interactivo de visualización de datos volumétricos representa un gran desafío dadas las posibilidades limitadas de hardware y de programación, ya que, además de la complejidad de los modelos a procesar, se incorporan las dificultades inherentes al trabajar con un equipo limitado. Pero por otra parte, la representación volumétrica de datos es actualmente de gran utilidad en la investigación científica en temas tan diversos como en matemática, medicina, ciencias naturales e ingeniería [9, 12], y es utilizada para representar datos que pueden provenir de diversas fuentes. Al mismo tiempo, los resultados de la visualización volumétrica de estos datos no son meramente una representación cuantitativa de los mismos, es decir, no se busca necesariamente la presentación fiel de valores. Por el contrario, se busca un entendimiento global de determinadas propiedades del modelo o de la simulación que produjo los datos. Estos objetivos son sumamente exigentes en términos de tecnología, tanto de hardware como de software [5, 16].

En este trabajo se desarrollan técnicas y presentan detalles de implementación de un sistema de visualización de volúmenes basado en una modificación de los métodos tradicionales. El sistema está basado en el procesamiento por celdas (*cell by cell*), y a través de simplificaciones en los distintos procesos intermedios, se logran mucho menores tiempos de procesamiento. El trabajo está organizado de la siguiente manera: en la sección siguiente se presenta una revisión de las diferentes técnicas de representación de volúmenes existentes. Entre ellas, se destaca el procesamiento por celdas como candidato a diversas mejoras. En la sección 3, entonces, se detallan las distintas etapas necesarias en el procesamiento por celdas, y se discuten algunas ideas tendientes a acelerar el proceso de *rendering*. Luego, en la siguiente sección, se presentan los detalles de implementación de un sistema acelerado de representación de datos volumétricos basado en las ideas precedentes. En la sección 5 se muestran los resultados obtenidos, observándose tiempos

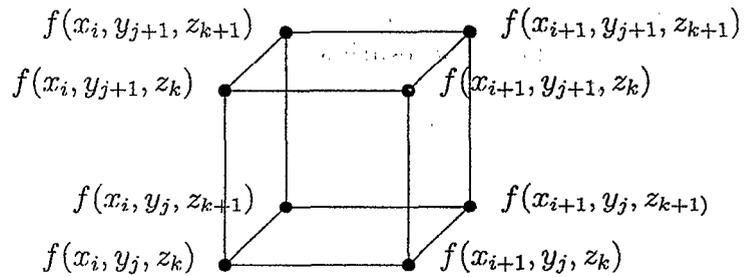
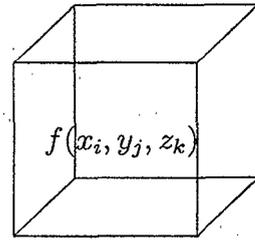
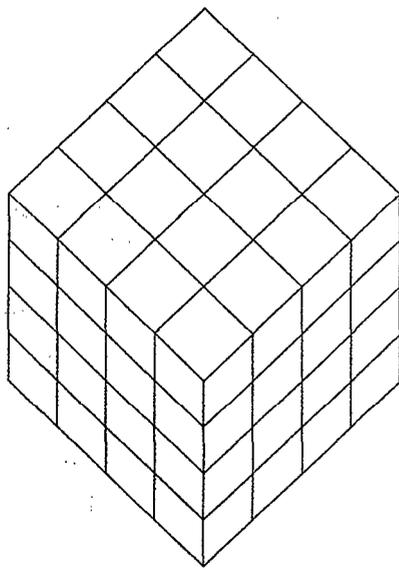


Figura 1: Un volumen de datos, y Figura 2: voxels vs. celdas

de cómputo mucho menores. Por último, en la sección 6 se discuten las conclusiones y las posibilidades de trabajo futuro.

2 Técnicas de representación de volúmenes

Los volúmenes de datos pueden pensarse abstractamente como matrices tridimensionales, en los que cada celda contiene valores uni o multivaluados (ver figura 1). Estos datos, como vimos, pueden provenir de diversas fuentes (tomógrafos, satélites, simulaciones computacionales). Estos datos fueron eventualmente preprocesados para extraer y enfatizar adecuadamente las características intuitivamente adecuadas en una determinada aplicación y para un determinado propósito en su visualización (por ejemplo, destacar determinadas áreas en la representación visual de una tomografía) [23, 26]. Una vez que los datos volumétricos están adecuadamente preparados para el rendering, el mismo procede según algoritmos de mayor o menor sofisticación. Las técnicas usuales de rendering de volúmenes están normalmente asociadas a una representación de los mismos en estructuras de celdas volumétricas o en voxels [22, 27] (ver figura 2). De esa manera, un voxel en particular representa el factor de ocupación que el sólido posee en una determinada fracción del espacio tridimensional. Como tal, es la menor cantidad de información que de la que puede disponerse en el espacio, como el pixel en el plano. En cambio, la celda está formada por "los 8 datos de sus vértices, con lo que se puede realizar una interpolación trilineal en cualquier lugar de su interior.

2.1 Técnicas primarias

Una de las primeras técnicas de rendering de volúmenes [8] consiste en graficar por capas el volumen de datos. Normalmente el volumen de datos se hace coincidir con los ejes del sistema de coordenadas del mundo [17], de modo que el eje z (hacia donde mira el observador) coincida con uno de los ejes del volumen de datos. Planos perpendiculares a dicho eje son entonces procesados de adelante hacia atrás. El procesamiento es sencillo, consistiendo en una : proyección paralela de los datos al buffer de pantalla, utilizando alguna técnica de *pseudocoloring* [19, 20] para asociar los valores a representar con colores de una paleta predeterminada (por ejemplo, asociar un determinado color a un determinado tejido). Cada voxel, en función de su valor, tiene a su vez una determinada transparencia, es decir que no es necesariamente opaco, permitiendo que se visualice parcialmente las partes del volumen que se encuentran detrás. La transparencia en cada dirección visual se computa acumulandola en un a-buffer de pantalla [3]. Para emular una proyección tridimensional, los datos de las capas se van desplazando una determinada distancia en x e y a medida que éstas son mas distantes en el eje z. Esta técnica es bastante primaria, pero por esa misma razón es implementable directamente con hardware específico. Su mayor limitación consiste en que, al no existir un sólido propiamente dicho en ningún momento del procesamiento, no es posible una representación con realismo [7, 24], por ejemplo, la interacción con iluminantes o con otros objetos.

Otros métodos más sofisticados buscan extraer la representación de un objeto tridimensional a partir del volumen de datos. Una de las primeras técnicas [10] consiste en procesar capa por capa al volumen de datos, en función de un determinado valor umbral. De esa manera, es posible identificar en una capa dada aquellos voxels en los cuales ocurre una transición cercana al valor umbral. Dichos voxels conforman un contorno. Entre dos capas adyacentes, entonces, es posible vincular los contornos para determinar un esqueleto de polígonos. El conjunto de polígonos encontrado entre todas las capas procesadas de esta manera constituye una representación del sólido con una estructura “intermedia”, en este caso, una superficie. Esta estructura de polígonos permite la visualización de los datos originarios, y tiene la ventaja de ser una estructura “tradicional” en el sentido de la computación gráfica, es decir, es un conjunto de polígonos, el cual puede graficarse con los algoritmos usuales, utilizando cara oculta, sombreado, iluminación, etc. Sin embargo, esta técnica encuentra problemas cuando no es directo encontrar el esqueleto de polígonos entre dos capas sucesivas (por ejemplo si ocurren discontinuidades topológicas).

Otra solución, más estable con respecto a este tipo de problemas, es la denominada “marching cubes” [15], en la cual se clasifican los voxels que pertenecen a una superficie umbral. Un voxel pertenece a la superficie umbral si por lo menos uno de sus vértices está por debajo del valor umbral y por lo menos otro está por encima. En este caso, cada uno de los ocho vértices de un voxel puede asumir un valor por debajo o por encima del umbral. El total de todos los casos posibles es $2^8 = 256$, pero por consideraciones de simetría se reducen a solo 14. Para cada uno de dichos casos es posible aproximar la superficie umbral con polígonos sencillos (normalmente triángulos) que cortan al voxel, y al mismo tiempo ubicar los voxels vecinos en los cuales dicha superficie debe continuar.

2.2 Ray casting

Una solución más completa (y también más compleja) para el rendering de volúmenes consiste en arrojar rayos desde el observador hacia el sólido, de una manera similar al ray *tracing* [11, 25], pero computando el comportamiento de la luz a través del volumen. Esta técnica, denominada ray casting, comienza por considerar que cada voxel es el modelo de un objeto físico, en el cual ocurre un fenómeno de interacción con la luz y con los rayos visuales provenientes de los demás voxels. Por lo tanto, es necesario establecer un modelo de iluminación que, a diferencia de los modelos tradicionales en computación gráfica como el de Phong [18], considere la interacción de la luz con una densidad volumétrica. Estos modelos fueron estudiados por Blinn [2] y por Kajiya [13, 14], llegando ambos a una formulación matemática similar.

Dada una densidad volumétrica $D(x, y, z)$ y un rayo visual v que la atraviesa entre dos puntos t_1 ***t2***, consideraremos por un lado la iluminación acumulada en su interacción con una distribución de energía luminosa I que representa una determinada condición de iluminación, y por otro lado la densidad acumulada por el rayo desde que ingresa al sólido en t_1 . Sea entonces un punto t entre t_1 y t_2 . La iluminación que recibe dicho punto es la sumatoria de las intensidades de las fuentes luminosas puntuales. El modelo considera que la densidad volumétrica puede pensarse como una distribución gaussiana de partículas idealmente especulares. Los algoritmos basados en esta técnica, normalmente simplifican esta ecuación según ciertas consideraciones. Por ejemplo, si los rayos visuales son paraxiales, es decir, con pequeña desviación angular respecto del eje z , entonces la distancia de t_1 a t_2 es constante en todos los voxels, y por lo tanto la integral puede aproximarse con una productoria [4].

2.3 Procesamiento por celdas

Esta técnica consiste en ir recorriendo la base de datos en una forma ordenada de adelante hacia atrás (según la posición del observador) y se va proyectando dato por dato, esto requiere de un buffer con la información de los datos ya proyectados (color y transparencia acumulados). En arquitecturas computacionales complejas, como por ejemplo en las máquinas Silicon Graphics, el buffer de pantalla está pensado como para brindar soporte a este tipo de operaciones, es decir, se opta por la solución más natural del **hardware** dedicado. En máquinas PC, sin embargo, el **buffer** de pantalla no tiene capacidad para soportar estos cálculos intermedios, por lo que hay que recurrir a la memoria de propósito general (RAM), con la consiguiente complicación en la programación, y los tiempos de cómputo mayores.

El procesamiento por celdas (o cell by cell) puede pensarse en forma diferente si se considera que los datos están formando voxels o formando celdas. El voxel es la mínima cantidad de información en 3D y vendría a ser lo que es el pixel en 2D. En cambio, la celda está formada por 8 datos que serían los vértices, entonces se puede realizar un promedio o bien una interpolación (trilineal) para obtener la información dentro de la celda y de esta forma evitar que en la imagen se perciban los pequeños cubos que la componen, cuando la matriz de datos es pequeña. La resolución de los gráficos está dada por el tamaño de la matriz de datos.

En este trabajo, describiremos la implementación de un sistema de rendering de datos volumétricos basado en la técnica de procesamiento cell by cell, por lo que describiremos a continuación en mayor detalle etapas más importantes del proceso.

- Preprocesamiento de datos, etapa que comprende la generación y acondicionamiento de la matriz de datos ó buffer volumétrico (**V-Buffer**).
- . Adecuación a las transformaciones de proyección. De acuerdo con el punto de observación se elige el orden en que se van a ir tomando los datos para su proyección y se toma el dato correspondiente.
- **Voxelización**, etapa que consiste en armar un voxel con una cierta ubicación en el espacio, de acuerdo a la posición que tiene el dato en la matriz.
- . Mapeo de color y transparencia, es decir, cada voxel va a tener un color y una transparencia asignada, de acuerdo al valor numérico del dato de la matriz que ha sido tomado.
- . Proyección de las caras de cada voxel que son visibles sobre la pantalla computando la contribución de color y transparencia que aporta el voxel sobre la imagen.

3 Implementación acelerada del procesamiento por c e l d a s

Una de las principales motivaciones para esta aplicación proviene de la posibilidad de producir visualizaciones interactivas, es decir, que el usuario pueda interactuar con el sistema de visualización y que éste responda en tiempos relativamente pequeños. Por dicha razón fueron necesarias las técnicas de rendering acelerado que se presentan a continuación. Recapitulando, en el procesamiento por celdas, un camino óptico que parte del observador y atraviesa el sólido, entonces, acumula una energía luminosa y una transparencia computada en función de los voxels con que se encuentra en su recorrido (ver figura 3a). Teniendo en cuenta que este cómputo se puede realizar luego de haber la matriz de datos según la transformación perspectiva, el procesamiento cell by cell se puede efectuar realizando la actualización del **buffer** siguiendo una proyección perpendicular al plano de proyección (ver figura 3b).

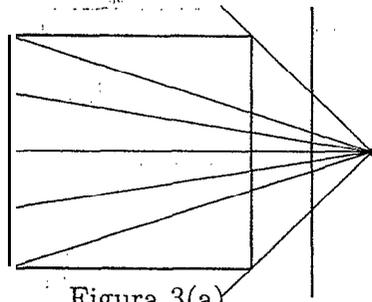


Figura 3(a)

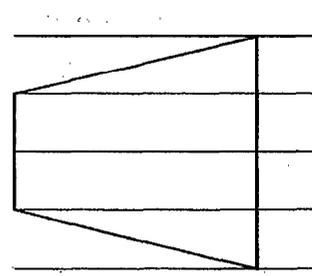


Figura 3(b)

A esta altura surgen algunas variantes a la secuencia clásica como, por ejemplo para ir proyectando los voxeles en un orden adecuado, las alternativas son:

1. Transformar toda la matriz de acuerdo al punto de observación, y luego recorrerla en un orden creciente del eje z (de adelante hacia atrás).
2. Precomputar la orientación en que va a quedar la matriz de datos de acuerdo al punto de observación, y generar la secuencia de barrido de la matriz sin transformar.

La primera alternativa es más rápida, pero tiene la desventaja de necesitar otra matriz de datos de igual tamaño conteniendo a la matriz transformada, por lo que en esta implementación se optó por la segunda alternativa. Otro punto que permitió una variante es el orden en que se debe realizar la secuencia voxelizar, transformar y proyectar. Si se realizara primero la transformación, luego la voxelización y por último la proyección, durante el proceso de voxelización se puede elegir la orientación de los voxels para que solo se vea una de las seis caras (la que da hacia la pantalla), con el consiguiente ahorro de cálculo a la hora de proyectar. Además de tener que transformar un solo punto por voxel y no ocho. En este trabajo se muestran las dos variantes para observar las diferencias, comenzando con el método convencional y más adelante se comenta la variante más rápida.

En definitiva, la secuencia de pasos a realizar en la implementación del procesamiento por celdas convencional es la siguiente (ver figura 4), mientras que en la figura 5 se muestra la modificación dentro de la iteración para producir el **rendering** acelerado.

- **Chequeo del sistema:** Verifica los requerimientos mínimos para la ejecución del programa.
- **Carga de la matriz de datos:** Verifica la existencia del archivo de datos y los carga en la memoria extendida (XMS) (ver más adelante las consideraciones relativas al manejo de memoria).
- **Ingreso de parámetros:** Permite la modificación del punto de observación y mapeo de color.
- **Cálculo del orden de proyección:** De acuerdo a las rotaciones ingresadas, analiza qué cara se ve primero y selecciona el orden en que se recorre la matriz.
- **Tomar el dato correspondiente:** Según la secuencia preestablecida de orden de recorrida, se accede a la memoria XMS y se trae el dato a proyectar.
- **Voxelización,** Convierte el dato en un cubo, siendo su ubicación en el espacio función de las coordenadas del punto dentro de la matriz.
- **Mapeo de color y transparencia:** Asigna un color y una transparencia al voxel, según sea el valor del dato.
- **Transformaciones:** Se le aplican a los 8 vértices del voxel las transformaciones ingresadas.
- **Rendering:** Esta rutina debe comprender los siguientes pasos:
 1. Determinar qué caras del voxel son visibles.
 2. Determinar el área que ocupa sobre el *buffer* de pantalla la proyección del voxel.
 3. Realizar la proyección de las caras visibles (algoritmo de conversión-scan [17], analizando la contribución de color del voxel a la imagen.
 4. Actualizar el *buffer* de pantalla.

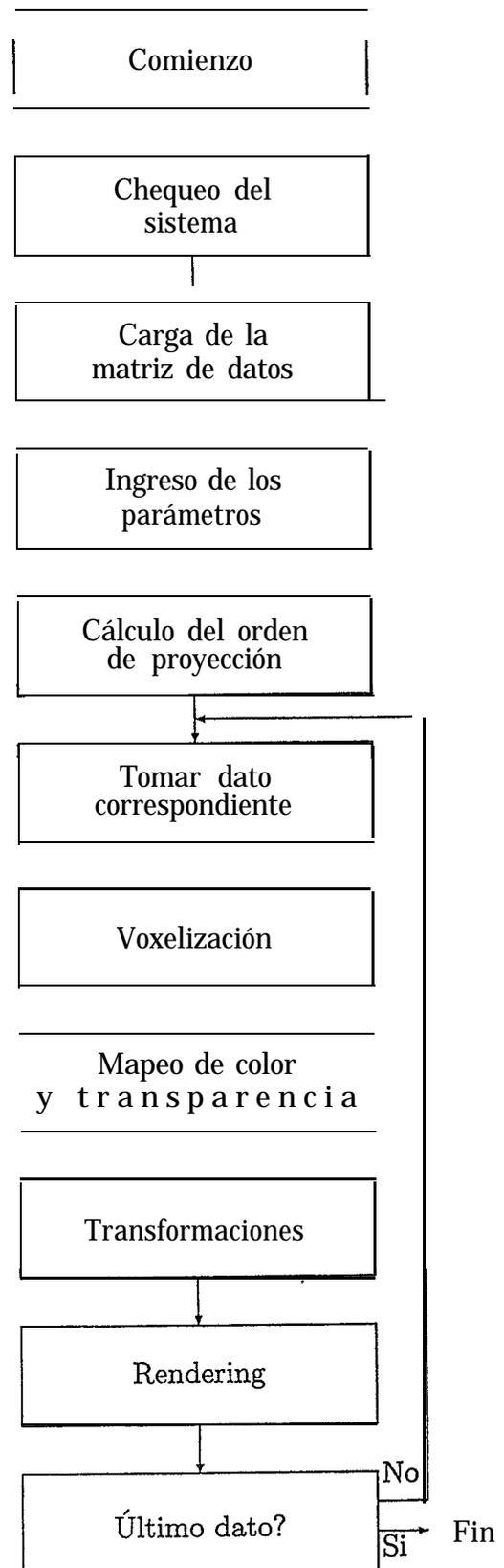


Figura 4: Secuencias principales del sistema.

4 Detalles de implementación

En esta sección describiremos algunos detalles prácticos e importantes de la implementación realizada. Los mismos reflejan las consideraciones aconsejadas por la experiencia, y por lo tanto son también una parte valiosa para tener en cuenta. El programa fue realizado en una 486 DX2 con 8 Mbytes de RAM y una placa de video de 1 Mbyte. El compilado y linkeado fue realizado con los siguientes parámetros: set de instrucciones i486, alineación Word, modelo de memoria large y con manejo del coprocesador matemático.

Se prefirió trabajar con el Borland C++, debido a cierta experiencia en su manejo y con la versión 4.0 para plataforma Windows, por poseer un ambiente de programación más cómodo. De todas maneras, el ejecutable de la versión realizada trabaja bajo DOS, por lo que se hubiese podido utilizar una versión anterior. Dadas las limitaciones en la memoria principal, se decidió trabajar con una base de datos de $128 \times 128 \times 128$, con un byte por dato alcanzando los 2 Mbytes = 2.097.152 datos. También se decidió utilizar como modo de video gráfico el IBM8514 con la resolución de 1024×768 y 256 colores, debiéndose poseer una placa compatible con este modo y el emulador activado para poder correr el programa. Este modo gráfico es el mejor disponible para una versión ejecutable bajo DOS,

Con respecto al mapeo de color y transparencia, como ya se ha mencionado, la adecuada asignación de color y porcentaje de transparencia a los distintos valores de la función a visualizar es necesaria para obtener buenos resultados. El primer problema a resolver consiste en decidir la cantidad de colores que se van a utilizar en la paleta de la tarjeta gráfica, dado que solo nos permite mostrar 256 colores simultáneos. Luego de realizar diversas experiencias de trabajo, se logró comprender que en este trabajo no es necesario representar simultáneamente muchas cromaticidades diferentes, sino que es necesario tener la mayor cantidad de combinaciones simultáneas posibles de unos pocos colores. Es por eso que se decidió trabajar con dos colores simultáneamente, con 16 intensidades de cada uno, 16 intensidades de rojo, 16 de azul, y los restantes 210 colores que forman la paleta son las combinaciones entre los anteriores. Dicha selección permite obtener interesantes resultados cuando los datos representan algún tipo de objeto real, pero si se desea representar datos de mediciones es necesario tener más colores para diferenciar bien, por ejemplo distintas densidades.

Otro punto clave es la asignación de transparencia u opacidad, con la cual se debe eliminar aquel tipo de información que no es necesario ver, es decir si no se desea ver algo se le mapea opacidad 0%. También se debe tener cuidado con evitar que la opacidad acumulada llegue al 100% antes de tiempo imposibilitando la observación de detalles esenciales. Esto último se debe a que si la proyección de algún voxel cae en un pixel que ya tiene una opacidad acumulada igual al 100% no se produce modificación de color ni de transparencia. La información del color de cada pixel y su opacidad se almacenan en un **frame buffer** en memoria XMS, y cada vez que se va a actualizar un pixel, se calcula el efecto de la combinación de los valores a poner con los ya acumulados y esta composición es la que se pone en pantalla y la que actualiza el **frame buffer**. En este trabajo se asignó un el tamaño de un long int (4 bytes) a la información de color y transparencia de cada pixel, totalizando 3 Mbytes para el uso del frame buffer. Esta precisión en la representación de la transparencia es necesaria para evitar los errores numéricos en las operaciones de

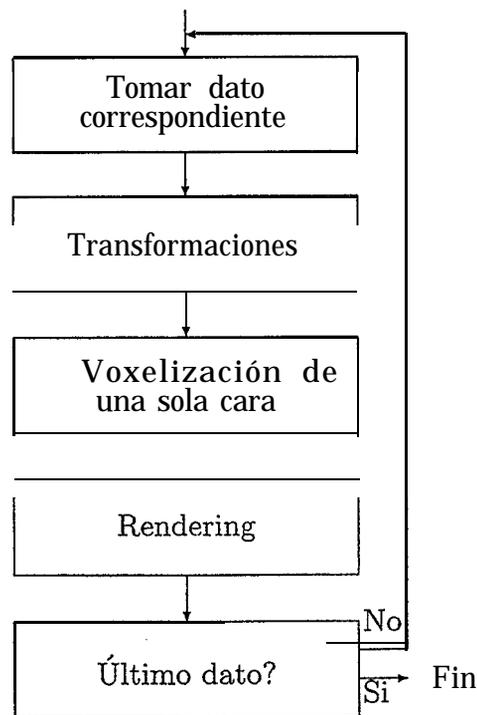


Figura 5: Secuencia del método acelerado.

combinación de colores lo que exige almacenar una gama amplia de transparencias y colores acumulados, aunque luego al aplicar el modelo de color (asignación del color más cercano existente en la paleta) gran parte de la diferencia entre dichos colores se pierda.

Por último, con respecto a la generación de los datos utilizados para ejemplificar los resultados obtenidos, debemos tener en cuenta que el rendering de volúmenes está directamente relacionado con la visualización de datos obtenidos por muestreo, por ejemplo a través de ultrasonido o resonancia magnética, datos que representan el valor de alguna variable en diferentes puntos del espacio (densidad, temperatura, etc.). A los datos también se les suele aplicar un filtrado para minimizar el aliasing y eliminar errores. Para este trabajo, los datos se generaron a través de una rutina, para ver algunos cuerpos simples con cierto grado de transparencia. La cantidad de datos se limitó a 2 Mbytes en una matriz de $128 \times 128 \times 128$ si a cada dato se le asigna un byte. Para generar una base de datos que funcione con el programa, ésta debe tener 2.097.152 Bytes, representar una matriz de $128 \times 128 \times 128$ datos y cada dato debe ser un 0, 1, 2 o 3, en formato char. 0 indica al programa que no hay nada en ese punto del espacio, 1 y 2 que en ese lugar se encuentra el cuerpo 1 o el cuerpo 2 respectivamente. Finalmente 3 indica que se encuentran superpuestos ambos cuerpos. Para realizar cortes de la base de datos de forma tal que se puedan observar el interior, existen dos momentos, uno es durante la proyección de la base de datos, pero no se ha implementado en este programa debido a que produce un tiempo de cómputo aún mayor, y el otro momento es durante el acondicionamiento de los datos. En este caso se puede hacer mediante una rutina que acceda al archivo de datos y que ponga ceros donde se desea, ahora ya no de acuerdo al valor de la función (densidad o temperatura), sino de acuerdo a la posición que tiene el dato en el espacio.

5 Resultados obtenidos

En esta sección compararemos los tiempos y las calidades de gráficos ‘ obtenidos con la implementación del método cell *by* cell descrita en secciones anteriores, y al producir algunas simplificaciones. Como ya mencionáramos, dichas simplificaciones se basan en una alteración de la secuencia en que se realizan los pasos de la secuencia principal. Esencialmente, en vez de generar los ocho vértices del voxel y obtener la ubicación del mismo luego de las rotaciones, lo que se realiza es primero obtener la ubicación del mismo, aplicándole las rotaciones ingresadas,, y luego generar el voxel. Como el voxel se puede “armar” en la orientación que uno desee, lo que se hace es generarlo de forma que solo una de sus caras sea visible. En realidad solo se genera esa cara.

El ahorro del tiempo de cálculo aparece también en el momento de proyectar, ya que no se necesitan eliminar las caras no visibles, y siempre se proyecta una sola cara. Además no se necesita el algoritmo de conversión-scan de la cara para saber cuáles son los lugares del buffer de pantalla que se deben actualizar, dado que el área de una cara proyectada perpendicularmente se puede computar con dos ciclos for. Este método simplificado es considerablemente más veloz, pero provoca un mayor aliasing en la imagen resultante, debido al solapamiento o superposición espacial de las caras generadas (ver figura 6). Este defecto se podría mejorar, sin embargo, con un postprocesamiento de la imagen, utilizando alguna técnica de filtrado o dithering.

Al realizar varias figuras, se puede observar que existen algunos ángulos o rotaciones que producen mayor aliasing que otras, tanto en el método normal como en el simplificado. Se intentó otra variante, que consistía en generar el voxel y transformarlo, como en el método normal, pero luego proyectar solo la cara directamente visible de cada voxel. Pero de esta forma tampoco se pudieron eliminar los defectos. Además, el cómputo era más lento que el método simplificado recién descrito, y no lo superaba en calidad.

El tiempo aproximado que tardan ambas técnicas (la normal y la simplificada), para recorrer una base de datos completamente vacía es de 45 segundos. Para algunos gráficos los tiempos de cómputo son los siguientes:

	Método normal	Método acelerado
Fig. 7	3' 08"	1' 47"
Fig. 8	9' 16"	4' 20"
Fig. 9	21' 43"	9' 05"

Estos tiempos son muy dependientes del momento en que se alcanza la opacidad al 100%, es decir de los porcentajes de opacidad ingresados. Además para mejorar el tiempo de cómputo se sacaron las protecciones de errores como el de transferencia de datos de memoria convencional a XMS y viceversa, entre otros.

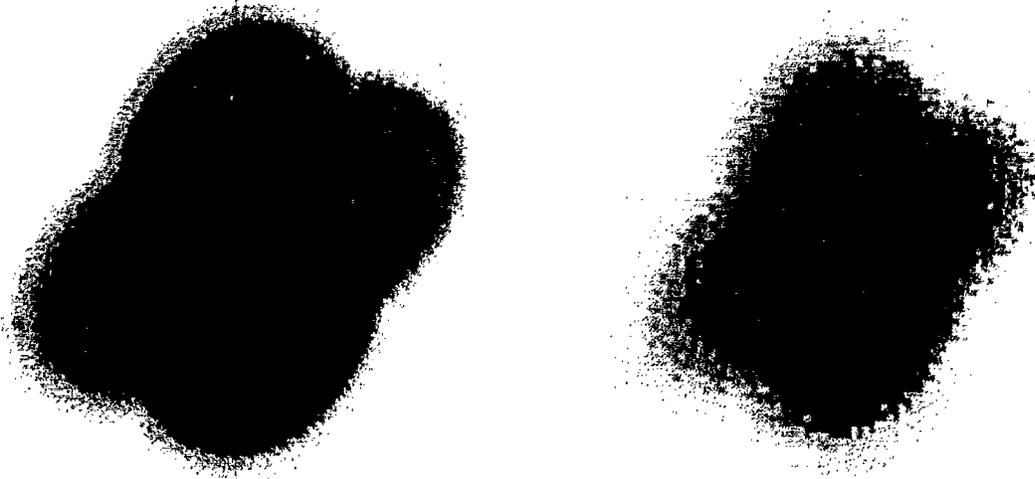
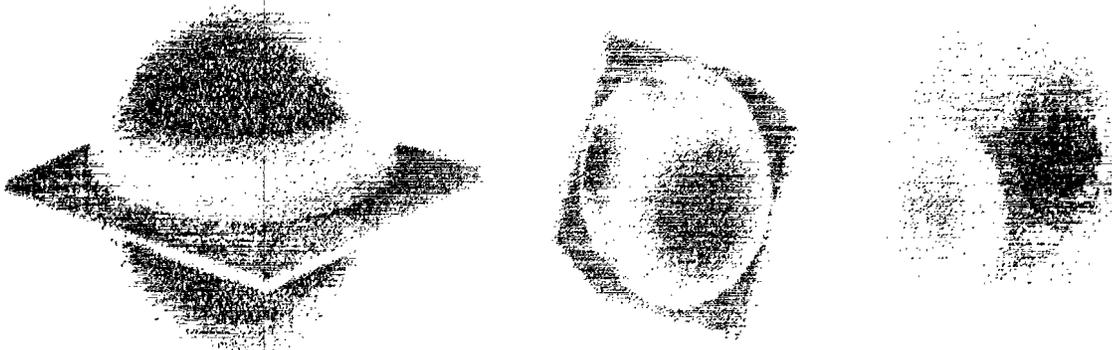


Figura 6: *Aliasing* en el método simplificado.



Figuras 7, 8 y 9: Comparación de resultados.

6 Conclusiones

Se presentó una discusión de la relevancia de la representación de datos volumétricos en los sistemas de visualización científica, así como una revisión de las diferentes técnicas de representación de volúmenes existentes. Entre ellas, se destacaron las posibilidades del procesamiento por celdas como método candidato a diversas mejoras. A partir del detalle de las distintas etapas necesarias en el procesamiento por celdas, se discutieron algunas ideas tendientes a acelerar el proceso de rendering. Luego se desarrollaron las técnicas y presentaron los detalles de implementación de un sistema de visualización de volúmenes basado en una modificación del procesamiento por celdas tradicional. A través de las simplificaciones mencionadas en los distintos procesos intermedios, se lograron mucho menores tiempos de procesamiento.

Entre las posibilidades de trabajo futuro, podemos mencionar la inclusión del filtrado y postprocesamiento dentro de la secuencia del sistema. Ésto se puede conseguir de por lo menos dos formas. Una de ellas consiste en tener una ventana de tamaño impar (por ejemplo 3x3), con coeficientes ponderadores de suma unitaria. De esa manera, al actualizar *el buffer*, no se lo hace únicamente en el pixel correspondiente sino en un grupo de 3x3 centrado en el mismo, y ponderado por los coeficientes respectivos. Esto produce un filtrado de antialiasing

Otra forma de producir un filtrado, más rápida, consiste en interpretar los coeficientes de la ventana como probabilidades. De esa manera, al actualizar un pixel, la probabilidad de que dicho pixel o cualquiera de sus vecinos inmediatos sea actualizada está reflejada en la ventana. Ésto produce un efecto llamado *dithering*, el cual también es una forma de filtrado antiuliasing.

Referencias

- [1] R. S. Avila, I. M. Sobierajski, y A. E. Kaufman. Towards a Comprehensive Volume Visualization System: En *Visualization '92 Proceedings*, págs. 13-20, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [2] James F. Blinn. Light Reflection Function for Simulation of Clouds and Dusty Surfaces. *ACM Computer Graphics*, 16(3):21-29, 1982.
- [3] James F. Blinn. Compositing 1 - Theory. *IEEE Computer Graphics and Applications*, 14(5):83-87, 1994.
- [4] Silvia Castro, Miguel Danzi, Claudio Delrieux, Marcelo Larrea, y Andrea Silvetti. Visualización de Volúmenes: Aplicaciones en Ciencias Geológicas. En *Actas del 2º Congreso Argentino de Ciencias de la Computación*, págs. 44-56, San Luis, Argentina, 1996. CACiC.
- [5] J. Chalinger. Scalable Parallel Volume Ray Casting for Nonrectilinear Computational Grids. En *1993 Parallel Rendering Symposium Proceedings*, págs. 81-88, Los Alamitos, CA, 1993. ACM SIGGRAPH, ACM Press.

- [6] T. A. Defanti, M. D. Brown, y B. H. McCormick. Visualization: Expanding Scientific and Engineering research Opportunities. En G. M. Nielson y B. D. Shriver, editores, **Visualization in Scientific Computing**, págs. 32-47. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [7] R. Drebin, L. Carpenter; y P. Hanrahan. Volume Rendering. **ACM Computer Graphics**, 22(4):65-74, 1988.
- [8] J. Farrell. Colour Display and Interactive Interpretation of Three-Dimensional Data. **IBM Journal of research, and Development**, 27(4):356-366, 1983.
- [9] H. Fuchs, M. Levoy, y J. K. Lam. Interactive Visualization of 3D Medical Data. En G. M. Nielson y B. D. Shriver, editores, **Visualization in Scientific Computing**, págs. 140-146. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [10] S. Ganapathy y T. Dennehy. A New General Triangulation Method for Planar Contours. **ACM Computer Graphics**, 16(3):69-75, 1983.
- [11] A. Glassner. **An Introduction to Ray Tracing** Academic Press, Cambridge, Massachusetts, 1991.
- [12] W. Hibbard y D. Santek. Visualizing Large Meteorological Data. En G. M. Nielson y B. D. Shriver, editores, **Visualization in Scientific Computing**, págs. 147-152. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [13] James T. Kajiya: The Rendering Equation. **ACM Computer Graphics**, 20(4):143-150, 1986.
- [14] James T. Kajiya y B. Von Herzen. Ray Tracing Volume Densities. **ACM Computer Graphics**, 18(4):91-102, 1984.
- [15] W. Lorensen y H. Cline. A High Resolution 3D Surface Construction Algorithm. **ACM Computer Graphics**, 21(4):163-169, 1987.
- [16] B. Lucas. A Scientific Visualization Renderer . En **Visualization '92 Proceedings**, págs. 227-234, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [17] W. Newman y R. Sproull. **Principles of Interactive Computer Graphics**. McGraw-Hill, New York, 1973.
- [18] Bui-Tong Phong. Illumination for Computer-Generated Pictures. **Communications of the ACM**, 18(6):311-317, 1975.
- [19] P. Rheingans. Color, Change and Control for Quantitative Data Display. En **Visualization '92 Proceedings**, págs. 252-259, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [20] P. K. Robertson. Visualizing Color Gamuts: a User Interface for the Effective Use of Perceptual Color Spaces in Data Displays. **IEEE Computer Graphics and Applications**, 8(5):50-64, 1988.

- [21] L. Rosenblum. Scientific Visualization at Research Laboratories. *IEEE Computer*, **22(8):68-100, 1989.**
- [22] G. Sakas y J. Hartig. Interactive Visualization of Large Scalar Voxel Fields. En *Visualization '92 Proceedings*, págs. 29-38, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [23] R. Santaney, D. Silver, N. Sabusky, y J. Cao. Visualizing Features and Tracking Their Evolution. *IEEE Computer*, **27(7):20-27, 1994.**
- [24] Alan Watt y Mark Watt. *Advanced Animation and Rendering Techniques*. Addison-Wesley, London, 1992.
- [25] T. Whitted. An Improved Illumination Model for Shaded Displays. *Communications of the ACM*, **23(6):343-349, 1980.**
- [26] C. Williams, J. Rasure, y C. Hansen. State of the Art of Visual Languages for Visualization. En *Visualization '92 Proceedings*, págs. 202-209, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [27] J. R. Wright y J. C. Hsieh. A Voxel-based, Forward-Projection Algorithm for Rendering Surface and Volumetric Data. En *Visualization '92 Proceedings*, págs. 340-348, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.

EXTRACCIÓN DE PARÁMETROS EN IMÁGENES FORENSES MEDIANTE MORFOMETRÍA ANALÍTICA

**FRANCO PESSANA* - VIRGINIA BALLARIN* - EMILCE MOLER* -
MANUEL GONZALEZ* - DARÍO OLMO****

**Laboratorio de Procesos y Mediciones de Señales, Dpto. de Electrónica,
Facultad de Ingeniería, Universidad Nacional, de Mar del Plata,
J. B. Justo 4302, Mar del Plata, C.P. 7600, Argentina, Tel.: +54-23-816600 ext.
255,*

*Fax: +54-23-810046, E-mail: fpessana @ fi.mdp.edu.ar
vballari@ fi.mdp.edu.ar*

FRANCO PESSANA: Graduado Distinguido en 1995 de la Fac. de Ingeniería de la Univ. Nac. de Mar del Plata con el título de Ingeniero Electrónico. Becario del Laboratorio de Procesos y Mediciones de señales, Área Procesamiento Digital de Imágenes, Dpto. de Electrónica, UNMdP desde agosto de 1994 hasta julio de 1996. Actualmente, integrante en las cátedras de Matemática II e Introducción al Procesamiento Digital de Imágenes de la Fac. de Ingeniería, UNMdP y realizando una Beca de Iniciación otorgada por el Consejo Nacional de Investigaciones Científicas y Técnicas, CONICET.

VIRGINIA BALLARIN: Egresada, en 1984, de la Fac. de Ingeniería de la Univ. Nac. de Mar del Plata con el título de Ingeniería Electrónica. Jefe de Trabajos Prácticos, dedicación exclusiva, en las Cátedras de Probabilidad, Estadística y Procesos Estocásticos e Introducción al Procesamiento Digital de Imágenes' de la Facultad de Ingeniería de la Univ. Nac. de Mar del Plata. (+)

Responsable del área de Procesamiento en Imágenes Digitales.

Representante de la Univ. Nac. de Mar del Plata en la red alfa SIMUS. (Integrated Manufactured System)

EMILCE MOLER: Egresada, en 1983, de la Fac. de Cs. Exactas y Naturales de Univ. Nac. de Mar del Plata con el título de Prof. en Matemática. Prof. Adjunta, dedicación exclusiva, en las Cátedras de Computación e Introducción al Procesamiento Digital de Imágenes de la Facultad de Ingeniería de la Mar del Plata. Responsable del área Matemática en Procesamiento Digital de Imágenes. (+)

(+) Desde 1989 es miembro del Laboratorio de Procesamiento y Mediciones de Señales en el área de Procesamiento Digital de Imágenes, Codirectora de becarios y proyectos finales en temas afines. Participó en Congresos nacionales e internacionales, cuenta con publicaciones en la disciplina. Actualmente, realizando la tesis final del Master en Epistemología y Metodología de las Ciencias.

MANUEL GONZALEZ Egresado de la Fac. de Ingeniería de la Univ. Nac. de Mar del Plata con el título de Ingeniero Electrónico.. Prof. Titular de la cátedra Mediciones-Eléctricas II de la Facultad de Ingeniería de la Univ. Nac. de Mar del Plata."

Director del Laboratorio de Procesamiento y Medición de Señales, Director de Proyectos de Investigación y becarios. Actual Decano de la Facultad de Ingeniería la Univ. Nac. de Mar del Plata.

** **DARÍO OLMO:** Licenciado en Antropología. Miembro del Equipo Argentino-de Antropología Forense (EAAF).

ABSTRACT

El presente trabajo describe la aplicación de técnicas de Procesamiento Digital de Imágenes (PDI) a imágenes forenses, para la extracción de parámetros que caracterizan a formas craneales. Estos métodos encuentran su campo de aplicación en la disciplina de Antropología Forense.

La extracción de parámetros se realiza mediante **Morfometría Analítica** y se utilizan como valores discriminantes para posibles identificaciones de personas.

1. INTRODUCCION

El **Procesamiento Digital de Imágenes (PDI)** es una rama del Procesamiento Digital de Señales que permite manipular, modificar y analizar imágenes digitales, originariamente continuas, a través de una computadora.

El PDI cuenta hoy con ramas tan diversas como: Compresión, Codificación, Mejoramiento, Restauración e Interpretación de imágenes, entre otras. Se encuentran aplicadas a distintos campos científicos tales como: análisis químicos, materiales, medicina, geología de costas, visión por computadora, robótica, comunicaciones satelitales, etc. Es necesario destacar las aplicaciones en las ciencias sociales y humanísticas mencionando tareas de reconstrucción de documentos y trabajos en **Antropología Forense**.

El PDI tiene una aplicación concreta en este campo y es aplicar la técnica de superposición cráneo-foto y su posterior análisis de las imágenes superpuestas para que se lleven a cabo posibles identificaciones.

El problema es extraer curvas representativas y perfiles de las imágenes superpuestas con el objetivo de realizar un análisis comparativo. Este análisis se desarrolla mediante aproximaciones polinomiales y Transformada de Fourier. Utilizando la teoría de Morfometría Analítica se obtienen una serie de parámetros que permitirán de una manera simple y eficiente evaluar la correspondencia entre perfiles para una posterior identificación positiva de personas,

La Morfometría Analítica supera en gran medida a la descripción numérica que se basa en medidas discretas entre puntos de referencia Canónicos estándar en los estudios antropológicos. A continuación se describen un conjunto de parámetros: forma fundamental, coeficientes de Fourier, área allométrica e isométrica, que serán utilizados como clasificador o evaluador para determinar a que clase pertenece cada cráneo.

Los algoritmos correspondientes fueron realizados en lenguaje C++. Los sistemas estándar de PDI, no contemplan la posibilidad de la extracción de dichos parámetros ni el análisis comparativo de los perfiles.

2. MORFOMETRÍA ANALITICA

A principios de siglo, la descripción numérica de formas craneales con medidas discretas entre puntos de referencia canónicos fue fuertemente cuestionada por varios científicos dado que los métodos empleados eran inadecuados y sin un fundamento teórico apropiado. A partir de esta dificultad surge la teoría de la Morfometría Analítica que basa la extracción de nuevas

características a través de formulaciones matemáticas analíticas, que superan las limitaciones de mediciones discretas, y son efectivas al describir formas.

Es fundamental identificar en un primer paso las características morfológicas fundamentales del objeto (**forma fundamental**) que nos permiten clasificar en grandes grupos o tipos principales. En un segundo paso se deben identificar las características secundarias para realizar una clasificación más estricta. De esta manera la información de la forma bajo estudio queda constituida por estas dos componentes.

En este trabajo se analiza los resultados para las siguientes figuras rectángulo, elipse y triángulo, que aunque parecen figuras simples, su desarrollo en serie de Fourier produce una gran cantidad de información. También de esta manera se pueden caracterizar numéricamente y sin ambigüedad otras formas cualesquiera, obteniendo una descripción paramétrica de perfiles de cráneos.

Fig. 1a, 1b, 1c, 1d, 1e, 1f, 1g, 1h, 1i, 1j, 1k, 1l, 1m, 1n, 1o, 1p, 1q, 1r, 1s, 1t, 1u, 1v, 1w, 1x, 1y, 1z

2.1 DETERMINACIÓN DE LA FORMA FUNDAMENTAL

Para simplificar posteriores esquemas de clasificación, es necesario modelizar de manera más simple la forma, esto es lo que se conoce como **forma fundamental**. También se necesita tener una curva de referencia para computaciones adicionales. La curva se obtiene mediante **aproximación polinomial**.

Los procedimientos comunes de aproximación polinomial son útiles, para describir una curva abierta, sin embargo son inadecuados para describir curvas cerradas. Este software elimina el problema de curvas cerradas tratando los valores de abscisa y ordenada de los puntos separadamente, *Figs. 1b, 1c; 3b, 3c; 5b, 5c*, y considerando estos valores como variables dependientes de acuerdo con el método de regresión lineal múltiple que permite calcular los coeficientes (parámetros) de dos polinomios de grado superior del tipo:

donde D es el grado del polinomio de aproximación y x es la variable independiente, representada por una serie de enteros positivos (N elementos), que es el número de puntos en el cual se divide el perfil. La variable dependiente se representa por el valor de abscisa o por el valor de ordenada. El método se aplica dos veces para cada perfil.

En realidad, la máxima aproximación polinomial nunca se alcanza. El propósito no es describir la curva exactamente, sino aproximarla en una forma controlada. Tal control consiste en interrumpir el grado de crecimiento del polinomio cuando el valor del error estándar es mínimo.

De los cálculos de estos coeficientes, se obtienen dos series de N valores interpolados, que considerados como par ordenado, dan una nueva serie de valores (x,y) que representan la **forma fundamental**, *Figs. 1f 3f; 5f*

De esta manera, incluso cuando la curva original es complicada, siempre se obtendrán dos series de valores no recursivos que son no ambiguos al considerar las coordenadas de los puntos. Cuando las curvas son abiertas, un polinomio es suficiente para definir la forma fundamental, en el cual los valores de abscisa son la variable dependiente respecto a la serie de N

puntos (la variable independiente) en que el perfil es dividido.

De las comparaciones entre la curva original y su forma fundamental, se obtiene un **parámetro** sintético que es función de las irregularidades del perfil y que está constituido por la suma de diferencias entre las dos curvas. También pueden ser obtenidas las dimensiones típicas de las curvas: **perímetro, diámetro de proyección, y para curvas cerradas área y diámetro máximo.**

La forma fundamental, debido a su efecto de suavizado puede eliminar del perfil todo lo que se refiere a características particulares de un espécimen o alteraciones secundarias suaves ocurridas durante una fosilización.

Antes de la síntesis de la forma fundamental, se debe realizar una normalización dimensional y una estandarización. La normalización dimensional permite, eliminando diferencias de tamaño, la comparación directa de los resultados obtenidos. Si el análisis se refiere a curvas abiertas, la normalización se obtiene a través de la ampliación o reducción de escala de las curvas a comparar. Cuando las curvas son cerradas, se normalizan las áreas. En síntesis, normalizamos a fin de obtener curvas sintéticas con igual cantidad de puntos.

La estandarización surge de la necesidad de que los puntos que caracterizan una curva deben estar ordenados. En el **caso** de los perfiles que provienen de cortes transversales de los cráneos, perfiles fronto-faciales, existen puntos iniciales fijos (bregma; prosthion). En los cortes transversales, craniogramas horizontales, este punto es único (glabella). Para posibilitar posteriores clasificaciones, es indispensable contar con estos puntos iniciales a fin de estandarizar las formas fundamentales.

2.2 CALCULO DE LOS COEFICIENTES DE LA FORMA FUNDAMENTAL

Como se mencionó en la sección anterior, los coeficientes de la forma fundamental se' obtienen aplicando el método de regresión lineal múltiple. Expresando el sistema de ecuaciones en forma matricial se obtiene:

$$\begin{bmatrix} n & \sum_{\mu} x_{\mu} & \sum_{\mu} x_{\mu}^2 & \dots & \sum_{\mu} x_{\mu}^D \\ \sum_{\mu} x_{\mu} & \sum_{\mu} x_{\mu}^2 & \sum_{\mu} x_{\mu}^3 & \dots & \sum_{\mu} x_{\mu}^{D+1} \\ \sum_{\mu} x_{\mu}^2 & \sum_{\mu} x_{\mu}^3 & \sum_{\mu} x_{\mu}^4 & \dots & \sum_{\mu} x_{\mu}^{D+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{\mu} x_{\mu}^D & \sum_{\mu} x_{\mu}^{D+1} & \sum_{\mu} x_{\mu}^{D+2} & \dots & \sum_{\mu} x_{\mu}^{2D} \end{bmatrix} \begin{bmatrix} \theta_0^* \\ \theta_1^* \\ \theta_2^* \\ \vdots \\ \theta_D^* \end{bmatrix} = \begin{bmatrix} \sum_{\mu} y_{\mu} \\ \sum_{\mu} x_{\mu} y_{\mu} \\ \sum_{\mu} x_{\mu}^2 y_{\mu} \\ \vdots \\ \sum_{\mu} x_{\mu}^D y_{\mu} \end{bmatrix} \quad [1]$$

donde: x_{μ} es la variable independiente de las curvas rectificadas; y_{μ} son las coordenadas del perfil y e^* es el vector estimado de coeficientes.

En forma reducida se expresa la Eq. 1 como:

despejando θ^* se obtiene

[3]

La resolución del sistema expresado por la Eq. 3, se resuelve **en este** software mediante el método de Gauss in-place.

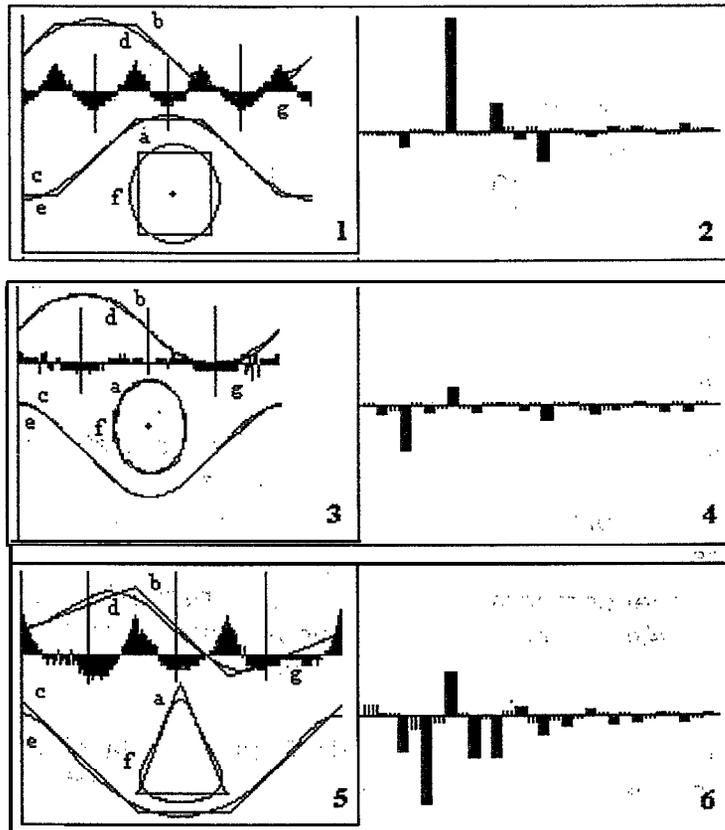
2.3 ANÁLISIS COMPARATIVO DE PERFILES MEDIANTE TRANSFORMADA DE FOURIER

Usualmente, el análisis armónico se basa en un desarrollo en series de Fourier de una señal periódica de período T. En este caso, se está trabajando con señales discretas, (la forma fundamental, como así también su descomposición en abscisas y ordenadas). En consecuencia ya no se puede hablar de una transformación continua de Fourier sino de una. transformada discreta de Fourier DFT de una secuencia x(n)

$$X(k) =$$

La secuencia transformada X(k) tiene una longitud, de N puntos. La IDFT de la secuencia X(k) es

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp\left(\frac{j2\pi nk}{N}\right) = \sum_{k=0}^{N-1} X(k) W_N^{-nk} \quad n = 0,1,2,\dots,N-1 \quad [2]$$



Figuras 1; 3; 5. Determinación de la Forma Fundamental de un rectángulo (fig. 1) de una elipse (fig. 3) y de un triángulo (fig. 5). a) Contornos normalizados. b) Valores de abscisa y c) ordenada de los cuales se obtienen los valores del polinomio. (d,e). Los polinomios se necesitan para la definición de la forma fundamental (f). g) Diferencias entre la forma original y la forma fundamental después de rectificadas.

Figuras 2; 4; 6. Espectro de Fourier obtenido del análisis de las diferencias entre los contornos originales y las formas fundamentales del rectángulo (fig. 2), elipse (fig. 4) y triángulo (fig. 6).

A partir de la DFT de la secuencia x(n), se puede realizar una expansión en series de senos y cosenos, que esta en pares de coeficientes relativos, para los cuales se calculan los valor de amplitud y fase de cada armónica

interviniente. El desarrollo en serie de senos y cosenos será

$$X\left(\frac{k}{NT}\right) = \frac{1}{N} \sum_{n=0}^{N-1} x(nT) \cos\left(\frac{j2\pi nk}{N}\right) - j \frac{1}{N} \sum_{n=0}^{N-1} x(nT) \operatorname{sen}\left(\frac{j2\pi nk}{N}\right) \quad [3]$$

Este es un procedimiento extremadamente poderoso, el cual puede dar una descripción exacta, sin remanentes, de la forma del objeto bajo estudio gracias al cual es posible realizar una nueva síntesis parcial o total, simplemente sumando las armónicas contribuyentes. Los coeficientes de estas armónicas se calculan en orden creciente, correspondiendo al período de las relativas sinusoides, hasta la mitad menos uno del máximo de puntos (N-1) que constituyen el perfil.

El procedimiento será identificar progresivamente que puntos de la curva original están localizados a la distancia más baja de un punto determinado de la forma fundamental. De esta manera se tendrá una serie de valores positivos y negativos con, respecto a la línea de cero. La curva se distribuye sobre 2π radianes, y por lo tanto es periódica.

Desde un punto de vista clasificatorio para curvas cerradas, si se obtienen indicadores sintéticos pero eficientes para amplitud y fase y para los caracteres de distribución de las armónicas, es suficiente realizar el análisis una sola vez usando la composición resultante de los valores de diferencia obtenidos de las abscisas y ordenadas. En este caso, el signo será definido por la posición de cada punto del perfil original con respecto a la forma fundamental, *Figs. 1a, 1f, 3a, 3f, 5a, 5f*.

El gráfico de la serie de valores de coeficientes seno-coseno (Espectro de Fourier) se muestran para valores de amplitud y fase, *Figs. 2; 4; 6*.

Los parámetros sintéticos obtenidos se representan por la amplitud de la armónica fundamental, y por la media y desviación estándar de las amplitudes del conjunto entero de armónicas. **Un parámetro** de distancia morfológica entre dos curvas se representa por el error absoluto total entre las series de amplitudes relativas, correspondiente a la suma de las diferencias entre armónicas simples,

Es posible hacer comparaciones mediante superposiciones de imágenes (cráneo-foto) a través de componentes sinusoidales simples o de sumas parciales de estos componentes y crear un modelo para comportamientos de amplitud y fase.

Las amplitudes de las armónicas de Fourier de la diferencia entre la forma fundamental y la curva original son elementos del vector evaluador que se utilizará para una posterior clasificación de los cráneos.

2.4. PARÁMETROS DE ASIMETRÍA

Para hacer una descripción polinomial, la curva original y la forma fundamental se subdividen en el mismo número de puntos. Sin embargo, debido a que la forma fundamental es una aproximación de la curva original tiene un desarrollo menor, ya que contiene un cierto número de puntos con coordenadas repetidas que constituyen información remanente relativa a las irregularidades de contorno. El exceso de tales puntos se elimina para permitir la evaluación correcta de las condiciones de asimetría relativa al plano de la

forma. Esto constituye un parámetro que indica diferencias entre las curvas fundamental y original.

El procedimiento de EAF (Evaluador de Asimetría de Forma) consiste en calcular, con el método de regresión lineal múltiple, los coeficientes de una parábola del tipo

$$y=f(x)=b_0.+b_1x+b_2X^2 \quad [1]$$

y en la construcción de un conjunto que consta de un arco de parábola y un segmento de línea recta que une sus extremos (cuerda). La parábola puede variar desde un segmento de recta a un arco más o menos convexo, de acuerdo a los valores del coeficiente de orden cuadrático. Los valores de abscisa de cada punto particular se usan como variables independientes, y los correspondientes valores de ordenadas como variable dependiente.

La forma fundamental derivada de los contornos originales está sujeta a una rotación respecto a su centro de masas para una excursión completa de n : radianes con un paso angular constante de valor, ligeramente pequeño. Esta rotación se lleva a cabo para encontrar el lugar geométrico de arco/cuerda para la máxima diferencia de desarrollo entre el arco y la cuerda (máxima condición de distorsión), y también para la condición de distorsión mínima. Se calculan indicadores de estadística relativa y valor medio para cada uno de los pasos de rotación de la figura.

La condición de mínima diferencia corresponde a la situación de la mejor simetría de la figura cuando la simetría es perfecta, el coeficiente de termino cuadrático de la parábola se hace cero, el arco de parábola es derecho y coincide con la cuerda que constituye el eje de simetría. Esto ocurre si las figuras son regulares, por ejemplo en el círculo o irregulares pero simétricas Fig. 7a.

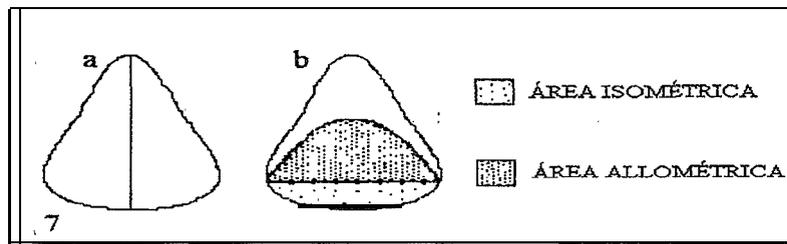


Figura 7. Análisis de simetría. a) situación de mínima asimetría (con individualización del eje de asimetría) y b) de máxima asimetría para el caso del triángulo.

Las fracciones de superficie se indican como “allométricas” (entre arco y cuerda, Fig. 7b) e “isométrica” (entre la cuerda y el perfil del lado que está opuesto a la convexidad de la parábola, Fig. 7b). En el caso en que la cuerda y el arco coinciden, el área allométrica será cero y los dominios de los dos lados de la cuerda serán iguales Fig, 7a. En cualquier otro caso, la suma de la áreas allométricas e isométricas serán igual a la mitad del área total de la curva fundamental.

El área isométrica representa la fracción que duplicada sobre el lado opuesto de la cuerda, da una figura simétrica, Fig. 8. El área allométrica es función de la fracción que modifica las condiciones de simetría de la figura que ha originado de tal duplicación. Se obtiene entonces un modelo más simple de la figura con respecto al cual la asimetría es cuantificada y expresada en

términos de porcentaje de áreas allométricas.

Debido a que la ecuación de la parábola es de segundo grado, admite tres grados de libertad donde el módulo del parámetro de EAF (Evaluador de Asimetría de Forma) está dado por el cociente, normalizado al círculo, de la suma de las longitudes de arco y de longitud de cuerda con el área allométrica. El coeficiente angular de la cuerda (en particular la posición de la normal con respecto a la cuerda) representa la dirección, mientras que la orientación de la convexidad de la parábola indica el sentido.

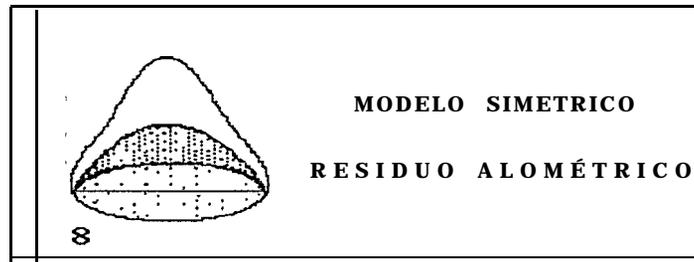


Figura 8. Modelo explicativo en el cual el área isométrica duplicada da una figura simétrica con respecto a la cual el Área allométrica es función de la fracción que modifica las condiciones de simetría.

La serie de parámetros que se obtienen con el análisis de asimetría son:

- . EAF (Evaluador de Asimetría de Forma). La dirección está dada por la perpendicular a la dirección de la cuerda. Finalmente, el sentido está formado por la convexidad de la parábola.
- . El área allométrica, que está constituida por la porción de superficie situada entre el arco de parábola y la cuerda relativa.
- . El área isométrica, representada por el dominio que no contiene la convexidad de la parábola.
- . La diferencia isométrica, dada por el área comprendida entre la cuerda y la línea recta que separa el dominio de los dos perfiles. La diferencia isométrica evalúa las diferencias dimensionales residuales después de la normalización.
- . La inclinación de la cuerda, que evalúa la diferencia de inclinación entre los dos perfiles.

2.4.1 CALCULO DE LOS PARÁMETROS DE ASIMETRÍA

Para realizar el cálculo del área allométrica e isométrica se procede a utilizar el centro de masa como semilla de inicialización en el algoritmo de llenado de regiones. Básicamente, consiste en realizar una serie de operaciones entre conjuntos de dilatación, complementos e intersecciones.

Una vez realizado el llenado de la región se debe proceder a calcular el valor numérico de las áreas allométricas e isométricas. La región llena que contiene las áreas allométricas e isométrica se muestra en la Fig. 10a. Para proceder al cálculo de las áreas, primero debe realizarse otra operación entre conjuntos de forma tal que elimine los contornos no deseados.

El área de la región, *Fig. 10b*, es la suma de las áreas allométrica e isométrica. Para la determinación individual de las áreas se procede a la construcción de una nueva región cuyo contorno estará compuesta por la parábola de asimetría 'de la Eq. 1, y una cuerda que una los extremos de la misma. Los extremos de la cuerda estarán dados por el par de coordenadas (x_1, y_1) y (x_2, y_2) de la *Fig. 9*. En este caso, el valor de las ordenadas y_1 e y_2 coinciden, indicando que la región tratada posee una asimetría mínima. la región resultante que se obtiene puede visualizarse en la *Fig. 11a*.

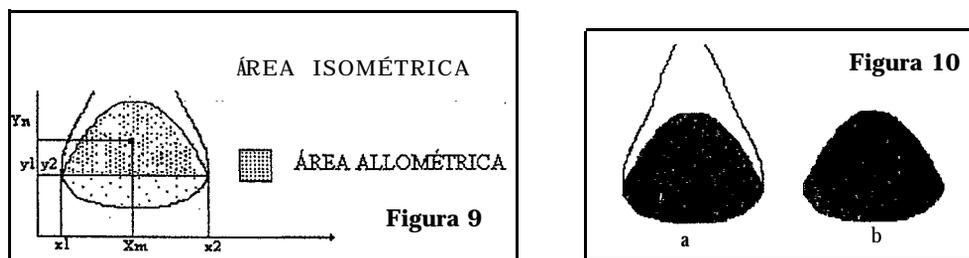


Figura 9. Coordenadas del centro de masas de la figura. Se observan además, las coordenadas de la cuerda que corta a la parábola separando las áreas allométrica e isométrica. En este caso la asimetría es mínima, coincidiendo y_1 e y_2 .

Figura 10. a) Región obtenida después de realizar el llenado de la región. b) Región obtenida después de aplicada la operación de apertura de la forma 2a.

Para poder calcular el valor de está área, se procede primero a un llenado de la región de la *Fig- 11a*. Se usa la Eq. 1 y se tiene en cuenta que la semilla inicial es un pixel inmediatamente inferior a la concavidad de la parábola. El resultado que se obtiene se aprecia en la *Fig. 11b*.

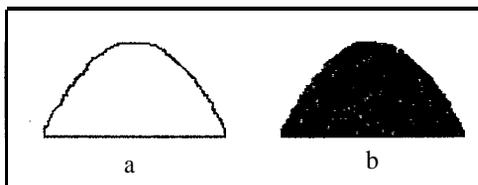


Figura 11. a) Región obtenida al combinar la parábola de asimetría con una cuerda que une sus extremos. b) Región obtenida al realizar un llenado morfológico con la implementación de Eq. 2.

Ahora se puede calcular el área de la región de a *Fig. 11b*. Su valor es el área allométrica. Con el valor de área hallado para la *Fig. 10b*, puede calcularse el valor de área isométrica restando el área de la *Fig. 11b* al área de la *Fig. 10b*.

3. RESULTADOS

En un principio el procedimiento de Morfometría Analítica se realizó para las tres figuras mencionadas, *Figs. 1; 3; 5*, Los polinomios de grado sexto para los valores de abscisas y ordenadas, *Figs. 1b, 1d; 3b, 3d; 5b, 5d* describen las formas fundamentales, *Fig. 1f; 3f; 5f* Las diferencias que permanecen entre la forma fundamental y la original, *Figs. 1f 1a, 3f, 3a; 5f, 5a*, después de la rectificación (representación gráfica en las *Figs 1g, 3g; 5g*), se describen por análisis armónico de Fourier. Las *Figs. 2, 4 y 6* reportan gráficamente los primeros 15 pares de coeficientes de senos/cosenos de Fourier para cada una de las tres figuras. Los coeficientes seno están representados por barras de

líneas; los coeficientes coseno están representados por barras llenas. La mayor parte de la información está en las armónicas de grado bajo (armónicas de baja frecuencia).

Orden	1	2	3	4	5	6	7
Rectángulo	0.5	0.7	0.6	44	0.8	11	2
	90°	81°	268°	270°	64°	276°	94°
Elipse	5	28	4	12	4	2	2
	108°	87°	97°	271°	78°	242°	104°
Triángulo	3	10	24	12	11	11	2
	185°	93°	82°	280°	90°	89°	242°

Tabla 1. Análisis armónico de Fourier realizado en un rectángulo, elipse y triángulo que tienen las mismas dimensiones de proyección. Valores de amplitud y fase hasta la séptima armónica. Los valores de amplitud se indican como porcentaje del total de 30 armónicas.

La Tabla 1 muestra los valores de amplitud de las primeras 7 armónicas como porcentaje de la suma total de las primeras 30 armónicas y sus correspondientes fases. La armónica que representa la máxima amplitud es diferente para valores de porcentaje y sobre todo para el orden de las curvas analizadas. Por lo tanto, se evidencia que mediante el análisis de los coeficientes de Fourier quedan perfectamente discriminados las distintas formas estudiadas.

La Tabla 2 muestra los porcentajes allométricos mínimos, máximo y promedio y la desviación estándar relativa también como los valores allométricos obtenidos para una rotación de 0°, es decir, en la orientación original de los perfiles. La exploración realizada con un paso angular de 10° es suficiente para indicar que en las tres formas hay al menos un eje de simetría. Aquí, el "ajuste" parabólico indica diferencias allométricas mínimas o nulas (Fig. 7a).

Allometría %	min.	máx.	med.	ds.	0° de rotación
Rectángulo	0	5.8	2.4	1.9	5.6
Elipse	0.3	5.5	3	1.4	2.5
Triángulo	0	83	55	26	83

Tabla 2. Análisis de las condiciones de asimetría de las formas fundamentales del rectángulo, elipse y triángulo realizados por el procedimiento de EAF (Evaluador de Asimetría de Forma).

Estos nuevos parámetros se complementan con los mencionados anteriormente brindando en su conjunto un vector ampliado de parámetros apropiado para una posterior clasificación.

4. CONCLUSIONES

Las técnicas del Procesamiento Digital de Imágenes permiten la extracción de características en imágenes craneales para su posterior clasificación. La Morfometría Analítica supera en gran medida a la descripción numérica que se basa en medidas discretas entre puntos de referencia canónicos. De esta manera, se obtienen una serie de parámetros que

permitirán de una manera simple y eficiente evaluar la correspondencia entre perfiles para una posterior identificación positiva de personas.

A través del análisis de Fourier y asimetría de las formas, se obtiene un vector de parámetros que será el clasificador o evaluador para determinar a qué clase pertenece el cráneo.

Los algoritmos fueron desarrollados en lenguaje C++. El software trabaja con cualquier tipo de perfil, inclusive curvas abiertas o cerradas. Actualmente, se están realizando las mediciones sobre los cráneos para una posterior verificación del sistema. Sin embargo, se ha evaluado para un gran número de formas sintéticas arbitrarias.

5. AGRADECIMIENTOS

Agradecemos a los miembros del Equipo Argentino de Antropología Forense (EAAF) por su valiosa colaboración en los temas específicos de su disciplina para que este trabajo se llevara a cabo.

6. REFERENCIAS BIBLIOGRÁFICAS

- * Mehmet Iscan, Richard Helmer, *"Forensic Analysis of the Skull"*, 1993.
- . John Hourly, *"Human Cranial Anatomy"*, 1989.
- T. White, *"Human Osteology"*, 1987
- . Anil Jain, *"Fundamentals of Digital Image Processin"*, 1992
- Peter Ralston, *"Introduction to the Numerical Analysis"*, 1987
- . F. Pessana, V. Ballarin, E. Moler, S. Torres, M. Gonzalez, *"Cuantificación Nodular mediante Propiedades Métricas Usando Procesamiento Digital de Imágenes"*, Proceedings del 3^{er} Congreso Interamericano de Computación Aplicada a la Industria de Procesos, Villa María, Cordoba, 12-15 Noviembre de 1996.
- . Yonathan Bard, *"Nonlinear Parameter Estimation"*, 1989.

SISTEMAS INTELIGENTES / INTELLIGENT SYSTEMS

TITULO:	POSICIONAMIENTO AUTOMATICO DE CELDAS NORMALIZADAS MEDIANTE REDES NEURONALES
AUTOR:	QUIJANO, ANTONIO; GAYOSO, CARLOS ; STELLI, IGOR.
INSTITUCION:	LABORATORIO DE COMPONENTES ELECTRONICOS. FACULTAD DE INGENIERIA. UNIVERSIDAD NACIONAL DE MAR DEL PLATA
DIRECCION:	J. B. JUSTO 4302 - (7600) - MAR DEL PLATA - BUENOS AIRES

TITULO:	REVISION RACIONAL DE TEORIAS BASADAS EN PLAUSIBILIDAD
AUTOR:	DELRIEUX, CLAUDIO.
INSTITUCION:	DEPARTAMENTO DE INGENIERIA ELECTRICA. ICIC - INSTITUTO DE CIENCIAS E INGENIERIA DE COMPUTACION. GIIA - GRUPO DE INTERES EN INTELIGENCIA ARTIFICIAL UNIVERSIDAD NACIONAL DEL SUR
DIRECCION:	C.C. 127 - (8000) - BAHIA BLANCA - BUENOS AIRES

TITULO:	APLICACION DE REDES NEURONALES PARA LA PREDICCION DE SERIES DE TIEMPO
AUTOR:	FORNARI, JAVIER; MARTINEZ, ERNESTO ; PEREZ, GUSTAVO.
INSTITUCION:	INGAR. INSTITUTO DE DISEÑO Y DESARROLLO
DIRECCION:	E-mail: fomariO.alDha.arcrde.edu.ar

TITULO:	ANOVERVIEW OF AN AUTONOMOUS INTELLIGENT SYSTEM WITH AN EMBEDDED MACHINE LEARNING MECHANISM
AUTOR:	GARCIA MARTINEZ, RAMON.
INSTITUCION:	CAPIS. BUENOS AIRES INSTITUTE OF TECHNOLOGY & INTELLIGENT SYSTEMS LABORATORY, COMPUTER SCIENCE DEPARTMENT, SCHOOL OF ENGINEERING. UNIVERSITY OF BUENOS AIRES
DIRECCION:	MADERO 399 - (1106) - BUENOS AIRES

TITULO:	CALIDAD DE BASES DE CONOCIMIENTO. UNA MEDIDA DE EVALUACION
AUTOR:	GARCIA MARTINEZ, RAMON.
INSTITUCION:	LABORATORIO DE SISTEMAS INTELIGENTES, DEPARTAMENTO DE COMPUTACION. FACULTAD DE INGENIERIA. UNIVERSIDAD DE BUENOS AIRES. CAPIS. ESCUELA DE POSGRADO. INSTITUTO TECNOLOGICO DE BUENOS AIRES.
DIRECCION:	M.A.D.E.R.O 399 - (1106) - BUENOS AIRES

POSICIONAMIENTO AUTOMATICO DE CELDAS NORMALIZADAS

MEDIANTE REDES NEURONALES

AUTORES: Ing. Antonio QUIJANO
Ing. Carlos Arturo GAYOSO
Igor Francisco STELLI

LABORATORIO DE COMPONENTES ELECTRONICOS
UNIVERSIDAD NACIONAL DE MAR DEL PLATA

Juan B. Justo 4302
7600 - Mar del Plata
República Argentina

Fax: 0054 23 810046
Tel: 0054 23 816600 (Int. 253)
Correo Electrónico: cgayoso@uni-mdp.edu.ar

RESUMEN

El presente trabajo trata sobre un método innovador para resolver el problema del posicionamiento (placement) de componentes en el diseño de circuitos integrados lógicos. Está orientado hacia las técnicas de implementación circuital por celdas normalizadas (*Standard Cells*).

Si en el proceso de posicionamiento se tiende a optimizar la longitud de interconexión entre *Standard Cells*, se logrará minimizar el área final de silicio del layout. Este proceso se realiza mediante una técnica basada en redes neuronales. Concretamente, las redes neuronales de Hopfield ayudadas por un método de *Simulated Annealing* tienen la habilidad de resolver problemas de optimización combinatorial en base a una función costo, la cual se obtiene de la topología del circuito de *Standard Cells*.

INTRODUCCION - TECNICA DE BIPARTICION DE GRAFOS

La técnica de bipartición de grafos es bien conocida como algoritmo clásico de resolución de problemas combinatoriales, donde las posibles combinaciones o soluciones dependen del factorial del número N de elementos del sistema. Supóngase que se simboliza a cada *Standard Cell* (SC) como un nodo de un grafo que representará el circuito a optimizar. Los nodos se unen entre sí por aristas, simbolizadas por líneas rectas entre SC's y que corresponden a las pistas que unen los distintos terminales.

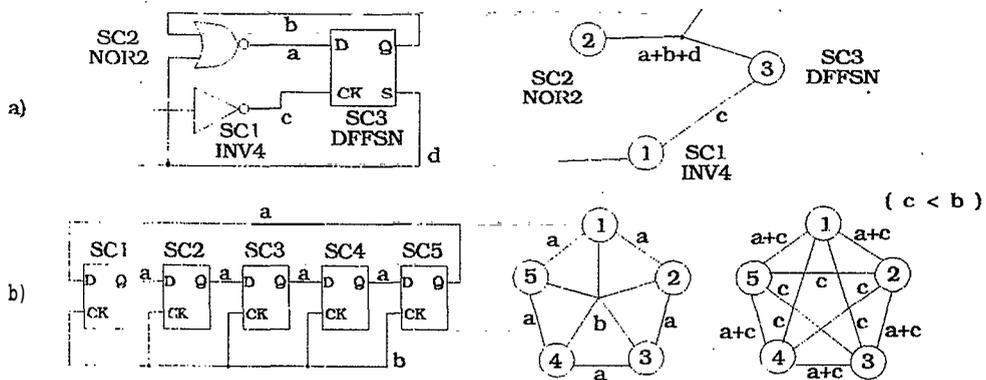


Fig. 1. Circuitos de SC's y sus representaciones de Grafos

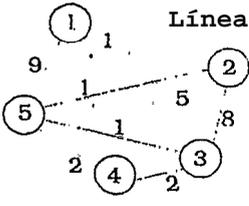
Luego, se asignan valores numéricos a cada una de dichas aristas,

intentando cuantizar el peso o importancia que tiene esa determinada conexión en el diseño del circuito. En Fig. 1. se exponen dos casos en los que deben considerarse pesos distintos. En a) dos SC's se hallan unidas por más de una pista. En b), una pista une a más de dos SC's.

Numerando cada nodo con un subíndice i, podemos referirnos a los pesos de las aristas como C_{ij} y ubicarlos en una matriz, obviamente simétrica (pues se da C_{ij} - C_{ji}). ; La Ecuación utilizada para calcular los pesos es:

$$C_{ij} = \sum_{r=1}^R \frac{1}{N_{s_r} - 1}$$

donde N_r es el número de nodos afectados a la arista r y R_{ij} es el número de aristas que unen a los nodos i y j. El método de bipartición se basa en tomar



Línea Bipartición partirla, con una línea divisoria, en dos sub-grafos de N/2 nodos cada uno. A efectos de reducir la longitud de interconexión se elegirá bipartir cortando por las aristas de menor peso. De ese modo, los-nodos menos conectados entre si quedarán en lados opuestos y aquellos con aristas de mayor peso deberán mantenerse juntos. Iterando sobre este procedimiento, tomando cada uno de los sub-bipartiéndolo, y repitiendo

Fig2. Bipartición de un

este proceso se llegará a agrupar, en grafos cada vez más pequeños, a los nodos más interconectados. Si a éstos se los ubica físicamente cerca, se estará reduciendo la longitud global de

interconexión de 'pistas; que es el efecto buscado. Luego, una secuencia de SC's con mínima longitud de interconexión pasada a un software de ruteador de canales generará una mínima área de core.

En principio, la optimización por bipartición generará una única hilera de SC's cuya longitud de interconexión es mínima o al menos muy buena, dada la cantidad de N!/2 soluciones combinatorias posibles. La hilera, deberá partirse en tramos o filas para organizarlas en un área rectangular de relación de aspecto deseada como lo ilustrado en Fig. 3. Esto se hace mediante algoritmos convencionales. El método de bipartición sólo optimiza la hilera de N Standard Cells.

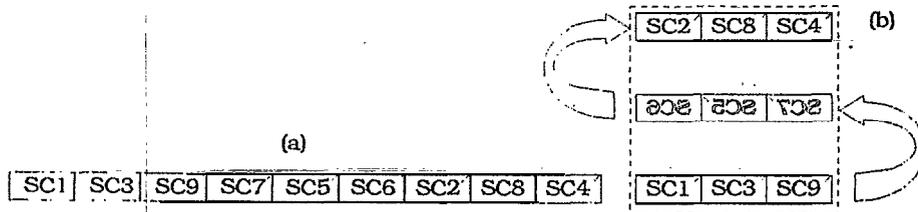


Fig. 3. a) Hilera optimizada de SC's y b) ordenamiento rectangular.

INTRODUCCION A LAS REDES NEURONALES

La esencia de la simulación de redes neuronales está en copiar la forma de funcionamiento; básica de los sistemas nerviosos centrales de los animales superiores mediante un programa de computador. El funcionamiento exacto de una neurona verdadera es muy complejo, especializado según su función, y no totalmente comprendido en la actualidad (para profundizar sobre el tema se recomienda ver [1] y [2]). Por eso se utilizan modelos simplificados de neuronas. El propuesto por Mc Culloch y Pitts [1] es uno de los más simples y efectivos y describe de una manera formal el comportamiento de una neurona, centrándose en la descripción de un modelo matemáticamente coherente y funcionalmente simple más que biológicamente estricto: Se define en base a:

- 1) La salida neuronal es un valor 'de sólo dos posibilidades discretas: puede valer +1 ó -1 solamente.
- 2) La forma en que, depende esta salida de las entradas es:

$$S_i = \text{sgn} \left(\sum_{j=1}^N S_j w_{ij} \right) = \text{sgn} (P_i) \quad \uparrow S_i$$

donde S_j es la salida de la neurona i , S_j son las entradas provenientes de las N neuronas de la red y los valores w_{ij} son los factores de ponderación o pesos con que se afecta a dichas entradas.

3) No existe retardo alguno entre entrada y salida; la neurona es matemáticamente 'instantánea'.

4) La ecuación que representa a la neurona no cambia en el tiempo, el modelo es siempre el mismo. Los valores de pesos si pueden cambiar según el funcionamiento deseado en la red.

Surge la analogía, buena en un primer momento, de identificar al modelo neuronal como un Amp. op. de ganancia infinita en configuración de sumador con resistores ponderados que vuelca entre $+V_{cc}$ y $-V_{cc}$ según el valor de entrada, tal como en Fig 5 (a). Obsérvese que los valores de los resistores podrían ser negativos (es decir pesos negativos), lo cual implicaría la inversión de polaridad de la señal S_i .

El próximo paso consiste en definir el modelo de red, es decir la estructura de la misma. Si bien existen muchos tipos de redes, la que más se presta a resolver el problema de optimización combinatorial es el llamado modelo de Hopfield. En esta topología, todas las neuronas de la red se encuentran interconectadas entre sí. De esta manera, cada neurona tiene una entrada proveniente de cada una de las otras neuronas de la red y su salida va a todas las demás neuronas. Los resistores en las entradas corresponden a los pesos w_{ij} de la red. En Fig 5 (b) se muestra una red de 4 neuronas.

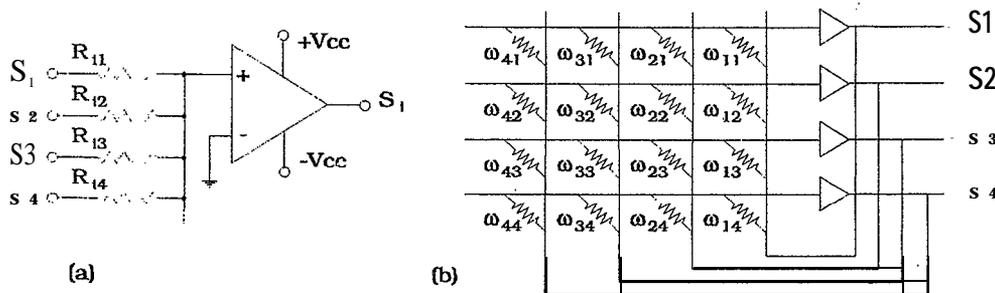


Fig. 5. Analogías con Amplificadores Operacionales.

Para simplificar el dibujo, se esquematiza cada neurona por un círculo, y los resistores pesados por sus valores numéricos en las aristas correspondientes. La función

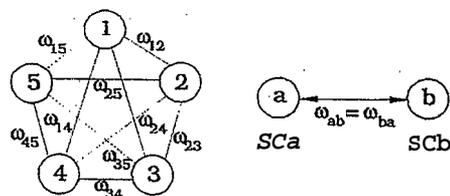


Fig. 6. Red neuronal de Hopfield esquematizada.

descriptiva de la actividad neuronal planteada en 2) de la definición de una neurona constituye un sistema de N ecuaciones no lineales acopladas con N incógnitas. He aquí la naturaleza N -dimensional del problema, pues el valor de salida de cada neurona constituye un grado de libertad. Luego el cambio de valor de una neurona puede ejercer cambios en otras por realimentación en sus entradas y este fenómeno terminar fácilmente en efectos tipo avalancha donde un sólo cambio en una neurona genere una solución estable de las ecuaciones totalmente diferente de la inicial.

Introducimos en este punto el concepto de 'evolución' de la red. Se trata de una

evolución matemática, que implica que si al conjunto de neuronas se las supone con un cierto valor de salida inicial, un estado inicial arbitrario, al iterar numéricamente sobre las N ecuaciones se irán modificando los valores de las salidas, pasando de +1 a -1 ó viceversa. Finalmente 'se llegará a una solución estable si luego de iterar repetidas veces no se producen más cambios en los valores de las salidas. Esta solución estable es de hecho una solución matemática exacta del sistema de ecuaciones. No existe una Única solución sino infinidad de ellas si N es grande. El proceso de cálculo iterativo implica que, aplicando la ecuación del modelo de McCulloch-Pitts, un nuevo conjunto de valores de salidas de neuronas se calcula en base a los valores anteriores según alguna secuencia a determinar. El nuevo valor de cada S_j tomará parte en el recálculo de la siguiente neurona s_{j+1} y así sucesivamente. Si en una actualización completa de los N valores S_j no hay cambios respecto de los N anteriores se considera la solución encontrada como estable y válida.

Considérese una función a la que se llamará energía H dada por:

donde la sumatoria es sobre todos los i y los j. El valor H es función del estado de la red neuronal. Distintas soluciones, estables o no, producirán distintos valores de energía. Mientras una red se encuentra evolucionando hacia una solución estable, se irán dando distintos valores de H. Se puede imaginar una superficie o "paisaje de energía" con valles y montañas correspondiendo a mínimos y máximos de energía en el eje z, mientras que sobre el plano X-Y se distribuyen todos los estados posibles. Un gráfico de este paisaje de energía podría ser el de Fig. 7.' Si bien es muy útil, la idea gráfica es sólo una analogía y no se debe caer en el error de olvidar la N-dimensionalidad del problema, de modo que la superficie real no es de tres dimensiones, sino de N. No es posible representarla gráficamente pero el modelo es conceptualmente correcto. Cada grado de libertad dado por cada neurona S_j implica un eje lineal de la superficie, con dos valores discretos posibles y sus dos niveles /de energía asociados.

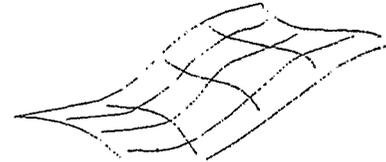


Fig. 7. Analogía del "Paisaje de Energía".

La sorprendente propiedad de esta función energía reside en que para una red neuronal simétrica que se encuentre evolucionando hacia una solución estable, la energía sólo puede decrecer monótonamente con cada iteración acorde a la ecuación dinámica, de recálculo de algún S_j [1] [2]. Así, las soluciones estables son mínimos de energía, o valles. Nuevamente, ayuda una analogía con una bola rodando por el paisaje de energía y siendo atraída a los valles por efecto de la gravedad. Cuando la bola, que representa los distintos estados por los que pasa la red, llega a un valle se detiene en el fondo, que constituye un mínimo de energía y se le llama 'atractor'. Se ha obtenido una Solución matemáticamente estable de la red. Cuanto más profundo sea un atractor y más pronunciadas sus pendientes, mayor atracción ejercerá sobre la bola y la solución estable se encontrará en menor cantidad de recálculos.

Para fusionar el método de bipartición de grafos con una red neuronal de modo que esta pueda resolver el problema de placement se propone:

- 1) Podrían utilizarse los mínimos de energía como soluciones de mínima longitud de interconexión a través de la línea de bipartición.
- 2) Los pesos del grafo de interconectividad pueden usarse para bipartir simulando pesos w_j de la red neuronal. Cada SC se simularía con una neurona.
- 3) La salida +1 ó -1 de las neuronas podría indicar el lado de la línea de bipartición del que se encuentran.

Se comienza por definir una longitud 'pesada' de interconexión L, a la que se llamará función costo:

$$L = - \sum_{i>j} C_{ij} \cdot S_i \cdot S_j$$

donde la notación i>j determina que cada arista del grafo se suma sólo una vez

(pues la matriz C_{ij} es simétrica). De inmediato surge la falsa solución trivial de dejar todas las neuronas de un sólo lado de modo que L sea el mínimo absoluto. Es necesario poner por lo tanto alguna restricción en L que 'castigue' o 'penalize' las soluciones que no tiendan a bipartir la red en dos partes iguales. La forma de generar esto es agregar un término:

$$H = - \sum_{i>j} C_{ij} \cdot S_i \cdot S_j + \mu \cdot \left(\sum_i S_i \right)^2$$

N.CI - C Oij, Si.Sj con: $w_{ij} C_{ij} = 2.p$
 $i > j$

donde se llamará a μ factor de penalización. Si la partición es en partes iguales, el segundo término es mínimo. El factor de penalización dará la debida importancia a la exactitud de la bipartición frente a la reducción de la longitud de interconexión al mínimo. Hemos llegado a un modelo exactamente igual a la función energía planteada por Hopfield y que la red intentará minimizar, implicando A) reducir la longitud total estimada de interconexión (TELL: *Total Estimated Inteconnection Length*, [4]) a través de la línea de bipartición y B) bipartir el grupo de neuronas lo más parejamente posible.

Nótese que según la constante μ , algunos valores de C . 'generarán' valores negativos de w_{ij} los cuales forzarán a dejar en lados diferentes de la bipartición a las normas S_j y S_i . Esto se deduce fácilmente observando la función energía e intentando minimizarla. La naturaleza totalmente dinámica de la red puede resolver (ser atraída por los atractores más potentes) hacia las mejores soluciones de este tipo de problemas N-dimensionales que implican ecuaciones no lineales y que numéricamente sería prácticamente imposible resolver de otro modo.

TEMPERATURAS Y TRANSICIONES DE FASE

Sea un modelo a nivel atómico de un material magnético, donde cada átomo constituye un dipolo magnético orientable al que se llama spin. En una estructura atómica regular, por ejemplo un cristal, cada átomo tiene su spin orientado según la dirección del campo magnético presente en esa posición. El campo magnético está dado por la suma de todos los campos dipolares de los demás átomos de la red y algún campo externo. La ecuación que caracteriza esto es:

$$S_i = \text{sgn}(h_i + h_{ext}) \quad \text{donde:}$$

La similitud con la red neuronal es absoluta pues se trata de un gran sistema dinámico de partículas elementales de características muy simples que interaccionan entre sí. Este modelo magnético es conocido como Modelo de Ising [1] en la mecánica estadística. Cada átomo orientará su spin acorde con esta ecuación. Como el material no tiene por que ser homogéneo, cada átomo responderá en distinta manera a los campos dipolares de los demás átomos según su posición relativa y detalles microscópicos de la red. Los factores w_{ij} que gobiernan esto pueden ser positivos o negativos, pero necesariamente simétricos. Así, un material homogéneo con todos los coeficientes iguales y positivos no es más que un ferromagneto, cuyos spins se alinearán todos en un sentido impulsados por un campo magnético inicial externamente aportado. Coeficientes todos negativos indican un comportamiento antiferromagnético. Los materiales heterogéneos tendrán dinámicas similares a las de una red neuronal, con evolución, soluciones estables y atractores.

Es conocido el efecto desordenante que tiene la temperatura sobre el spin de las estructuras cristalinas. La energía térmica de cada átomo tiende a invertir aleatoriamente los spins respecto de la posición indicada por el campo, "generando el conocido efecto de ruido térmico. Para caracterizar esto, el modelo de Ising introduce la temperatura T como parámetro según:

$$S_i = 2 \text{ Prob}(s_i=1) - 1 = \frac{2}{1 + e^{-h_i/T}} - 1 = \text{tgh} \left(\frac{h_i}{T} \right)$$

La ecuación! presentada es claramente análoga a la Función de Distribución de Fermi desplazada del origen y utilizada por la mecánica cuántica para describir la distribución de energía térmica en un grupo de fermiones. Esta función, llamada sigmoide, determina el valor medio estadístico de el spin de cada átomo orientado en una determinada dirección y desordenado por defecto el efecto térmico. Nótese que a $T = 0$ la función sigmoide se hace igual que la de actualización neuronal. Por el contrario, a altas temperaturas la sigmoide se 'achata', indicando que la probabilidad de hallar un spin en, determinada posición tiende a $1/2$, es decir, el ruido térmico tiene prevalencia sobre el campo magnético de los demás dipolos.

Surge de la introducción de la temperatura un hecho determinante en la dinámica de los modelos de Ising: las transiciones de fase. Uno esperaría inocentemente que al ir aumentando la temperatura gradualmente el desorden térmico creciese progresivamente. Pero no es así. Supóngase un material ferromagnético (FE) donde todos los coeficientes w_{ij} son iguales y positivos. La ecuación de la dinámica del modelo se reduce a:

$$S_i = \tanh(\sum_j w_{ij} S_j - h)$$

En el gráfico de Fig. 8 se han dibujado ambos lados de la ecuación. Los puntos de cruce son las soluciones matemáticas para distintos valores de T . Existen dos soluciones estables de signo opuesto y una solución trivial en 0. Esta última es inestable, pues cualquier 'flip' térmico de un spin haría evolucionar la red en algún sentido. Como todos los pesos w_{ij} son iguales, los spin tendrán todos igual probabilidad. A medida que la temperatura aumenta, esta tiende a $1/2$. Pero existe una temperatura a partir de la cual el único cruce de las curvas es en cero, por lo que la solución trivial es la única estable, es decir que a partir de ésta, la probabilidad vale: $1/2$ y la red está tan desordenada que al observarla, los spins toman valores totalmente aleatorios. Ha ocurrido una transición de fase: De ser un material FE con un cierto campo interno remanente estable, se ha convertido en un material paramagnético. El punto en que los materiales FE experimentan este cambio violento de propiedades magnéticas es la Temperatura de Curie.

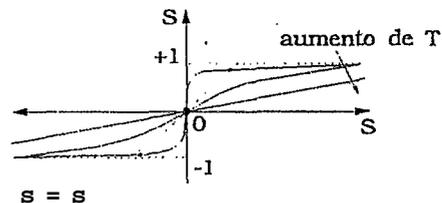


Fig 8. solución del caso FE

Este comportamiento caótico por encima de T Curie responde a un efecto de avalancha. A medida que la probabilidades se acercan a $1/2$, la suma de campos de cada dipolo magnético se torna más y más inestable, a tal punto que cerca de T se produce un efecto de avalancha de magnitud tal que todo el material pasa violentamente a la otra fase. Descendiendo luego la temperatura, el efecto se revierte, consiguiéndose alguna de las soluciones estables (cuasiestables si $T > 0$ debido a los flips térmicos) antes mencionadas.

El caso de un ferromagneto es simple, pero para materiales heterogéneos, la complejidad vuelve a ser N -dimensional. Aún así, la transición de fase existe en estos materiales, aunque se da a distintas temperaturas.

TEORIA DEL CAMPO MEDIO Y REDES DE VALOR CONTINUO

La cantidad de soluciones matemáticas estables a la dinámica de la red es grande si N es grande. Algunas de las soluciones con atractores muy amplios y profundos serán mejores, otras no tanto pues sus atractores no serán tan profundos. Como el objetivo de optimización del área de silicio implica reducir el TEIL al mínimo, es de interés llegar a estados estables de mínima energía: las mejores soluciones. Se catalogará a las mejores soluciones como mínimos globales del paisaje de energía, mientras que a las no tan buenas como mínimos locales. La calificación es totalmente cualitativa, no cuantitativa. Los mínimos locales surgen a causa de que si existen al menos dos buenas soluciones, es muy posible que exista un "estado mezcla" de estas dos. Este

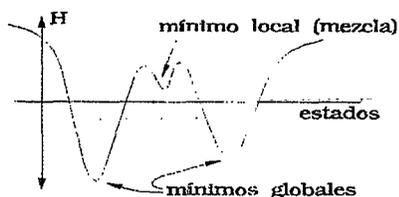


Fig. 9. Mínimos locales de la función energía.

es un atractor estable formado por partes de las dos soluciones que son mínimos globales. En Fig. 9 se ilustra este fenómeno en un paisaje de energía de una sola dimensión para aclarar el concepto. El peligro consiste en partir de un estado inicial tal que el atractor que más cerca se encuentre sea precisamente un mínimo local. La posición y número de estos estados mezcla de múltiples soluciones es totalmente desconocida e incalculable por la naturaleza combinatorial de elevado orden del problema. Incluso existen otros mínimos locales, que no están en absoluto correlacionados con ninguno de los mínimos globales. Estos estados se llaman *spin glass* y son conocidos en la mecánica estadística y los modelos de Ising e incrementan la magnitud del

problema de llegar a buenas soluciones.

Una vez que se ha llegado a un mínimo local no es posible salir de ahí, pues es una solución estable y la energía H no decrece más al recalcular. Como ésta no puede crecer, es imposible salir del mínimo local.

Para sortear esta dificultad se recurre a la Teoría del Campo Medio. Esta es otra herramienta de la mecánica estadística. Aplicándola a las redes neuronales dice que en un gran sistema, las variaciones inducidas por la temperatura sobre una neurona pueden ser tratadas como fenómenos transitorios que no modificarán el comportamiento en estado estable de la red. Los *flips* aleatorios de una neurona establecen un cierto comportamiento dinámico, particular a cada caso, pero eventualmente y a pesar de ellos, los valores medios que posee una neurona S_i son independientes del tiempo. Replanteando la ecuación de recálculo para este campo medio de las 'Salidas neuronales

$$(S_i) = \tanh (\sum_j c_{ij} \cdot (S_j))$$

Luego, los mismos estados estables que existían en el modelo de salida neuronal de valores discretos continúan existiendo en un modelo que evoluciona.. a partir de valores medios como en la ecuación propuesta, es decir que, S_i es una variable continua entre -1 y +1. Para este tipo de redes también valen todas las pruebas de monotonicidad decreciente de la energía, estabilidad de los atractores y existencia de transiciones de fase.

La solución al tema de como salvar los mínimos locales y no quedar atrapado en ellos la proporciona la temperatura y el carácter alineal de la función sigmoide. La característica esencialmente alineal del sistema de ecuaciones hace que varíe completamente la dinámica de la red pues cuando evoluciona, ésta lo hace en forma distinta según sea su temperatura, y los mínimos de energía modifican sus áreas de atracción según ésta. Se van dando Pequeñas transiciones de fase que hacen desaparecer los atractores de menor importancia. La demostración matemática de este efecto es bastante ardua, recurriendo a muchas herramientas de la mecánica estadística, como *Replica Trick*, aproximaciones 'de Saddle Point', funciones de partición y entropía [1] [2]. Se exponen, cuatro zonas cualitativamente diferenciadas: En la zona D, la red neuronal no admite otra solución que la trivial. Las zonas corresponden a los ejemplos gráficos de la izquierda de Fig. 10. Aquí, no hay mínimo alguno ni evolución posible. En la zona C, sólo existen los mínimos globales, por lo que es deseable ser atraído por ellos, hasta 'caer' al más profundo de los existentes. En la zona B aparecen los mínimos locales, pero teóricamente la red debería estar evolucionando ya dentro de un valle global de modo que se sorteó efectivamente el problema de mínimo local.

Finalmente los *spin glass* aparecen en zona A. Para esta temperatura, la solución ya está dada por el mínimo global dominante mientras se evolucionaba en la zona C. Este descenso de la temperatura en forma gradual y controlada es conocido como *Simulated Annealing* en los procesos de optimización.':'

En la gráfica se aprecia la temperatura de Curie para distintos Valores de a parámetro que indica la heterogeneidad de los pesos de red.

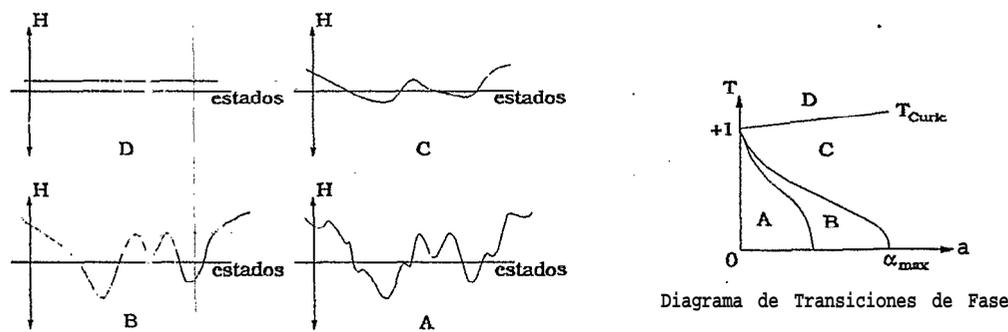


Fig. 10. Diagrama de Transiciones de Fase en función de T.

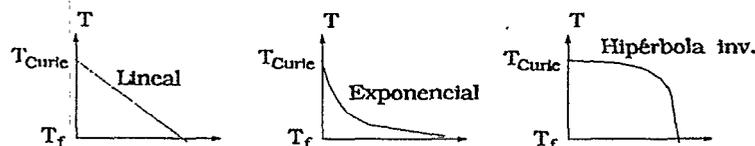
CRITERIOS PARA LA TEMPERATURA INICIAL, FINAL Y DECREMENTOS

'Por encima de T_{CURIE} existe un estado uniforme, con todos los valores S_i igual a cero. Es una excelente forma de partir desde un estado neutro, no polarizado en el sentido de que no existen atractores y por lo tanto no se podrá estar cerca de ninguno de ellos al comenzar la evolución. Al bajar T

los más poderosos de los atractores que aparezcan, es decir las mejores soluciones serán los estados más estables. La idea ahora es no decrementar la temperatura hasta que se esté en una solución realmente estable. Se verificará esta condición cuando la red no evolucione más al iterar sobre la ecuación de actualización. El análogo físico de la solución estable es esperar el equilibrio termodinámico del material que se enfría. Recién ahora podrá decrementarse la temperatura y repetirse este proceso.

La estimación de la Temperatura de Curie no es sencilla como en el caso del material FE. Para un material heterogéneo, se dispone de N ecuaciones no lineales con N incógnitas. Del gráfico de Fig. 10 no puede obtenerse T_{Curie} pues no se conoce, el parámetro a, "carga de la red". Haciendo un análisis estadístico de los pesos en un circuito típico, la función densidad de probabilidad de los valores w_{ij} indica que la red se comporta como un material heterogéneo pero fuertemente antiferromagnético. Una buena aproximación estadística es resolver una temperatura de Curie individual, $T_{Curie-i}$ para cada neurona, y luego obtener la media del valor absoluto de ellas.

Las iteraciones de recálculo a una temperatura levemente inferior al límite de Curie terminarán por arribar finalmente a una solución estable, donde se ha logrado el equilibrio termodinámico. En este punto puede bajarse la temperatura. Debe decidirse en que cantidad y de que forma. El descenso la misma puede ser en; forma lineal, exponencial, hipérbolica invertida, etc...



Fig, 11. Posibles formas de descenso de Temperatura.

No pudo determinarse para este punto ningún criterio lógico, matemático o físico que indicara la mejor de las opciones. La bibliografía consultada [1] [2] [4], apunta al método experimental para determinar el óptimo. Obviamente, dado el carácter aleatorio del proceso de bipartición pues se parte de un estado inicial $S_i = 0$, la evaluación de la mejor de las opciones debe hacerse sobre una base estadística. En el caso de optimización por

Simulated Annealing, un criterio útil es hacer descender la temperatura de forma tal que la energía (función costo proporcional al TEIL) decrezca en forma lineal [4]. El problema es que en estos modelos la energía depende del TEIL, mientras que en la red neuronal es función del costo de bipartición, que si bien reduce el TEIL no está relacionada directamente. De hecho la función energía no contempla la operación de inversión de secuencias, de vital importancia en el resultado final.

La mejor de las estadísticas correspondió al modelo lineal de descenso de T, donde se obtuvieron los mejores TEIL con reducidas dispersiones.

La cuestión restante está en cuanto debe valer el decremento en valor absoluto. Es equivalente a plantear la temperatura final T_f y el número de decrementos $6T$ constantes. Nuevamente el único criterio aportado por la bibliografía reside en "usar el $6T$ lo más pequeño posible cuidando que el tiempo de cálculo no sea inaceptablemente largo" [sic. 4]. Los experimentos estadísticos han sido la única manera de buscar el valor óptimo.

INVERSION DE SECUENCIAS - ESPEJADO

Como ya se explicó, cada bipartición generará dos grupos de neuronas al obtenerse la solución estable a temperatura T_f . Cada uno de estos grupos se bipartirá a su vez en otros menores, hasta llegar a grupos de sólo dos SC's. Llegado a este punto deberá hacerse una inversión de secuencias o 'espejado'. Esto se hace para buscar la óptima de todas las posiciones relativas entre dos bloques. Como se ve en la figura, las combinaciones posibles son cuatro.

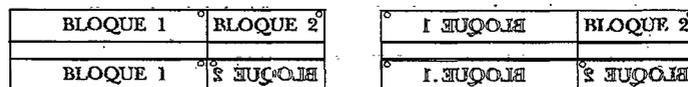


Fig. 12. Las 4 combinaciones posibles de Inversión de Secuencias.

Se evalúa el TEIL como la 'distancia' entre dos neuronas afectada por el valor de las aristas que las unen. El TEIL así obtenido es el mínimo.,

Debe notarse que ésto es sólo una inversión de la secuencia de SC's, no una inversión de las máscaras a nivel físico. La inversión de secuencias debe hacerse comenzando por las biparticiones más pequeñas, pues una vez que se ha 'reducido este bloque al mínimo TEIL pueden tomarse estas posiciones como definitivas hacerlas participar en el espejado de secuencias de biparticiones mayores.

En este punto comienza la operación de transformar la hilera de SC's optimizada en un área rectangular dividiendo la hilera en filas y canales. El número de filas se determinará en base a una relación de aspecto deseada para el core, considerando el alto de las SC's y de los canales (dependiente del ruteador de canales).

El último paso para lograr la mínima área consiste en hacer un espejado a nivel físico de las máscaras y evaluar cual de las posiciones conduce a la menor longitud de interconexión.

APLICACION

El programa de *placement* por redes neuronales forma parte de un proyecto de CAD más amplio. Consiste en una serie de paquetes de software completo para el posicionamiento y ruteo de circuitos integrados diseñados en la modalidad de celdas normalizadas, a fin de utilizar las librerías desarrolladas en el Laboratorio de Microelectrónica de la Facultad de Ingeniería de la Universidad de Mar del Plata en diferentes tecnologías (1,5 , 1,2 y 1 μ m) .

Para operación del soft fueron dibujadas las librerías de SC's, para el programa de diseño gráfico de circuitos ORCAD. El diseño se realiza totalmente sobre este soporte gráfico comercial ampliamente disponible, que permite fácilmente la inclusión de nuevos componentes o librerías. El circuito se

procesa con la utilidad NETLIST de OrCAD que genera un listado de nodos circuitales y componentes. Luego este archivo es utilizado por el posicionador basado en redes neuronales para generar la estructura del grafo de conectividad y bipartir. Una vez optimizada la posición de cada SC, se genera un archivo conteniendo estas posiciones y destinado al ruteador de canales, que es un soft implementado en lenguaje Prolog (en desarrollo) para el trazado de las pistas de interconexión de las SC. La salida en formato standard CIF de los canales ruteados es unida a la información de máscaras de las SC para proveer el layout final del circuito, también en formato CIF.

La operación de la red neuronal es totalmente transparente al operador, pues todos los parámetros de la red son elegidos automáticamente.

El sistema maneja incluso la posibilidad de agregar los PAD's de entrada y salida al circuito. Estos se dibujan directamente en OrCAD,, en donde se indica. por cual lado del CI deben estar disponibles. El posicionador y ruteador de canales se encargan de interpretarlos como tales y trazar las pistas necesarias para su conexión.

*Considerando usar el sistema en PC's compatibles, casi sin lugar a duda el soporte de programación para este trabajo es el lenguaje C. Especialmente porque se necesitan grandes estructuras de tablas en memoria. Para un circuito de 2000 SC's se necesita generar una matriz wji de $N^2/2$ elementos en precisión float, lo que implica 16 MBytes de RAM disponibles. El C compilado en 32 bits puede manejar un bloque de memoria continuo tan grande como la RAM disponible, por lo menos en procesadores tipo 486 en arquitecturas PC compatibles. De hecho la cantidad de memoria RAM es por ahora el único límite al tamaño del circuito. Eventualmente podrían utilizarse técnicas de *disk swapping* para vencer estos límites.

Según mediciones de las optimizaciones, el tiempo de cálculo depende de N^2 siendo éstos muy buenos y prácticamente despreciables en comparación al tiempo del proceso de diseño de un circuito típico..

CONCLUSIONES

Se ha presentado una parte de un conjunto de herramientas de CAD diseñadas íntegramente en la Facultad de Ingeniería de la U.N.M.d.P. La idea de la utilización de la técnica de redes neuronales para optimización del área de core no tiene ninguna referencia bibliográfica previa. Es un método innovador que corresponde a un desarrollo experimental.

Los resultados obtenidos son buenos, tanto en optimización como en velocidad. Por ejemplo para un circuito consistente en 101 SC's se obtiene soluciones hasta 13 veces la dispersión por debajo de la media de todos los TEIL (recordar que éste tiene una distribución gaussiana), siendo que el tiempo de ejecución en un procesador 486DX4-100 MHz es de 124 segundos.

BIBLIOGRAFIA

- [1] HERZ John, KROGH Anders, PALMER Richard G., "INTRODUCTION TO THE THEORY OF NEURAL COMPUTATION", Santa Fe Institute, Addison-Wesley Publishing Company, EE.UU. - 1991.
- [2] MULLER B., REINHARDT J. "NEURAL NETWORKS, AN INTRODUCTION", Springer Verlag, Berlin, RFA - Jul. 1990.
- [3] DE OLIVEIRA Elisamara, "PINCEL: LAYOUT COMPLETO DE STANDARD CELLS E BLOCOS", IV Congreso da Sociedade Brasileira de Microeletrónica, Anais 1989, Vol 2: Brasil - 1989.
- [4] SECHEN Carl, "VLSI PLACEMENT AND GLOBAL ROUTING USING SIMMULATED ANNEALING", /Yale University, Kluvier Academic Publishers, EE.UU. - 1989
- [5] GAYOSO C.A, GONZALEZ C.M, ARNONE L.J. "SISX.EMA DE DISEÑO DE CIRCUITOS INTEGRADOS AC94", Segundo Workshop IBERCHIP, Anais, Sao Paulo, Brasil - Feb. 1996.

Revisión Racional de Teorías

Basada en Plausibilidad

Claudio Delrieux

Departamento de Ingeniería Eléctrica
ICIC - Instituto de Ciencias e Ingeniería de Computación
GIIA - Grupo de Interés en Inteligencia Artificial
Alem 1253 - Bahía Blanca - ARGENTINA
claudio@acm.org

Abstract

La revisión racional o cambio de teorías lógicas ha sido intensamente investigada durante la última década. El estudio del cambio de teorías adquirió su identidad definitiva en 1985 a partir de la lógica del **cambio** de **teorías**, de Alchourón, Gärdenfors y Makinson (AGM). En la lógica AGM las operaciones más importantes son la **contracción** y la revisión. En la contracción, por ejemplo, se parte de una base de creencias K , y se desea abandonar la creencia en una sentencia s que se sigue de K , encontrándose una nueva base K' que no implique lógicamente, a s , pero que retenga de K la mayor cantidad posible de conocimiento.

‘En general, existen muchas bases K' que satisfacen dicho criterio,’ requiriéndose un mecanismo de preferencia. Dicha preferencia se establece a partir de una relación binaria $<$. El criterio generalmente adoptado es considerar que la relación $<$ es una relación de mayor o menor información. Por lo tanto, si $K_1 \subset K_2$ entonces $K_1 < K_2$. Esta propiedad tiene varias desventajas desde un punto de vista epistémico. La incorporación de información incierta o potencialmente equivocada produce un falso fortalecimiento de las diversas bases. Por dicha razón, en este trabajo se investiga otro punto de vista ‘ para seleccionar y comparar bases: El

mismo se basa en considerar que la importancia epistémica no está en el contenido informacional de las mismas, sino en su plausibilidad o credibilidad. Es decir, la preferencia se establece en función de la mejor o peor justificación de cada una de las piezas de conocimiento que constituyen cada base.

1 Introducción y motivaciones

La temática asociada: a la revisión racional de teorías lógicas (también denominada cambio **de creencias o actualización de buses de datos** en algunos contextos), ha sido intensamente investigada durante la última década. Los diversos acercamientos fueron propuestos casi exclusivamente desde dos ámbitos diferentes: en la lógica filosófica (por ejemplo en la filosofía de la ciencia [7, 8]), y en las ciencias de la computación (por ejemplo en los sistemas de razonamiento revisable [3, 4]).

El estudio del cambio de teorías adquirió su identidad definitiva en 1985 a partir de la lógica del **cambio** de teorías de Alchourón, Gärdenfors y Makinson (AGM) [1]. Dentro de la lógica AGM, las operaciones de cambio de teorías más importantes son la **contracción** y la **revisión**, donde, respectivamente, una creencia es abandonada o es modificado su valor de verdad [5, 6]. Actualmente existe consenso en aceptar que ambas operaciones son interdefinibles, es decir, que una operación de contracción puede construirse en términos de una operación de revisión, o viceversa.

En la contracción, se parte de una base de creencias K , y se desea abandonar la creencia en una sentencia s que se sigue deductivamente de K . El objetivo de la contracción es encontrar una nueva base K' que no implique lógicamente a s , pero que retenga de K la mayor cantidad posible de conocimiento. En general, existen muchas sub-bases K' que pueden ser candidatas genuinas para dicho propósito. Por lo tanto, la selección de la sub-base preferida se debe basar en algún mecanismo extralógico de preferencia entre subconjuntos de K . Dicha preferencia se establece a partir de una relación binaria $<$ que ordena parcial o totalmente los subconjuntos de K .

El criterio generalmente adoptado es considerar que la relación $<$ es una relación de mayor o menor **información**. Es decir, $K_1 < K_2$ sí y solo sí K_2 tiene mayor valor informacional que K_1 . Por lo tanto, si $K_1 \subset K_2$ entonces $K_1 < K_2$. Esta propiedad tiene varias desventajas desde un punto de vista epistémico. La incorporación de información espúrea, incierta o potencialmente equivocada produce un falso fortalecimiento de las bases de creencias. Ésto produce mecanismos inadecuados de selección para un operador de contracción.

Por dicha razón, en este trabajo se investiga otro punto de vista para seleccionar y comparar subteorías. El mismo se basa no en el contenido informacional de las mismas, sino en

su credibilidad o plausibilidad. Es decir, la preferencia se establece en función de la mejor o peor justificación de cada una de las piezas de conocimiento ' que constituyen una base de creencias.

El trabajo está organizado del siguiente modo: en la siguiente sección presentamos una descripción de los aspectos esenciales de los sistemas de revisión de teorías, "dentro de los cuales se destaca la necesidad de una relación de preferencia. En la sección 3 se presenta la fundamentación de una relación de preferencia basada en plausibilidad. Luego se reconstruyen los operadores de contracción y revisión de teorías en función de las propiedades de la mencionada relación, presentándose ejemplos que ilustran las ventajas del acercamiento propuesto. En la sección 5, por último, se discuten las conclusiones, y el trabajo futuro.

2 La lógica del cambio racional de teorías

Dado un lenguaje lógico de primer orden L , consideraremos la dinámica de las creencias de un agente idealmente racional. Este agente representa su conocimiento por medio de un conjunto K de sentencias en L . Si K es un conjunto deductivamente clausurado, es decir, si $K = \text{Th}(K)$, entonces se lo denomina **conjunto** de creencias. Esta forma de representar es computacionalmente inadecuada, por lo que existe otra representación posible¹ en la cual K es el menor conjunto de sentencias del cual es posible deducir todas las demás creencias. De esta manera, K : es una **base de** creencias.

Normalmente, el conocimiento del mundo que posee nuestro agente es incompleto. Eso determina que en una teoría lógica con K como axiomas propios existan sentencias indecidibles. La situación dual, es decir, la existencia de creencias contradictorias, es normalmente inaceptable. De esa manera, dada una sentencia s cualquiera, puede suceder que nuestro agente tenga la creencia (en la verdad) de s , si $s \in \text{Th}(K)$, o bien que rechaze (o crea en la falsedad de) s , si $s \in \text{Th}(K)$, o bien que permanezca agnóstico con respecto a s si s es indecidible (ni $s \in \text{Th}(K)$ ni $s \in \text{Th}(K)$). Dicho de otra manera, si le preguntamos a nuestro agente por su creencia respecto a s , sus respuestas posibles serán Si, creo ,que s es, verdadero o No, creo que s es falso, o No sé.

Ahora bien, como es evidente, la base de creencias de cualquier agente verosímil, está sujeta a cambios, porque el conocimiento del mundo que tiene el agente cambia, o porque cambia el mundo mismo. Ésto significa que deben darse situaciones que posibiliten u obliguen a nuestro agente a que modifique' su base de creencias, Dichas situaciones no son explícitamente consideradas en este trabajo. Más bien, nos concentraremos en las consecuencias que racionalmente deben tener sobre la base de creencias de un agente. Una situación específica' .que obligue a

1A la que adherimos en este trabajo.

modificar la base de creencias de un agente, proveniente por ejemplo de una entrada epistémica como ser los sentidos físicos, la información de otros agentes, la conjetura o suposición, etc., debe tener como consecuencia la modificación de la misma. En el marco formal de la lógica del cambio de teorías; estas modificaciones se representan por medio de operaciones. Dada una base de creencias K y una sentencia s que debe acomodarse en K , la consecuencia racional que se espera que nuestro agente efectúe queda representada como una operación que genere una nueva base de creencias K' a partir de K y s .

No es necesario enfatizar que, luego del análisis anterior, lo más importante en este tema es obtener una caracterización adecuada en términos formales de las operaciones de modificación de una base de creencias. Lo primero que se impone es encontrar un conjunto básico de operaciones, es decir, aquellas operaciones que permitan realizar un modelo adecuado de todas las situaciones epistémicas de un agente. El efecto esencial de una operación es modificar la actitud epistémica del agente respecto de una sentencia. Teniendo en cuenta que las actitudes epistémicas de un agente frente a cualquier sentencia s del lenguaje eran tres (aceptar, rechazar e ignorar), entonces pueden existir seis cambios posibles de actitud, agrupables en tres operaciones básicas. Si s era ignorada antes del cambio, y pasa a ser aceptada o a ser rechazada, entonces la operación es una expansión, porque el agente ha aumentado su base de creencias, teniendo una opinión respecto de s que antes no podía tener. En cambio, si s antes era aceptada, o era rechazada, y luego de la modificación s pasa a ser ignorada, entonces la operación es una contracción, porque el agente pasa de tener opinión a ser agnóstico respecto de s . Por último, si s era aceptada y pasa a ser rechazada, o viceversa, si era rechazada y pasa a ser aceptada, entonces la operación es una revisión porque el agente revisa su creencia respecto de s .

Recordando lo expresado más arriba, el estudio del cambio racional de teorías es esencialmente el estudio de estas tres operaciones. Dicho estudio tiene como objetivo encontrar una caracterización formal adecuada de las mismas. Dicha caracterización se efectúa (por ejemplo en [5]) por medio de postulados, es decir, de propiedades que deben cumplir las bases de creencias resultantes. ' Por ejemplo, el postulado de éxito establece que s debe ser consecuencia de expandir cualquier base K con s . Luego es necesario representar las operaciones por medio de procedimientos que las definan, y que al mismo tiempo cumplan con los postulados establecidos. Por ejemplo, la expansión puede considerarse satisfactoriamente representada por medio de operaciones conjuntistas. Si utilizamos el símbolo $+$ para denotar el operador binario de expansión de una base de creencias K : por una sentencia s , entonces $K' = K + s$ expresa el hecho de que K' es la base de creencias resultante de expandir K por s .

DEFINICIÓN 2.1 (*Levi*)

Dada una *base de creencias* K y una *sentencia* s en un *lenguaje* L , el *operador de expansión*

de K por s es

Es posible demostrar que los postulados de expansión se cumplen en la definición dada más arriba (ver [5, 6]). Pero las cosas no son tan sencillas con los otros operadores. La contracción, por ejemplo, puede pensarse como la operación dual de la expansión. Sin embargo, no es directo eliminar una creencia dada de una base, puesto que, pueden existir muchas formas distintas de hacerlo.

EJEMPLO 2.1 Sea $K = \{a, a = b\}$.

Entonces b , es también una creencia. Para contraer K por b existen dos posibilidades, abandonar la creencia en a o abandonar la creencia en $a - b$.

Más que estar relacionada Con la expansión, la contracción está directamente relacionada con la revisión. Es decir, es posible definir la contracción a partir de la revisión (con la llamada *identidad de Harper*) o definir la revisión a partir de la contracción (*identidad de Levi*). La identidad de Harper define la contracción de K por s como aquellas creencias que están a la vez en K y en la revisión de K por s . La de Levi define la revisión de K por s como la expansión por s de la contracción de K por s . Si utilizamos los símbolos $-$ para contracción y $*$ para revisión, entonces las mencionadas identidades quedan expresadas de la siguiente manera.

DEFINICIÓN 2.2

(Harper): $K - s = K \cap K * s$

(Levi): $K * s = (K - s) + s$

Los postulados que deben cumplir la contracción y la revisión, por lo tanto, están estrechamente relacionados. Mencionaremos aquí solamente los de la contracción, porque algunos de ellos son específicamente cuestionados por este trabajo.

DEFINICIÓN 2.3 (AGM)

Los postulados de un operador $-$ de contracción son:

Éxito: Si s no es tautología, entonces $s \notin K - s$.

Inclusión: $K - s \subseteq K$.

Fracaso Si s es tautología, entonces $K - s = K$.

Vacuidad: Si $s \notin K$, entonces $K - s = K$.

Relevancia: Si $t \in K$ pero $t \notin K - s$, entonces existe K' , $K - s' \subseteq K' \subseteq K$:
tal que $s \in Th(K)$ pero $s \notin Th(K \cup \{t\})$.

Extensionalidad: Si $a(S b \text{ entonces } K - a = K: - b.$

Recovery: $K \subset Th((k - a) \cup \{A\}) .$

¿Cómo debe ser un operador de revisión para satisfacer todos o algunos de estos postulados? Como vimos, el resultado de contraer una base de creencias K por una de sus creencias s es un subconjunto de K que no implique lógicamente a s . Como muchos de los subconjuntos de K : pueden tener dicha propiedad, debemos utilizar algún mecanismo extralógico que permita establecer una preferencia entre ellos. Una de las ideas esenciales es considerar la *maximilidad*, es decir, que el subconjunto resultante sea lo mayor posible, de modo que la contracción sea lo mas conservadora posible.

DEFINICIÓN 2.4 (Alchourrón y Makinson)

Sea K : una base de creencias y s una sentencia. Entonces el conjunto de residuos $K \perp s$ es el conjunto de todos los subconjuntos de K tales que no implican lógicamente a s y son maximales (no existe creencia en K que pueda agregársele sin que ello haga que impliquen lógicamente a s).

Cualquiera de los miembros de $K \perp s$ es una base de creencias que preserva lo máximo posible de K . ¿Con qué criterio se elige un elemento de $K \perp s$? La propuesta de este trabajo es establecer una comparación entre los distintos miembros del conjunto de residuos en función de la importancia epistémica de cada una de las creencias que lo componen, lo cual refleja o bien su vulnerabilidad o plausibilidad, o también su utilidad. Estos criterios serán discutidos en la siguiente sección, para luego, en la sección 4, establecer un mecanismo de contracción (o revisión) de bases de creencias basado en la importancia epistémica.

3 Importancia epistémica

Cuando nuestro agente racional está obligado a abandonar una creencia, y para hacerlo debe elegir uno entre varios conjuntos de residuos, evidentemente elegirá aquel que mantenga las creencias remanentes que considere más importantes. Este tipo de situaciones es muy común en el razonamiento científico, por ejemplo, donde la evidencia experimental puede dar por tierra con un aparato teórico bien fundamentado y establecido. Normalmente en estas situaciones se suele preferir la teoría establecida y no el experimento que la refuta (ver [9], por ejemplo).. Entonces, en una base de creencias, algunos elementos son considerados más importantes que otros, por poseer un mayor atrincheramiento epistémico [5]. En este trabajo nos referiremos a dicha propiedad simplemente como importancia epistémica.

Gärdenfors introdujo un dispositivo teórico para representar la importancia epistémica [5]. Dadas dos sentencias a y b , la relación $a < b$ expresa el hecho de que la sentencia b es de mayor importancia epistémica que la sentencia a . Es importante destacar que la importancia epistémica tiene un significado exclusivamente pragmático, es decir, no es ni sintáctico ni semántico, tratando de representar el grado de soporte asertivo de una determinada sentencia [2]. A diferencia de la probabilidad, la importancia epistémica de una sentencia **es externa** al contenido intensional de la misma. Por ejemplo, una amiga arroja un dado y nos dice que salió el 4. La probabilidad de dicho evento es $1/6$. Pero nosotros no tenemos razones para desconfiar de nuestra amiga, por lo que la importancia epistémica que para nosotros tiene el **reporte** es mucho mayor que $1/6$.

La relación de importancia epistémica propuestas por Gärdenfors debe satisfacer determinados postulados. Entre ellos se cuenta la **conjuntividad**: o bien $a < a \wedge b$ o bien $b < a \wedge b$. Este postulado expresa que una conjunción nunca es menos importante epistémicamente que cualquiera de sus elementos. Por lo tanto, el acercamiento de Gärdenfors a la importancia epistémica está estrechamente relacionado con una noción de información. Una sentencia será considerada más importante que otra si provee mayor. (y no mejor) información.

Es importante destacar aquí lo inadecuado de esta caracterización de la importancia epistémica. Consideremos nuevamente el problema científico en el cual la evidencia experimental da por tierra con una teoría. El comportamiento normal de la comunidad científica de rechazar el experimento coincide aproximadamente con el criterio de la mayor información. Siempre las teorías científicas se expresan con sentencias generales, y los experimentos son siempre particulares. Por lo tanto, el criterio nómico de la teoría de la ciencia implica conservar la mayor cantidad de información al efectuar una contracción por inconsistencia. Si ésto fuese así, entonces las teorías nunca cambiarían. Pero también es importante en el razonamiento el origen o la justificación del conocimiento que se utiliza. Y dentro de este marco, las teorías científicas son siempre especulaciones, más o menos fundadas. En cambio, la evidencia experimental debidamente confirmada es verdadera por ostensión. Esto explica por qué las teorías, aún las fuertemente 'establecidas, pueden llegar a caer. Tal vez el caso histórico más famoso es el derrumbe de la mecánica de Newton, una teoría que fue, corroborada infinidad de veces. Sin embargo, ciertos experimentos astronómicos y subatómicos presentaban resultados anómalos, y previo a la postulación de la Teoría de la Relatividad, ya gran parte de la comunidad científica operaba con sistemas que se apartaban de la mecánica de Newton.

Lo que ocurre en esos casos es, simplemente, que se elige la teoría mejor fundamentada, y no la que provee mayor información. El razonamiento basado en premisas de mayor o menor credibilidad fue estudiado, entre otros, por Rescher bajo el nombre de Razonamiento Plausible [10, 11]. La propuesta de Rescher consiste en asignar a cada fórmula de un lenguaje proposicional un índice de plausibilidad en una familia $\{1/n, 2/n, \dots, n-1/2, 1\}$ para un n dado. La plausibilidad de cualquier fórmula del lenguaje puede encontrarse con un conjunto de

desigualdades. Ante inconsistencias, el criterio de racionalidad es rechazar la premisa menos plausible que las origina. Los detalles pueden consultarse en las obras citadas, pero son innecesarios para este trabajo, ya que en este punto nuestro enfoque se aparta del enfoque de Rescher por varias razones.

Es importante observar que el razonamiento plausible no obedece la conjuntividad, más bien, obedece la anticonjuntividad: o bien $a \wedge b < a$ o bien $a \wedge b < b$. La razón de esto es sencilla, la plausibilidad de una conjunción debe ser menor o igual que la plausibilidad de la más débil de las sentencias que la componen. Desde el punto de vista de la información, agregar sentencias a una conjunción aumenta la información. Desde el punto de vista de la plausibilidad, en cambio, cada sentencia en una conjunción aumenta su vulnerabilidad.

EJEMPLO 3.1 *Un juez se encuentra con el siguiente caso:*

a: el reo fue visto cometiendo un robo.

b: el reo declara que fue por equivocación.

a = c: el autor de un robo debe ser condenado.

a \wedge b + c: si existe! justificativos no debe ser condenado.

b - j: el error o la equivocación son justificativos.

¿Debe el juez condenar al reo?

Tenemos dos posibles razonamientos. Con base a solamente, se concluye c , y con base a $a \wedge b$ se concluye c . Desde el punto de vista de Gärdenfors, el razonamiento con base en $a \wedge b$ es más importante, por lo que el reo debe ser absuelto en base a sus dichos. En cambio, desde el punto de vista del razonamiento plausible, la única manera en que el reo sea absuelto es que se justifiquen sus dichos de modo tal que sean tan plausibles como la denuncia misma.

En este trabajo utilizaremos entonces una valoración de la importancia epistémica inspirada en el razonamiento plausible, pero con algunas modificaciones. En principio parece más razonable referirse a la plausibilidad de la fuente de información que a la plausibilidad de la información misma. Por dicha razón adoptamos la visión de asignar plausibilidades a las sentencias en función del origen que las justifica. Por otra parte, un orden lineal parece inadecuado. No siempre es posible comparar la plausibilidad de las distintas justificaciones y decidir en quién confiamos más. Por dicha razón adoptamos un orden parcial para representar la plausibilidad de las fuentes de información.

4 Cambio de teorías basado en plausibilidad

Los operadores de contracción basados en importancia epistémica que se han propuesto en la literatura están definidos para conjuntos (clausurados) de creencias, y no para bases de creencias, y se estima que es muy difícil adaptar la definición de los operadores para conjuntos a bases de creencias. Sin embargo, nuestra definición de un operador de contracción basado en plausibilidad será sencilla. Primero necesitamos establecer un mecanismo de comparación de subconjuntos de una base de creencias, K a partir de la plausibilidad de cada una de las creencias. Luego, dado que un operador de contracción elige un elemento del conjunto de residuos (los cuales son subconjuntos de K), entonces será posible comparar el más plausible de los elementos de dicho conjunto de residuos.

DEFINICIÓN 4.1 Dada una base de creencias k , parcialmente ordenada por una relación de plausibilidad $<$, es decir cada par de elementos a, b en K : es tal que o bien $a < b$, o bien $b < a$ o bien no se pueden comparar. En este caso, la cota de plausibilidad Pk' de un subconjunto K' de dicha base se define como el conjunto de mayores cotas inferiores bajo $<$ de la base de creencias que conforman a K' :

EJEMPLO 4.1 Supongamos que en una determinada base de creencias encontramos un subconjunto $K' = \{a, b, c, d, e\}$ y en la relación de plausibilidad encontramos $(a < b, a < c, d < c)$. Entonces $Pk' = \{a, d\}$, porque no existe ningún elemento en K' que sea de menor plausibilidad que los pertenecientes a dicha cota.

DEFINICIÓN 4.2 Un subconjunto K' epistémicamente más importante que otro K'' si y solo si cada uno de los elementos de Pk' es a lo sumo tan plausible como cada uno de los elementos de Pk'' pero contiene por lo menos un elemento estrictamente más plausible. En dicho caso escribiremos $Pk'' < Pk'$, o simplemente $K'' < K'$, sobrecargando la relación $<$.

EJEMPLO 4.2 Supongamos que en el ejemplo anterior tenemos que comparar a $K' = \{a, b, c, d, e\}$ con $K'' = \{b, c, d, e\}$. En este caso se ve que $Pk' = \{a, d\}$ y que $Pk'' = \{b, c, d\}$. Por lo tanto, $Pk'' < Pk'$

DEFINICIÓN 4.3 Dada una base de creencias K y una sentencia s , definimos el operador de revisión - basado en plausibilidad como encontrando el elemento epistémicamente más importante del conjunto de residuos.

$K : -s = K' \iff K' \text{ es el subconjunto de } K \text{ tal que } K' < KY$, si y solo si no existe $K'' \text{ tal que } K'' < K' \text{ y } K'' \text{ es un subconjunto de } K \text{ tal que } K'' \text{ contiene a } s$.

EJEMPLO 4.3 Supongamos que tenemos una base de creencias

$IC - (a, b, a \sim c, b \sim d, c = e, d \sim e),$

con una relación de plausibilidad

$(a + c < b + d, d = e < b + d, b + d = c + e, b + d < a, b + d < b).$

En ese caso, $K: - e = K' = (a, b, b \sim d, c \sim e)$. Es fácil ver que no se puede agregar a K' ningún elemento en K sin que vuelva a implicar lógicamente a e , y que cualquier otro subconjunto en las mismas condiciones es menos importante epistémicamente.

Para definir el operador $*$ de revisión podemos recurrir a la identidad de Levi.

EJEMPLO 4.4 Supongamos que tenemos la misma base de creencias que en el ejemplo anterior, y buscamos revisar la creencia e por e . En ese caso, $K: * e = IC' + e = IC' \cup \{e\} = \{a, b, b + d, c \sim e, Te\}$.

5 Conclusiones y trabajo futuro

En este trabajo se investigó un nuevo enfoque para el cambio racional de teorías, en función de un criterio para seleccionar y comparar subteorías. El mismo se basa no en el contenido informacional de las mismas, sino en su credibilidad o plausibilidad. Es decir, la preferencia se establece en función de la mejor o peor justificación de cada una de las piezas de conocimiento que constituyen una base de creencias. El criterio generalmente adoptado era considerar una relación de mayor o menor, información. Se mostró que esta propiedad tiene varias desventajas desde un punto de vista epistémico.

Se presentó la fundamentación de una relación de preferencia basada en la plausibilidad de las creencias. Luego se reconstruyeron los operadores de contracción y revisión de teorías en función de las propiedades de la mencionada relación, presentándose ejemplo que ilustran las ventajas del acercamiento propuesto. Entre el trabajo futuro, podemos destacar necesidad de encontrar los teoremas de representación que cumple el operador de contracción aquí definido, es decir, cuáles postulados obedece.

Referencias

- [1] Carlos Alchourón, Peter Gärdenfors, and David Makinson. On The Logic of Theory Change. *The Journal of Symbolic Logic*, 50(2):510-530, 1985.

- [2] Roderick M. Chisholm. *Theory of Knowledge*. Prentice Hall, Englewood Cliffs, New Jersey, 1977.
- [3] Jon Doyle. A Truth Maintenance System. *Artificial Intelligence*, 12(3):231-272, 1979.
- [4] R. Fagin, G. Ullman, and M. Vardi. On the Semantics of Updates in Databases. *ACM SIGACT-SIGMOD*, pages 352-365, 1983.
- [5] Peter Gärdenfors. *Knowledge in Flux, Modeling the Dynamics of Epistemic States*. The MIT Press, Cambridge, Massachusetts, 1988.
- [6] Sven O. Hansson. A 'Textbook on Belief Dynamics. Technical Report Manuscript, Department of Philosophy, Uppsala, Sweden, Feb 1996.
- [7] W. Harper. Rational Conceptual Change. *PSA*, pages 462-494, 1977.
- [S] Isaac Levi. *The Fixation of Belief and its Undoing*. Cambridge Univ. Press, 1991.
- [9] Nicholas Rescher. *Scientific Explanation*. McGraw-Hill, New York, 1969.
- [10] Nicholas Rescher. *Plausible Reasoning*. Van Gorcum, Dodrecht, 1976.
- [II] Nicholas Rescher. *Dialectics, a Controversy-Oriented Approach to the Theory of Knowledge*. State University of New York Press, Albany, 1977.

APLICACION DE REDES NEURONALES PARA LA PREDICCIÓN DE SERIES DE TIEMPO

Fornari J.F., Martínez E.C., Pérez G.A.
INGAR, Instituto de Diseño y Desarrollo
Avellaneda 3657 - 3000 Santa Fe
Tel: +54-42533451 / Fax: +54-42553439
E-mail: {ecmarti, fornari, ~gus}@alpha.aicride.edu.ar

Resumen:

Las redes neuronales artificiales han sido estudio para la investigación en diversos campos del conocimiento y emergen como un revolucionario y poderoso paradigma computacional alternativo a las tecnologías tradicionales.” **En este** trabajo se compara la utilización de redes ,neuronales artificiales para la predicción del cierre del índice S&P500, utilizado en el contexto de análisis de series de, tiempo financieras. Se : trata de comprobar que la utilización de herramientas *no* tradicionales en el área del mercado de valores, resulta más eficiente en la solución de *ciertos* problemas de predicción, que la utilización de las curvas de ajuste tradicionales.

Palabras Claves: redes neuronales artificiales, estrategias financieras, mercado bursátil.

1. Introducción.

La habilidad para detectar patrones a través del tiempo ha ‘sido de suma utilidad para la humanidad. Los.-sacerdotes de civilizaciones antiguas usaban su conocimiento de la astronomía para predecir el pasaje de estaciones, el curso de la lluvia y el crecimiento de las plantaciones de cereales., Hoy en día, una de las aplicaciones mas útiles de las redes neuronales artificiales’ ‘en los negocios es la habilidad. de capturar relaciones en la predicción de series de tiempo. Conociendo cual *es la* dirección que un mercado esta encabezando o identificar un producto pedido **antes** que los competidores lo hagan, tiene implicaciones obvias en la manera de realizar negocios.’

La predicción de series de tiempo no es una ciencia exacta, puesto que no todos los factores son conocidos y sus interrelaciones son altamente complejas como para ser tratados por un simple modelo matemático. Existen trabajos que demuestran que una red neuronal artificial (con topología backpropagation) es capaz de predecir datos de una serie de tiempo con una performance superior a las tradicionales técnicas como lo es el método predictivo lineal o el ajuste por medio de minimo cuadrados.

Las observaciones efectuadas de las fluctuaciones diarias en los precios de las acciones del Mercado de Valores, inspiró la posibilidad de analizar dichas fluctuaciones.,en .el marco de ciertos índices financieros usados actualmente. En este caso se eligió el cierre del índice S&P500, ya que es uno de los más utilizados a nivel mundial, especialmente en la Bolsa de Valores de Norteamérica.

Concretamente se trata de demostrar que la evolución’ bursátil puede ser analizada con métodos no tradicionales y de una manera más exacta, permitiendo

una mejor performance en el resultado que se desea obtener. Es interesante resaltar que el motivo en la adopción del índice de cierre S&P500, responde a sus características especiales de “dinamismo y complejidad”. El objetivo principal perseguido consiste en demostrar la sustancial superioridad del paradigma neuronal en el campo predictivo respecto a técnicas tradicionales.

2. Redes Neuronales Artificiales.

Una red neuronal artificial es una arquitectura de procesamiento en paralelo distribuido integrada por unidades de procesamiento masivamente interconectadas, también denominadas neuronas o nodos. La red neuronal se compone de nodos organizados en niveles. Usualmente existen tres niveles, el nivel de entrada, el oculto y el de salida. El nivel de entrada recibe los datos externos al sistema. El oculto presenta los resultados al entorno externo o nivel de salida. El número de nodos es determinado por el problema a resolver, mediante sus entradas y salidas.

En la actualidad existen diferentes tipos de modelos o paradigmas de redes neuronales. En cada conferencia que se realiza sobre este tema se presentan literalmente cientos de variaciones. Los modelos más populares de redes neuronales pueden clasificarse en base al paradigma de aprendizaje, su topología de conexión básica y sus funciones de procesamiento y capacidades.

Paradigmas de Aprendizaje:

- *Aprendizaje supervisado*: en este enfoque, se le presenta un problema a la red neuronal y realiza una predicción o clasificación. El algoritmo de aprendizaje realiza la diferencia entre la salida deseada y la predicción actual y utiliza dicha información para ajustar los pesos de la red neuronal. Por lo tanto, en el próximo paso la predicción estará mas cerca de la respuesta correcta.
- *Aprendizaje no supervisado*: es el caso en el cual disponemos de mucha información de entrada, pero no conocemos la salida o respuesta. La pregunta es: “¿Cómo están relacionados estos datos?” En efecto, lo que realiza la red neuronal en este caso es una agrupación en diferentes conjuntos de los diferentes patrones o información de la que disponemos.
- *Aprendizaje reforzado*: en este paradigma disponemos de ejemplos del problema pero no conocemos la respuesta exacta, o por lo menos, no la conocemos inmediatamente. Realizamos una serie de decisiones, y solamente al final de las acciones que realizamos podemos saber si lo hicimos bien o mal.

Topologías de Conexión:

- *Redes de alimentación hacia adelante*: en este tipo de red neuronal, la información fluye a través de la red en una sola dirección, y la respuesta se basa en el conjunto actual de entradas.
- *Redes referentes limitadas*: la información acerca de entradas pasadas es retroalimentada y mezclada con las entradas recurrentes de la retroalimentación.
- *Redes totalmente recurrentes*: provee conexiones de dos vías entre todos los procesadores en la red neuronal. La información de la entrada circula hacia adelante y hacia atrás hasta que la activación de las unidades o procesadores se estabiliza.

Modelos de Redes Neuronales:

- Backpropagación.
- Mapa de Características Autoorganizativas (Kohonen)
- Teoría de Resonancia Adaptiva.

3. Problema de Predicción y Composición del Modelo de Datos.

Para nuestro ejemplo de predicción, hemos seleccionado como variables el índice S&P500, cuyos valores son alto (el más alto de la jornada), bajo (el más bajo de la jornada) y un promedio de la diferencia de ambos (alto y bajo) de los últimos cinco días, De esta manera, lo que intentaremos predecir será, el índice S&P500 de cierre de los próximos cinco días (tabla 1).

Entradas	Salida
Indice S&P500 Alto	Indice S&P500 Cierre
Indice S&P500 Bajo	
Promedio de los últimos cinco días	

Tabla 1. Entradas y salida de la red neuronal artificial

Las variaciones de precios diario del **S&P500** se tomó durante los tres primeros meses del año 1994. Tres redes neuronales artificiales fueron entrenadas separadamente, cada una con un horizonte de tiempo diferente. La primera red fue entrenada con las variaciones de precios del primer mes. La segunda red con las **variaciones** del primero y segundo mes, y la tercera red con las variaciones de precios de los tres meses. La serie de tiempo de nuestro interés se muestra en la figura 1 (tres meses).

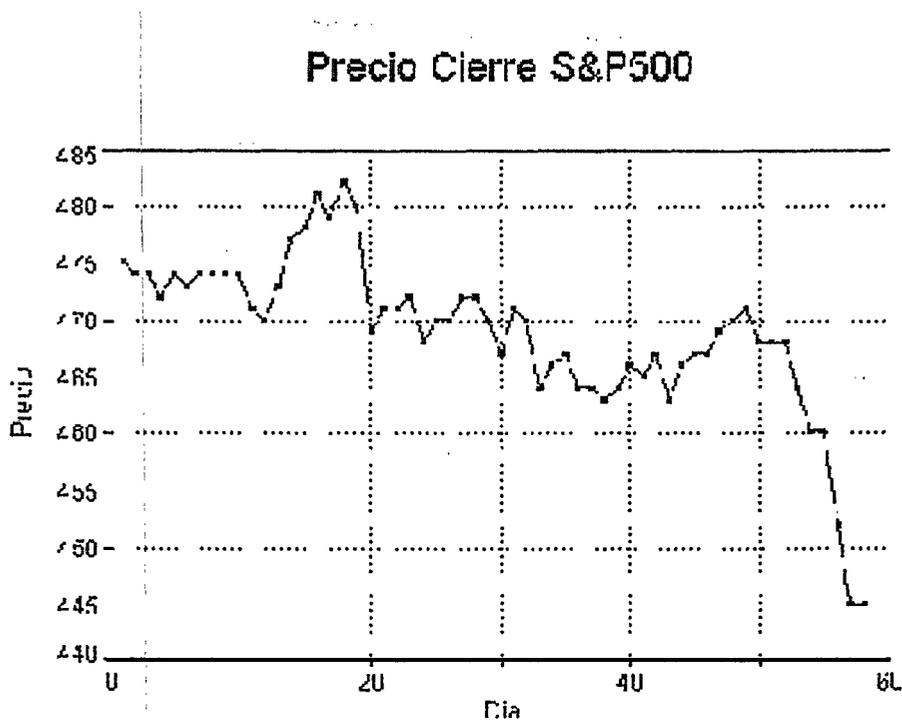


Fig: 1 - Serie de Tiempo S&P500 (primer trimestre 1994)

4. Selección Arquitectura de la Red

Existen ciertos elementos claves para la selección del modelo de red neuronal a utilizar en esta aplicación particular. Primero se debe seleccionar el tipo de función que se desee realizar (clasificación, clustering, modelado, aproximación de series de tiempo): En este caso aproximación de series de tiempo. Segundo, seleccionar los datos de entrada para realizar el entrenamiento de la red, y la salida que se desea aproximar, seleccionando de esta manera la mejor arquitectura (selección de la cantidad de nodos en la capa de entrada, en la capa de salida y en la capa oculta).

La arquitectura de la red neuronal artificial elegida fue el perceptron multicapa, el cual fue; entrenado utilizando el método de backpropagación estándar. La capa de entrada posee tres nodos, mientras que la de salida solamente uno. Los nodos de la capa intermedia fueron calculados en base a la cantidad de patrones de entrenamiento. Por lo tanto para la primera red neuronal se utilizaron 3 nodos en la capa intermedia, en la segunda red 6 nodos y en la tercera red 9 nodos.

	SP500A	SP500B	Dif.	SP500C		SP500A	SP500B	Dif.	SP500C
1	475,27	469,54	2,428	475,27	30	471,09	466,07	3,854	467,69
2	475,28	473,27	3,058	474,13	31	471,65	467,58	4,01	471,46
3	475,06	472,14	2,97	474,17	32	472,4	469,46	4,236	470,69
4	474,17	471,8	3,174	472,47	33	470,65	464,27	4,188	464,28
5	475,32	472,5	3,252	474,91	34	466,48	464,26	5,018	466,07
6	474,91	472,83	3,17	473,3	35	469,16	466,08	4,126	467,14
7	475,19	473,29	2,44	474,25	36	467,43	462,02	3,738	464,44
8	474,7	472,21	2,418	474,3	37	464,87	457,49	4,006	464,81
9	475,01	473,42	2,332	474,98	38	464,83	462,49	4,894	463,01
10	475,56	473,72	2,176	474,72	39	466,16	462,41	4,086	464,74
11	475,2	471,49	1,98	471,97	40	468,07	464,74	4,392	466,91
12	472,55	470,27	2,306	470,92	41	467,78	465,02	4,442	465,88
13	473,44	470,72	2,382	473,2	42	467,42	463,4	3,912	467,06
14	477,52	473,2	2,428	477,05	43	467,29	462,46	3,24	463,9
15	479,45	477,06	2,974	478,7	44	466,61	462,54	3,738	466,45
16	482,85	478,7	3,084	481,61	45	467,6	466,08	3,802	467,39
17	481,64	479,18	3,172	479,61	46	468,99	466,04	3,44	467,01
18	482,23	479,57	3,208	482,02	47	469,85	465,48	3,478	469,42
19	481,96	478,71	3,196	480,71	48	471,04	468,62	3,548	470,9
20	481,02	469,28	2,982	469,81	49	471,09	467,83	3,066	471,06
21	472,08	467,57	4,852	471,76	50	471,06	467,23	2,904	468,54
22	472,33	469,5	4,924	471,05	51	470,47	467,88	3,366	468,8
23	473,41	471,05	4,998	472,77	52	470,38	468,52	3,294	468,54
24	473,13	468,91	4,938	468,93	53	468,57	462,42	2,792	464,35
25	471,13	466,89	5,132	470,19	54	465,29	460,58	3,538	460,58
26	471,99	469,05	3,632	470,23	55	461,12	456,1	3,828	460
27	473,41	470,23	3,318	472,52	56	460,33	452,43	4,066	452,47
28	474,16	471,93	3,388	472,79	57	452,49	445,54	5,128	445,55
29	475,12	468,44	3,362	470,34	58	447,16	436,16	6,146	445,76

Tabla 2. Entradas y salida utilizados para la red neuronal

5. Selección de la Curva de Ajuste.

Mediante la utilización de un programa de ajuste de curvas se procedió a realizar el cálculo para la obtención de la curva que mejor aproximara a todos los datos del primer trimestre del precio de cierre del índice S&P500. El cálculo efectuado procedió a realizar el ajuste con diferentes tipos de curvas, pero solamente se eligieron aquellas que aproximasen en forma no lineal al conjunto de datos, debido que la información seleccionada es de carácter no lineal, y generalmente como una red neuronal artificial se comporta como un modelo de regresión no lineal se procedió con esta selección.

Del proceso de ajuste resultó seleccionada la siguiente función: $y = a + bx^c$, donde $a = 473.7952204$, $b = -0.000111297$ y $c = 2.96113579$. De esta manera se obtuvo una curva de ajuste para poder realizar un comparativo con los resultados obtenidos con la red neuronal.

Los valores que se obtuvieron con esta curva de ajuste se representan en la tabla 3. En la figura 2 se puede observar la gráfica de la curva de ajuste junto con los datos originales del precio de cierre del índice S&P500.

Precio Cierre S&P500

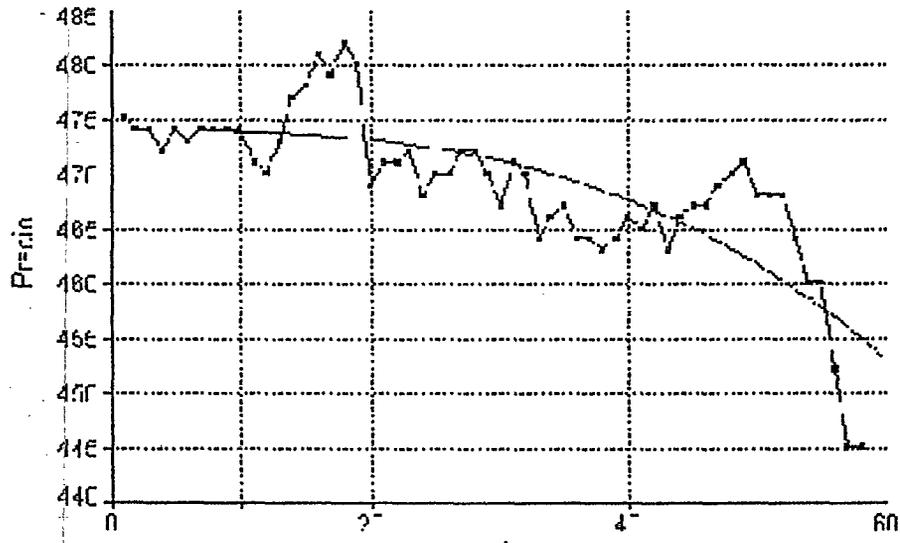


Fig. 2 - S&P500 Cierre con Curva de Ajuste

Valor Aprox.	Dif. con S&P500 ^2
473,7951	2,175308
473,7943	0,112667
473,7923	0,142659
473,7884	1,738098
473,782	1,272485
473,7725	0,223219
473,7593	0,240792
473,7419	0,311509
473,7196	1,588593
473,6919	1,056947
473,6582	2,850144
473,618	7,27909
473,5706	0,137325
473,5154	12,49305
473,452	27,54114
473,3798	67,73679
473,2981	39,84048
473,2064	77,67985
473,1041	57,84905
472,9908	10,11741
....
....
463,0467	61,67356
462,37	75,51648
461,6657	47,25615
460,9332	61,88705
460,1719	70,02461
459,3814	24,68666
458,5611	4,075803
457,7105	5,241787
456,829	19,00083
455,9161	107,4553
454,9712	84,84582

Tabla 3. Valores de la curva de Ajuste y diferencia al cuadrado

En promedio, el error que se obtuvo (diferencia entre el valor obtenido mediante la curva de ajuste y el valor real elevado al cuadrado) al usar la curva de ajuste para los valores del mes de abril fue de un 16 %. Se podrían haber utilizado otras técnicas modernas de portfolio y estrategias de mercado para realizar una curva de ajuste mejor, pero en todos los casos se demuestra en la conclusión que las redes neuronales artificiales son superiores en todo aspecto.

6. Predicción de las Redes Neuronales Artificiales.

Los resultados de la predicción pueden observarse para en las figuras 3, 4 y 5 para cada red neuronal respectivamente. En cada figura, A representa al valor real del índice de cierre S&P500, mientras que B representa la predicción hecha por la red neuronal artificial.

En el caso de la primera red neuronal obtuvimos un error promedio general del 5% (el mejor en las tres redes neuronales), calculando el error de la forma anterior.

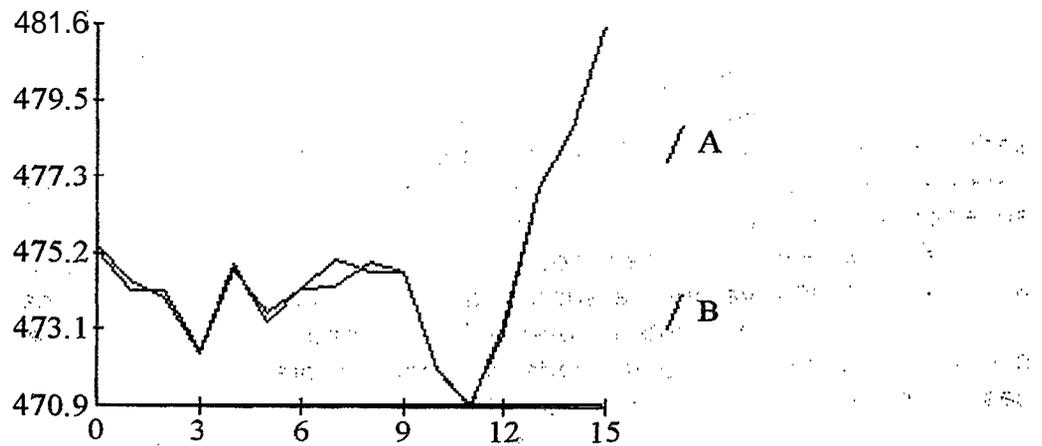


Figura 3. Predicción Primera Red Neuronal

En el caso de la segunda red neuronal obtuvimos un error promedio general del 10%.

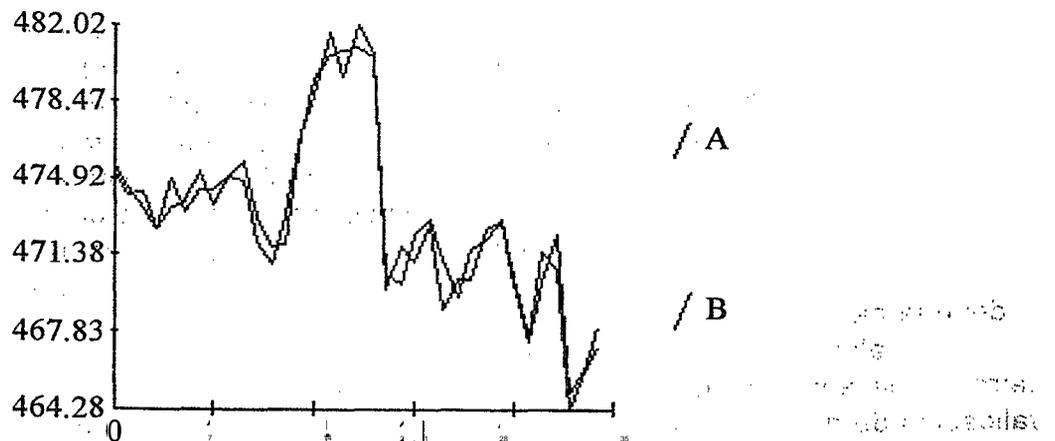


Figura 4. Predicción Segunda Red Neuronal

En el caso de la tercera red neuronal obtuvimos un error promedio general del 13%.

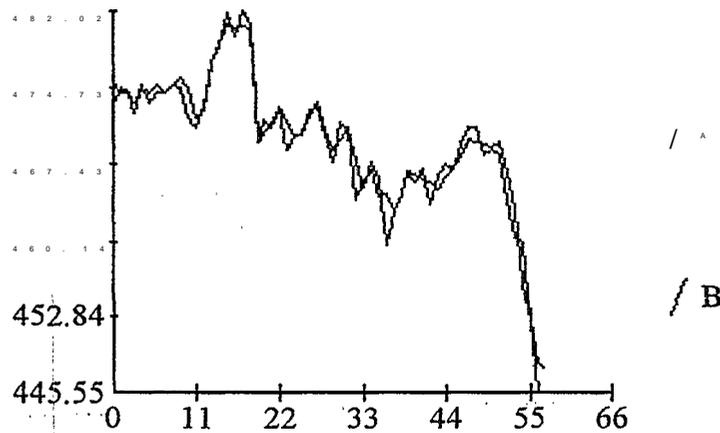


Figura 5. Predicción Tercera Red Neuronal

En los ejemplos presentados aquí, tres redes neuronales artificiales, cada una entrenada independientemente para extraer las características de la misma serie de tiempo pero a diferentes horizontes de tiempo donde se utilizó la red estándar de backpropagation. En todos los casos se obtuvieron resultados superiores a la mejor curva de ajuste.

En los casos de la segunda-y tercer red neuronal los resultados obtenidos no son satisfactorios, ya que se supone que a medida que la red crezca (aumentando los nodos de la capa intermedia) y la cantidad de información recolectada sea mayor, la red debería ajustar cada vez mejor la predicción de los datos. En estos Casos, no se cumplió, debido a otros factores como ser:

- la cantidad de, nodos utilizados en la capa intermedia debieron ser suficientemente grandes que no permitieron una mejor aproximación de la predicción, sino que la red solamente memorizó los valores de la serie de tiempo en estudio en vez de lograr la generalización
- los valores de la serie de tiempo no fueron suficientes para lograr una mejor aproximación, sino que deberían haberse tenido en cuenta otros valores y/O índices para lograr una mejor predicción.
- en la composición de los datos de entrada se utilizó una "ventana de tiempo" (time-window), la cual representó los valores de 5 días consecutivos (promedio) a la red, obteniendo como salida el valor futuro del índice del quinto día. Si consideramos a t como el tiempo de inicio de una ventana determinada, tendremos una serie del tipo $[x(t), x(t+1), \dots, x(t+n)]$ para la ventana en cuestión, pero en nuestro caso utilizamos un promedio, con lo cual se puede haber caído en un caso donde los datos utilizados no cumplan con el requisito de muestra estadística y haber provocado una mayor error en la predicción de la red, cuando debería haber sido menor.

La validación' de los datos se realizó con el mismo conjunto de datos de los patrones de entrenamiento, pero realizando la experiencia luego con un grupo de validación de datos de meses posteriores al primer trimestre de 1994, demostrando en todos los casos que las tres diferentes redes producen una reducción del error

en promedio hasta un 7%, mientras que utilizando la curva de ajuste crece hasta un 23%, demostrando nuevamente de esta manera que las redes neuronales artificiales poseen sustanciales ventajas sobre la predicción de series de tiempo utilizando técnicas tradicionales.

7. Conclusiones

Las redes neuronales artificiales evolucionan como un potente paradigma alternativo. Como lo demuestran los resultados obtenidos, sus capacidades predictivas superan ampliamente a las tradicionales técnicas de estrategia financiera. Cabe destacar que en el área financiera un pequeño incremento en la exactitud de la predicción informativa implica un significativo impacto en el entorno económico.

Como comentario adicional es importante mencionar, que los resultados obtenidos en este trabajo no son los óptimos, ya que no se explotaron la infinidad de combinaciones de parámetros posibles en la configuración de una Red Neuronal Artificial para la aplicación en consideración.

Cabe destacar que la selección de la serie de tiempo en consideración (S&P500) es arbitraria, ya que podría haberse elegido el índice Dow Jones (DJ) o utilizar el índice del mercado de la bolsa de valores de argentina (Merval). La especificación de los datos se resumió simplemente al valor del índice, no considerando otras variables fundamentales que podrían incidir en el comportamiento de la red, 'puesto que el objetivo principal se limita a demostrar las propiedades sobresalientes de las Redes Neuronales Artificiales, no incluyendo el desarrollo de un modelo optimizado para la inversión bursátil.

Dos simples enfoques fueron realizados para predecir índices financieros. Existen por supuesto numerosas otras posibilidades para la combinación de componentes, logrando una hibridización (redes neuronales con sistemas expertos, algoritmos genéticos y lógica difusa) que optimice los resultados obtenidos en este trabajo.

Bibliografía

- [1] Gately, E., *Neural Networks for Financial Forecasting*, 1996, John Wiley & Sons.
- [2] Haykin, S., *Neural Networks, a Comprehensive Foundation*, 1994, Macmillan College Publishing Company.
- [3] Thomason M., A First Multi-Network Hybrid for Financial Forecasting, 41-45, 1997, Neurove\$st Journal.
- [4] Bigus, J., *Data Mining with Neural Networks*, 1996, McGraw-Hill.
- [5] Martínez O., Procesos Bursátiles Markovianos Discretos, 292-305, 25 Jornadas Argentinas de Informática e investigación Operativa, 1996.

AN OVERVIEW OF AN AUTONOMOUS INTELLIGENT SYSTEM WITH AN EMBEDDED MACHINE LEARNING MECHANISM

Ramón García Martínez (1, 2)

1- Buenos Aires Institute of Technology

2.- Intelligent Systems Laboratory. Computer Science Department. School of Engineering. University of Buenos Aires

Address: Bynón 1605. Adrogué (1846). Buenos Aires. ARGENTINA'
e-mail: rgm@mara.fi.uba.ar, rgm@tba.edu.ar

ABSTRACT

An overview of the architecture of the system is given, the system sensor sub-system and the process of building local theories is defined, the planner that the system uses% build plans that permit to it to reach the self-proposed objectives is introduced and the ponderator that permits to obtain a measure from how good or how bad is the generated plan is given, finally some results of experimentation are described.

19 INTRODUCTION

Machine Learning [Carbonell et al., 1983; 1986; Kodratoff & Carbonell, 1990] is a field of Artificial Intelligence which has taken impulse in the last years. Among the different types of Machine Learning techniques, those based on observation and discovery are the best modelizers for the human behavior.

From this point of view, it is interesting to study the ways in which an autonomous system [Fritz, 1984; Fritz et al., 1989; García Martínez, 1990] can build automatically theories which "modelices its environment. These theories can be used by the system to improve its behavior in response to changes in its environment.

Theory formation by heuristic mutation of observations [García Martínez, 1992; 1993a; 1993c] have been proposed as a type of Machine Learning technique based on observation and active experimentation for autonomous intelligent systems [García, Martínez, 1993b; 1993d].

In this paper is proposed the integrated architecture of a system that with elemental components shows an autonomous intelligent behavior. The presented architecture is guided to try of solving the machine learning problem of the model of the environment (this is: what are the consequences of applying determined actions to the environment) departing of no previous knowledge. To accelerate the convergence of the learned model is proposed the utilization of heuristics and to handle the contradictions among the generated theories is proposed the utilization of estimators of probability distribution.

Considerations are accomplished on the problem of building theories; the limits of the problem are defined; a method of building theories based on the application of heuristic of

generated theories mutation is proposed, the definition of each heuristic is given, the limits of the problem of theories weighting are defined, a weighting method is proposed:-

An overview of the accomplished experimental work is presented : the experimental design, the variables, the results through graphics are shown and an interpretation for them is provided.

In the conclusions are indicated the original contributions of the paper, are described some found limitations and are outlined future research work.

2. GENERAL SYSTEM DESCRIPTION

2.1 ENVIRONMENT

It was selected the robot model used by Korsten, Kopeczak and Szapakowicz [1989], the one which describes the autonomous agents behavior as compared to various stages. For the description of the stages model we base on the model suggested by [Lozano-Perez and Wesley, 1979; Iyengar; and other 1985; Gil de Lamadrid and Gini, 1987; Mckendrick, 1988; Dudek and other 1991 ; Borenstein and Korent, 1991; Evans and other, 1992], those which establish to study the learning processes, planning and simulation two-dimensional stages.

The description of the environment can be simulated in a counterfoil in the one which each element represents a portion of the space, the one which can be: an obstacle, an energy point or a point of free-to-pass, space by the robot.

2.2 ARCHITECTURE

The system can be described as an exploring robot that perceives the environment. AS the situation in the one which is found the system attempts to determine an actions sequence that permit to it to reach a nearby objective, the one which calls plan. This sequence is presented to the evaluator of Plans, who determines its acceptability. The plans controller in execution is entrusted with verifying that the plan will be fulfilled successfully.

All movement of the robot is accompanied of the description of its environment, the conjunction of the action applied to such description and obtained the resulting situation make to the apprenticeship of the system. If the knowledge involved in this apprenticeship already it was learnt it is reinforced, in other case is incorporated and are generated mutant theories.

2.3 SENSING SYSTEM

The model of the sensing system was extracted and modified from the one proposed by [Mahadevan and Connell 1992], who suggest a system of 24 sectors, distributed in three levels, Garcia Martínez [1992c] suggested that the model had to present 8 Sectors

constituted in two levels and distributed in three regions: a Lateral Left, a Frontland and a Lateral Straight. The Frontal region is found shared in vertical form in two subregions.

As previously we mention each region possesses two scope levels, a level of sensing nearby and a level of sensing distant. The sensing system possesses eight sectors, each sector is corresponded with a binary representation in a string of bits shown in fig.1 describes the perception of a situation.

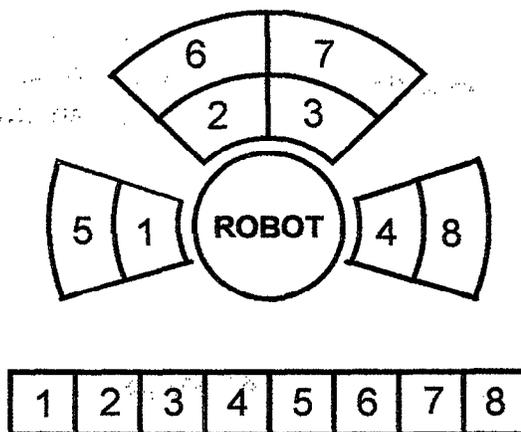


Figure 1. Correspondence between bits and occupied regions

3. THEORIES FORMATION

The structure model for the theories training was developed by García Martínez (1991a, 1991 b, 1992a, 1992b, 1992c). The same is an extension of the model of Fritz [Fritz and other, 1989), in the one which an experience unit was constituted by:

$$-^{\wedge}, \quad [\text{Initial Situation, Acción, Final Situation}].$$

García Martínez proposes the to aggregate coefficients that permit to determine the acceptability of a theory. The proposed experience model possesses the following structure:

$$Ti = [SI,, A_{ii}, SF,, P, K, U]$$

Theory:	'Ti
	SI, Initial Situation
	A Action
	SF _i Final Situation
	P Quantity of times that the theory Ti conclude successfully (the waited final situation is obtained).
	K Quantity of times that the theory Ti was used
	U Usefulness level reached applying the action the initial situation of the theory .

We have assumed to calculate the usefulness of the theories, the one which reflects an estimate measure of the distance of the robot to the nearest energy point.

Given the Initial Situation perceived by the autonomous intelligent System, in an instant of the time (let it be T_1) applying an action, we arrive at the Final Situation (in a space of the time T_2). Taking into account the presented theory model can say that given a theory T_i , the supposed conditions of the theory T_i , the action, the predicted effects of the theory, the quantity of times that the theory is applied with success P , the number of times that the theory is applied K , being S the situation to the one is applied and S' the situation to the one is arrived (see example in Fig. 2).

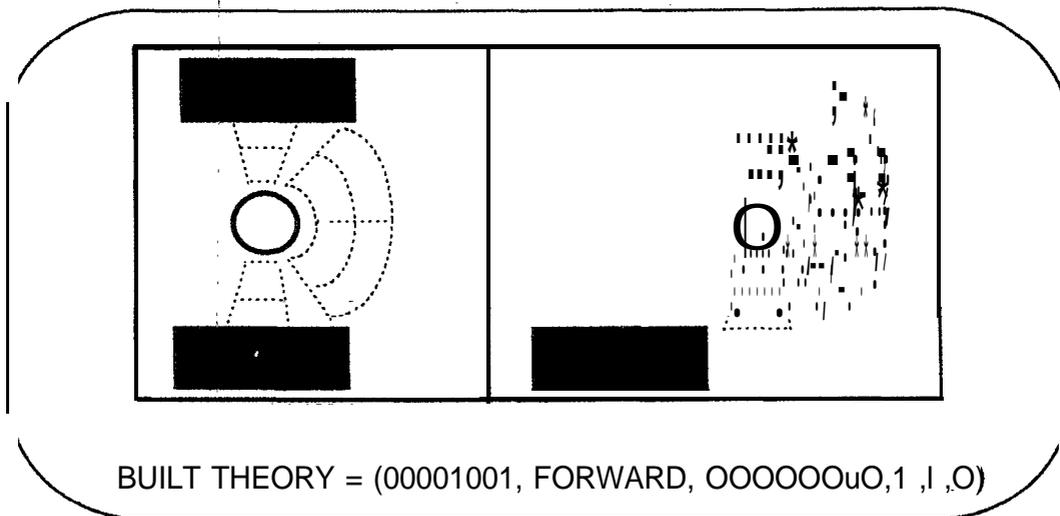


Figure 2. The situations, an action (**FORWARD**) and the corresponding built theory

4. PLANNING

The planner is mechanism through which the system build a plan (actions succession) that allows to it to reach its objectives in an efficient form. Each time the system is found in a particular situation, tries to build a plan that applied to the current situation permits to the system to reach a desirable situation. If this plan exists the desirable situation is converted into an objective situation. Eventually could occur that if it did not exist a plan, for something which one must to select other situation as desirable, if no one of the desirable situations can be reached by a plan, then is executed some of the plans by contingency.

A situation is known if it is registered as a situation of a theory of the set of theories. The desirable situations stack constituted by the predicted effects of the theories that possess greater usefulness level, ordered decreasingly usefulness level. Each situation -can be interpreted as the resulting of have applied an action to an anticipated situation. This determines the precedence among the different situations, as of those which will be armed the graph of situations

Once the planner armed the graph is proceeded to find the plan among the desirable situation and the current situation. If the plan is not found, there is taken other situation of

the desirable situations stack, and so on until there do not exist more desirable Situations or is found a plan. If the stack is emptied and no plan was obtained, there are generated plans by contingency. If at least a plan was found, is weighted to obtain its acceptability.

5. PONDERATION OF PLANS

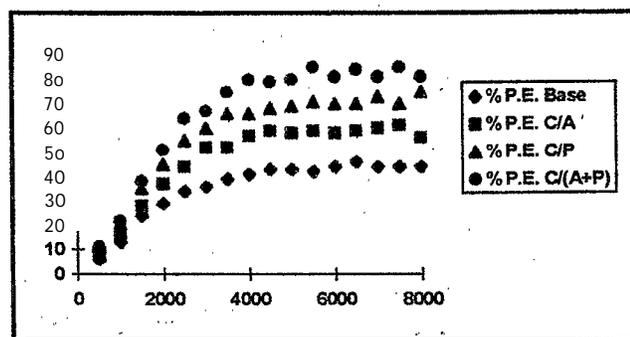
The knowledge that the system possesses as the set of theories at a given time can be viewed as a model of how its environment will react to to the system's actions. The quotient P/K of a given theory, can be asumed 'as an estimator of the probability [Calistrí-Yeh, 1990] of the fact that given the the action A_k applied to a situation S_i gives ,the resulting situation S_j ..

The situation S_j verifies the predicted effects of' the considerate theory. Therefore the knowledge that the system has for a given instant can be thought as the transition matrix M_k of the adión A_k , that has on the posición (i, j) the quotient P/K of the theory that possesses the supposed conditions S_i , the predicted effects S_j and the action A_k .

6. EXPERIMENTS

The autonomous system that was simulated answered. successfully to the experiments in which it was ran. In the environment the system demonstrated capacity to understand and express intelligent behaviors while records experience (theories) of the environment in which it was proven. For the presentation of the results obtained in the experiment this paper have been guided by the structure proposed by Matheus [1990, 1990b]: The results of experimentation can be shown in the following graphic:

Axis Y: Average of Successful Plans
 Axis x: Time
 %P.E. BASE: Average of Successful Plans without mutation and ponderation
 %P. E. C/A: Average of Successful Plans with mutation
 %P. E. C/P: Average of Successful Plans with ponderation
 %P.E C/(A+P): Average of Successful Plans with mutation and ponderation



This results empiracally show that:

1. Theory mutación by heuristics improve the system behavior.

2. The incorporation of a weighting mechanism for planning allows to specify the plans acceptability, increasing the number of concluded plans successfully.

7. CONCLUSIONS

An overview of the architecture of a system with an embedded machine learning mechanism based on theory mutation and theory ponderation has been given, estimators that permits to obtain a measure of how good or how bad is the generated plan has been proposed and the improvement of the behavior of the system with the embedded unsupervised learning mechanism has been shown experimentally.

8. REFERENCES

- Borenstein, H. Y Koren, Y. 1991. *Histogramic in Motion Mapping for Mobile Robot's Obstacle Avoidance*. IEEE Transactions on Robotics and Automation. Volumen 7. Nber 4.
- Calistri-Yeh. *Classifying and Detecting Plan Based Mconceptions for Robust Plan Reco&tion*. Ph.D. Dissertation. Department of Computer Science. Brown UniveRsity. 1990.
- Carbonell, J., Michalski, R. y Mitchell T. 1983. *Machine Learning: The Artificial intelligence Aproabh Vo/ 1*. Morgan Kaufmann.
- Carbonell, J., Michalski, R. y Mitchell T. 1986. *Machine Learning: The Artificial intelligence Aproach Vo/, ii*. Morgan Kaufmann.
- Dudek, G., Jenkin, M., Millos, E. Y Wilkes, D. 1991. *Robotics Expioratkk as graph Obstayie Avoidance*. IEEE transactions on Robotics and Automation Vol. 7 N°6. pp. 859-865. EEUU.
- Evans, J., Krishnamurty, B., Barrows, B., Skewis, T. Y Lumelsky, V. 1992. *Handiing Reai World, Motion Planning : A Hospital Transport Robot*. IEEE Control Systems. Volumen 12. N 1.
- Fritz, W. *The intelligent System*. 1984. ACM SIGART Newsletter. Nber 90. October.
- Fritz, W., García Martínez, R., Blanqué, J., Rama, A., Adobbati, R. y Sarno, M. 1989. *The Autonomous intelligent System*. Robotics and Autonomous Systems. Vol. 5 Nber. 2. pp. 109-125. Elsevier.
- García Martínez, R. 1990. *Un Algoritmo de Generación de Estrategias para Sistemas inteligentes Autonomos*. Proceedings II Iberoamerican Congress on Artificial intelligence. pp. 669-674. LIMUSA. México.
- García Martínez, R. 1992. *Aprendizaje Basado en Formación de Teorías sobre el Efecto de las Aycjones en el Entorno*. Master Thesis. Artificial Intelligence Department. School of Computer Science. Politechnic University of Madrid.
- García Martínez. IQ93a. *Aprendizaje Automatjco basado en Método Heurístico de Formación y Ponderación de Teorías*. Tecnología. Vol.15. pp. 159-182. B r a r s i .
- García Martínez, R. / 1993b. *Heuristic theory formation as a machine learning method* Proc edings VI International Symposium on Artificial Inteligente. pp 294-298. IMUSA. México.
- García Martínez, R. / 1993c. *Heuristic-based theory formatjon and a solution to the theory reinforcement problem in autonomous intelligent systems*. Proceedings III

- Argentine Symposium on Artificial Intelligence. pp. 101-108. Science and Technology Secretary Press. Argentine.
- García Martínez, R. 1993d. *Measures for theory formation in autonomous intelligent systems*. Proceedings RPIC'93. pp 451-455. Tucumán University Press. Argentine.
- García Martínez, R. & Borrajo Millán, D. 1996. *Unsupervised Machine Learning Embedded, in Autonomous intelligent Systems*. Proceedings of the XIV International Conference on Applied Informatics. PAginas 71-73. Innsbruck. Austria.
- García Martínez, R. 1997. *Un Modelo de Aprendizaje por Observación en Pianificación*. Ph.D. Thesis. Artificial Intelligence Department. School of Computer Science. Politechnic University of Madrid.
- Gil De Lamadrid, J. 1987. *Obstacle avoidance Heuristic for Three Dimensional' Moving Objects*. TR 87-42. Computer Science Department. Institute of Technology. University of Minnesota.
- Gil De Lamadrid, J. Y Gini, M. 1987. *Path Tracking for a Mobile Robot among Moving Obstacle with Unknown Trajectories*. TR 87-42. Computer Science Department. Institute of Technology. University of Minnesota.
- Iyengar, S., Jorgensen, C., Rao, S. Y Weisbin, C. 1985. *Learned Navigation Path for a Robot in Unexplored terrain*. Proceedings of the Second Conference on Artificial Intelligence Applications. pp. 148-155. IEEE CS Press. Massachusetts
- Kersten, G., Koperczak, Z. Y Szapakowics, S. 1989. *Modeling Autonomous Agents in Changing Environments*. Computer Science Department TR-89-32. Universidad de Ottawa.
- Kodratoff 1. y Carbonell J. 1990. *Machine Learning: The Artificial intelligence Approach*. Vol. iii. Morgan Kaufmann.
- Lozano-Perez, T. Y Wesley, M. 1979, *An Algorithm for Planning Collision Free Paths Among Polyhedral Obstacles*. Communications of the ACM. Volumen 22 N° 10. 1979.
- Mahadevan, S. Y Connell, J. 1992. *Automatic Programming of Behavior-Based Robots using Reinforcement Learning*. Artificial Intelligence Vol 55 pp. 311-365.
- Matheus, C. 1990a. *Feature Construction: An Analytic Framework and An Application to decision Trees*. Ph.D. Tesis. Graduate College. University of Illinois. Urbana Champaign.
- Matheus, C. 1992b. *Adding Domain Knowledge to SBL through Feature Construction*. Proceedings of the Eight National Conference on Artificial Intelligence. pp. 803-808. Boston.
- Mckendrick, J. 1988. *Autonomous Knowledge-Based Navigation in Unknown Two-Dimensional Environment*. Proceedings of the first Florida Artificial Intelligence Research Symposium. pp. 59-63. University of South Florida.
- Shen, I-1. Y Signarowski, G. 1985. *A Knowledge Representation for Moving Robots*. Proceedings 2nd Congress on Artificial Intelligence Applications. pp. 621-628.

CALIDAD DE BASES DE CONOCIMIENTO UNA MEDIDA DE EVALUACION

García Martínez, R. [1, 2)

[1] Laboratorio de Sistemas Inteligentes
Departamento de Computación. Facultad de Ingeniería. Universidad de Buenos Aires.
Paseo Colón 850. 4to Piso. (1063) Capital Federal. Argentina
e-mail: rgm@mara.fi.uba.ar

[2] CAPIS. Escuela de Posgrado. Instituto Tecnológico de Buenos Aires”
e-mail: rgm@itba.edu.ar

ABSTRACT In this paper we present experimental results over de Expert System behavior and the relation between these results and the process Of knowledge base design

RESUMEN En este artículo se presentan resultados experimentales sobre el comportamiento de los Sistemas Expertos. Se relacionan los resultados obtenidos con el proceso de diseño de Bases de conocimiento:

1: INTRODUCCIÓN A LOS SISTEMAS EXPERTOS

Podemos definir los Sistemas Expertos (SE) como una clase de programas que son capaces de: aconsejar, categorizar, analizar, comunicar, consultar, diseñar, diagnosticar, explicar, explorar, formar conceptos, interpretar, justificar, planificar; son en suma, programas capaces de manejar problemas que normalmente requieren para su resolución la intervención humana especializada [Lenat el' 'Al. 1979, Rauch-Hindin, 1985; Genesereth & Nilsson, 1987; Greenwell; 1988; Debenham; 1989)

Son desarrollados con la ayuda de Expertos de Campo [García Martínez y Blanqué, 1988; Brulé & Blount, 1989], los cuales revelan información acerca de aquellos procesos mentales [Joseph, 1989; García Martínez; 1992; García Martínez 1993) que le permiten solucionar los distintos problemas, el otro profesional interviniente es el Ingeniero de Conocimiento, cuya función específica es la de dar forma simbólica y automáticamente manipulable al conocimiento proporcionado por el Experto de Campo [Lenat & Feigenbaum, 1991).

Los expertos normalmente solucionan problemas mal definidos y desestructurados, los cuales generalmente involucran diagnóstico o planificación [Rychener, 1983; Scott & Vogt, 1983; Shen & Simon, 1989), para resolverlos generalmente usan heurísticas, es decir métodos que determinan que parte de su experiencia son aplicables, estas heurísticas deben ser descubiertas por el Ingeniero de Conocimiento y programadas en el sistema experto.

2. CARACTERÍSTICAS DE LOS SISTEMAS EXPERTOS

1. Aplican su experiencia de una manera eficiente para solucionar problemas, pudiendo realizar inferencias a partir de datos incompletos o inciertos.
2. Explican y justifican lo que están haciendo.
3. Se comunican con otros expertos y adquieren nuevos conocimientos.
4. Reestructuran y reorganizan el conocimiento.
5. Pueden quebrantar reglas, es decir, interpretan simultáneamente el espíritu y la letra de las mismas.
6. Determinan cuando un problema está en el dominio de su experiencia, conocido como determinación de la relevancia del problema.

Los sistemas expertos conocidos modelan a lo sumo las tres primeras características.

3. ESTUDIO EMPÍRICO DEL COMPORTAMIENTO DE SISTEMAS EXPERTOS

3.1. UNA MEDIDA DE CALIDAD

Definimos el rango de datos como la cantidad de conceptos (aserciones) diferentes involucrados en las reglas de la base de conocimiento.

Sea Y el rango de datos y II la cantidad de reglas definimos el factor de dispersión A como el cociente entre Y y II . El factor de dispersión es una medida de la calidad de la base de conocimiento que se va generando.

Observar que si, el factor de dispersión A es muy cercano a cero, esto nos está indicando que las reglas contienen mucho conocimiento sobre los conceptos involucrados, en cambio, si el factor de dispersión D está muy alejado de cero, esto nos está indicando que las reglas contienen poco conocimiento sobre los conceptos involucrados, este comportamiento podrá ser observado en las gráficas de las secciones siguientes.

3.2. DESCRIPCIÓN DE LOS PARÁMETROS OBSERVADOS

Las estadísticas se llevaron a cabo haciendo variar y cruzando los siguientes parámetros:

Cantidad de reglas de los Sistemas Expertos:

Se modelizaron 9 tipos de S.E. distintos en función de la cantidad de reglas desde SE, de solo 10 reglas hasta S.E. de 90 reglas.

Cantidad de antecedentes por regla:

Se modelizaron 3 tipos de S.E. distintos en función de la cantidad de antecedentes por regla:

- S.E. de no mas de 3 antecedentes por regla.
- S.E. de no más de 5 antecedentes por regla.
- S.E. de no mas de 7 antecedentes por regla.

Porcentaje de Sistemas Expertos exitosos:

Indica que cantidad de Sistemas Expertos fueron exitosos para una determinada configuración de parámetros.

Cantidad de reglas disparadas:

Indica cual fue el promedio de reglas disparadas en los sistemas expertos exitosos para una determinada configuración de parámetros.

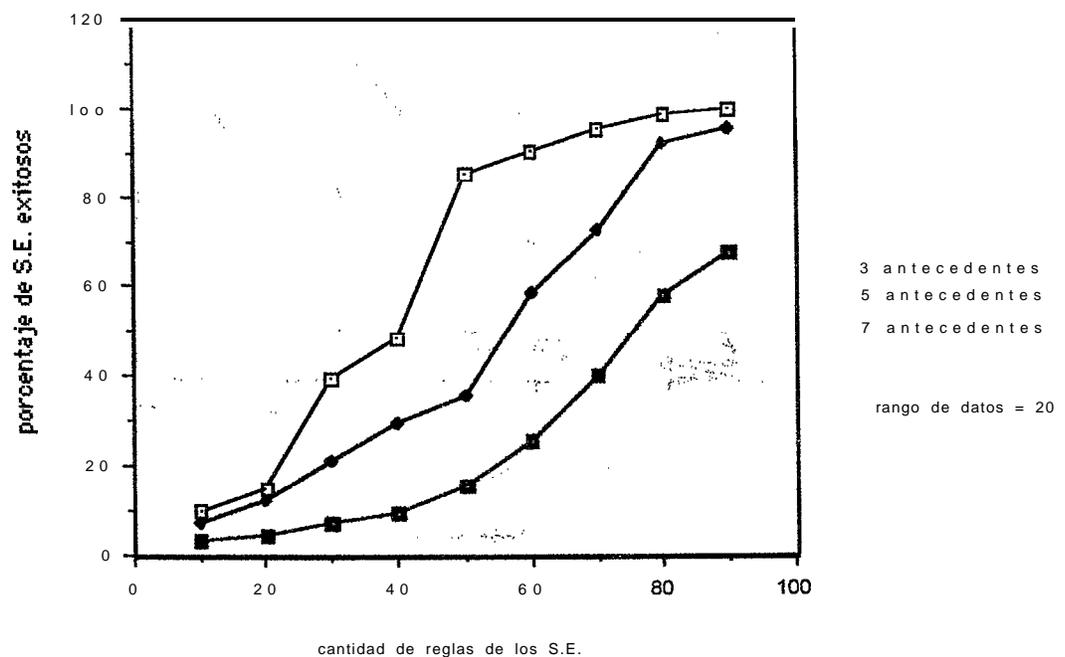
Rango de Datos:

Indica cuantos conceptos hay Involucrados en la base conocimiento. Se utilizaron 2 grupos:

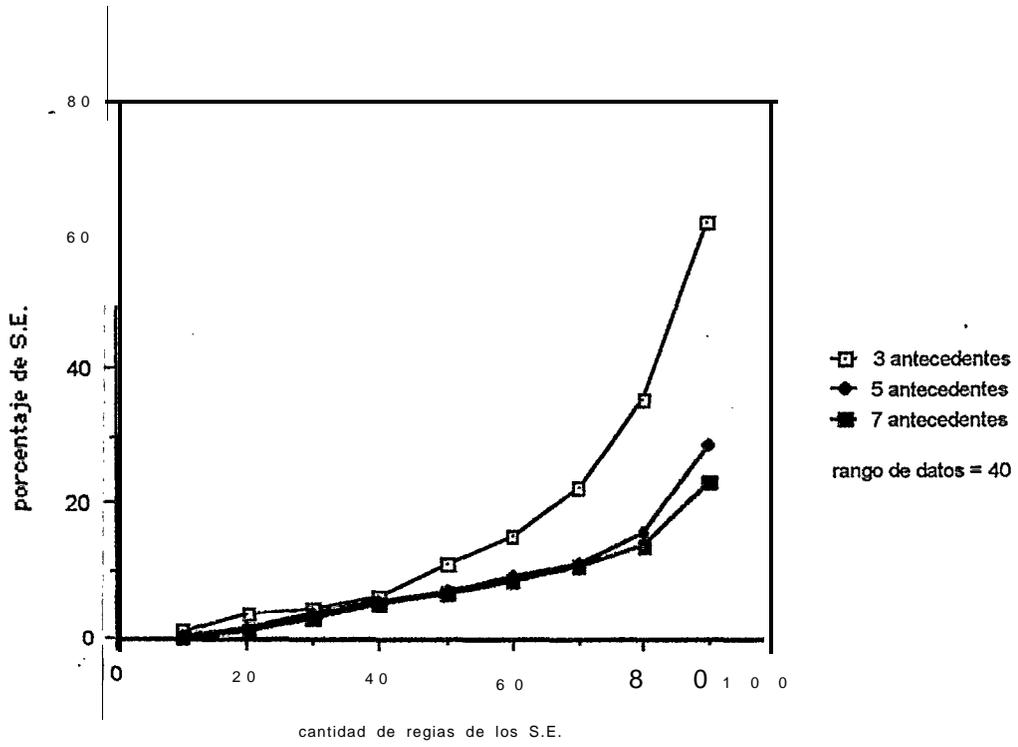
- 20 conceptos
- 40 conceptos

Nota: Para cada cruzamiento de parámetros se 'generaron en forma aleatoria mil sistemas expertos de prueba.

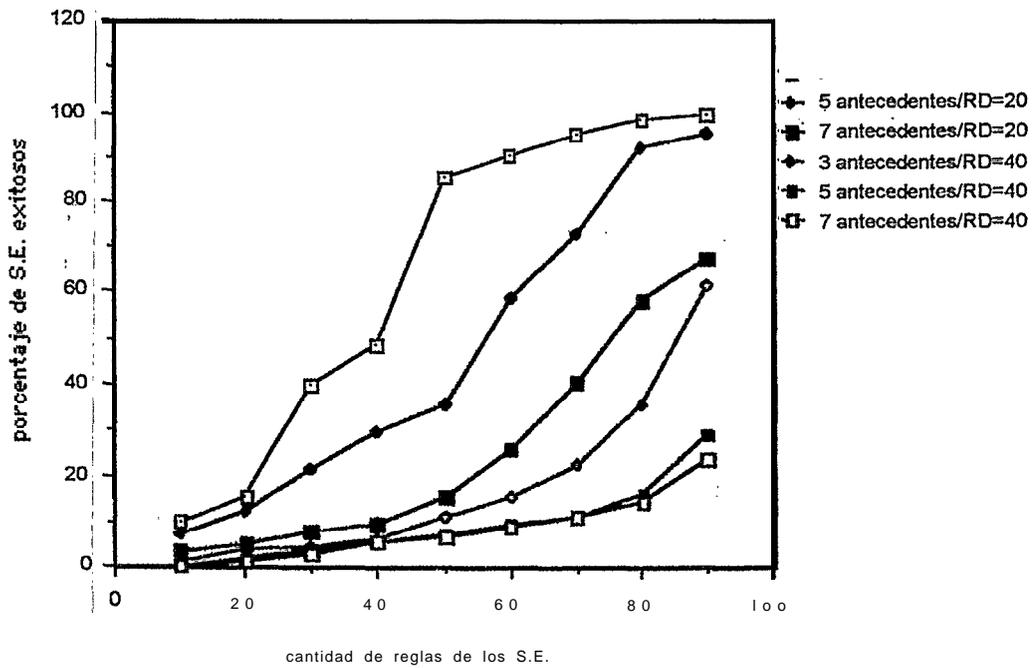
3.3. GRÁFICAS



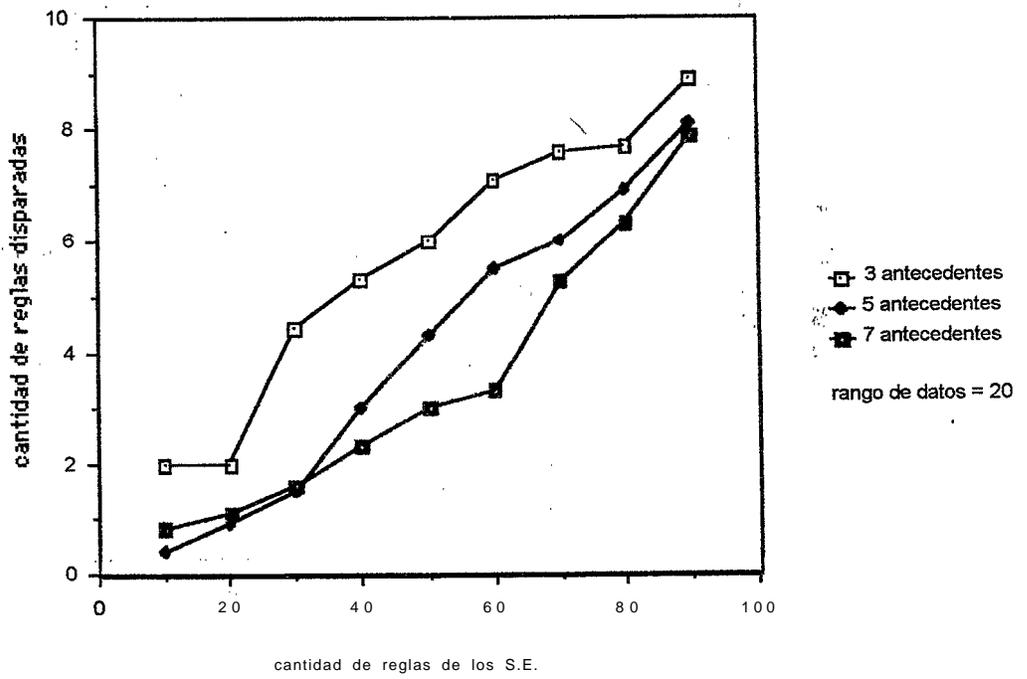
Gráfica 1



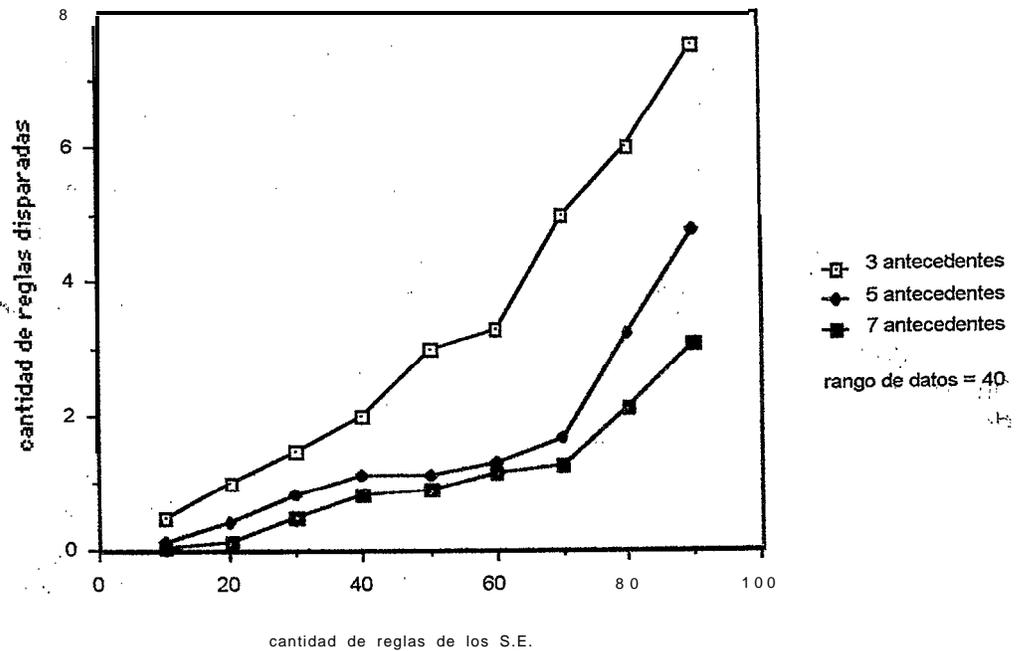
Gráfica 2



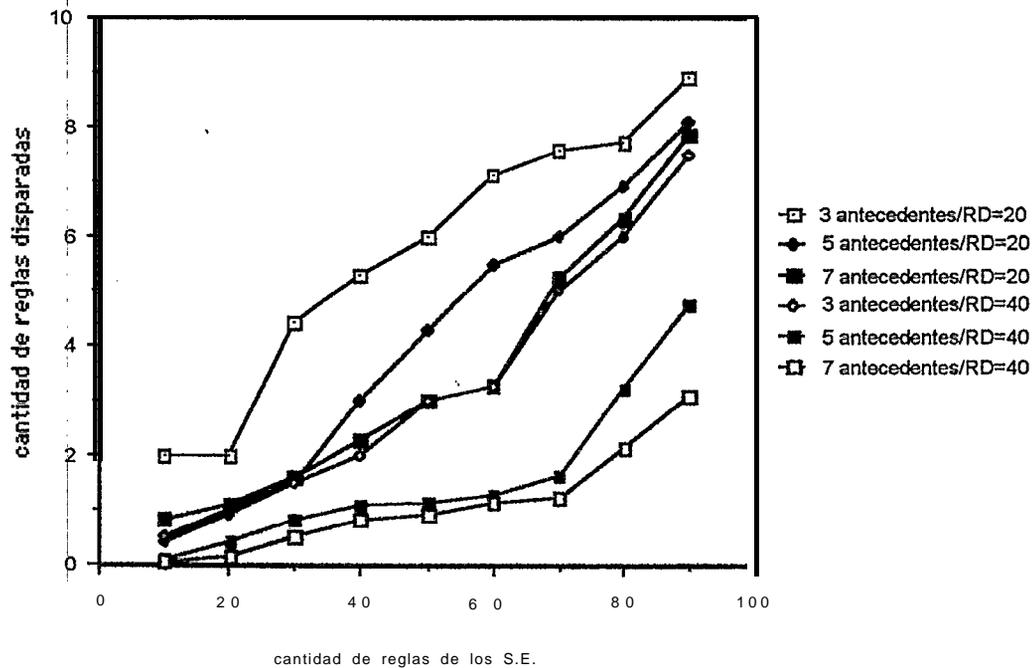
Gráfica 3



Gráfica 4



Gráfica 5



Gráfica 6

3.4. INTERPRETACIÓN

En la gráfica 1 y en la gráfica 2 podemos observar como el aumento de la cantidad de antecedentes en cada regla disminuye la posibilidad de éxito del sistema experto. Este comportamiento es razonable si pensamos que cada antecedente es una restricción más que la regla debe satisfacer para ser disparada. Esto nos lleva a postular la primer ley empírica sobre el comportamiento de los Sistemas Expertos:

LEY 1: A mayor porcentaje de reglas con gran cantidad de antecedentes a satisfacer, se hace menor la 'probabilidad' de éxito del Sistema Experto.

En la gráfica 3 podemos observar como los Sistemas Expertos cuyo factor de dispersión es más chico tienen un mejor comportamiento que aquellos cuyo factor de dispersión es mas grande. Esto se deduce del hecho que los sistemas expertos con Rango de Datos igual a 20 tienen mejor comportamiento que aquellos con Rango de Datos igual a 40. Recordando que el Factor de Dispersión esta 'definido como el cociente entre el Rango de Datos y la Cantidad de reglas llegamos a postular la segunda ley empírica sobre el comportamiento de los Sistemas Expertos:

LEY 2: A menor Factor de Dispersión, se hace mayor la 'probabilidad' de éxito del Sistema Experto.

En la gráfica 4 y en la gráfica 5 podemos observar como el aumento de la cantidad de antecedentes en cada regla disminuye el promedio de la cantidad de reglas disparadas en el sistema experto. Este comportamiento es razonable si pensamos que a mayor cantidad de antecedentes aumenta la 'probabilidad' de que la regla no sea disparada. Esto nos lleva a postular la tercer ley empírica sobre el comportamiento de los Sistemas Expertos:

LEY 3: A mayor porcentaje de reglas con gran cantidad 'de 'antecedentes' a satisfacer, se hace menor la cantidad de reglas disparadas del Sistema Experto.

En la gráfica 6 podemos hacer una observación análoga a la realizada en la gráfica 3, teniendo presente que la cantidad de reglas disparadas se corresponde con el espacio de búsqueda generado a partir de las aseveraciones y reglas, podemos postular la **cuarta** ley empírica sobre el comportamiento de los Sistemas Expertos:

LEY 4: A menor Factor de Dispersión, se hace mayor el espacio de búsqueda generado el Sistema Experto.

4. CONCLUSIONES

Podemos enunciar como conclusión los resultados obtenidos:

LEY 1: A mayor porcentaje de reglas con gran cantidad de antecedentes a satisfacer, se hace menor la 'probabilidad' de éxito del Sistema Experto.

LEY 2: A menor Factor de Dispersión, se hace mayor la 'probabilidad' de éxito del Sistema Experto.

LEY 3: A mayor porcentaje. 'de' reglas con gran 'cantidad de antecedentes a satisfacer, se hace menor la cantidad de reglas disparadas del Sistema Experto.

LEY 4: A menor Factor de Dispersión, se hace mayor el espacio de búsqueda generado el Sistema Experto.

5. BIBLIOGRAFÍA

Brulé, J. y Bount, A. *Knowledge Acquisition*. McGraw-Hill. New York. 1989.

Debenham, J. *Knowledge System Design*. Prentice Hall. Sidney. 1989.

Feigenbaum, E. y Barr, A. *Handbook of Artificial intelligence*. Vol. 1. Morgan Kaufmann. EE. UU. 1982

García Martínez, R. y Blanqué, J. *Como construir Sistemas Expertos*. Mundo Informático. Argentina. Volumen 5 Nro 163. Páginas 33-35. 1987.

García Martínez, R. y Blanqué, J. *Hacia una Metodología de Desarrollo de Sistemas Expertos*. Mundo Informático. Argentina. Volumen 6 Nro 173. Páginas 8-10. 1988.

García Martínez, R. *Construcción de Sistemas Expertos*. 154 páginas. Imprenta del CEI-UBA. Argentina. 1992.

García Martínez, R. *Detección de Ambigüedades en una Base de Conocimiento*. Anales del Primer Congreso Internacional de Informática, Computación y Teleinformática. Páginas 171-178. Editorial EXCELENCIA. Mendoza.

- Genesereth, M. y Nilsson, N. *Logical Foundations of Artificial Intelligence*. Morgan Kaufman. Los Altos. 1987.
- Greenwell, M. *Knowledge Engineering for Expert Systems*. Ellis Horwood Limited. Chichester. 1988.
- Joseph, R. *Graphical Knowledge Acquisition*. Proceedings 4th Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff. Canada,
- Lenat, D. y Feigenbaum, E. *On the Thresholds of Knowledge*. Artificial Intelligence. Vol. 47 Pág. 185-250. EE.UU. 1991
- Lenat, D., Hayes-Roth, F. y Klahr, P. *Cognitive Economy in Artificial Intelligence Systems*. Proceedings of the Sixth International Joint Conference on Artificial Intelligence. Pág. 531-536. Tokio. 1979.
- Rauch-Hindin, W. *Artificial Intelligence in Business, Science and Industry*. Val. II. Prentice, Hall. EE.UU. 1985.
- Rychener, M. *The Instructible Production System: A Retrospective Analysis*. En *Machine Learning: The Artificial Intelligence Approach Volumen 1* editado por Carbonell J., Michalski R. y Mitchell T. Morgan Kaufmann. 1983.
- Scott, P. y Vogt, R. *Knowledge Oriented Learning*. Proceedings of 8th International Joint Conference on Artificial Intelligence. Pág. 432-435. Karlsruhe. 1983.
- Shen, W y Simon, H. *Rule Creation and Rule Learning Through Environmental Exploration*. Proceedings del Eleventh International Joint Conference on Artificial Intelligence. pág. 675-680. Morgan Kauffman 1989.

MODELOS / MODELS

TITULO:	IMPLEMENTACION COMPUTACIONAL A LAS SOLUCIONES CLASICAS DEL PROBLEMA DEL REGATEO
AUTOR:	QUINTAS, LUIS (1) ; CAVALIE, PABLO (2) ; WELCH, DANIEL (2)
INSTITUCION:	(1) DEPARTAMENTO DE MATEMATICA - IMASL (INSTITUTO DE MATEMATICA APLICADA SAN LUIS) FACULTAD DE CIENCIAS FISICO MATEMATICAS Y NATURALES. UNIVERSIDAD NACIONAL DE SAN LUIS, (2) DEPARTAMENTO DE INFORMATICA. FACULTAD DE CIENCIAS FISICO MATEMATICAS Y NATURALES. UNIVERSIDAD NACIONAL DE SAN LUIS.
DIRECCION:	(1) CHACABUCO 124 SUR - (5700) - SAN LUIS - ARGENTINA

TITULO:	ANALISIS DE REFINAMIENTOS DEL EQUILIBRIO DE NASH USANDO IMPLEMENTACIONES COMPUTACIONALES
AUTOR:	QUINTAS, LUIS (1) ; NECCO, CLAUDIA.
INSTITUCION:	IMASL - INSTITUTO DE MATEMATICA APLICADA SAN LUIS. FACULTAD DE CIENCIAS FISICO MATEMATICAS Y NATURALES. UNIVERSIDAD NACIONAL DE SAN LUIS.
DIRECCION:	(1) CHACABUCO 124 SUR - (5700) - SAN LUIS - ARGENTINA

TITULO:	SPF - UN AMBIENTE DE PROGRAMACION PARA FP
AUTOR:	QUINTAS, LUIS (1); FUNES, ANA MARIA ; KAVKA, CARLOS.
INSTITUCION:	IMASL - INSTITUTO DE MATEMATICA APLICADA SAN LUIS. FACULTAD DE CIENCIAS FISICO MATEMATICAS Y NATURALES. UNIVERSIDAD NACIONAL DE SAN LUIS.
DIRECCION:	(1) CHACABUCO 124 SUR - (5700) - SAN LUIS - ARGENTINA

TITULO:	DETERMINACION DE RUTAS OPTIMAS DE DISTRIBUCION PARA LA CIUDAD DE BAHIA BLANCA
AUTOR:	CASAL, RICARDO; CORRAL, RAFAEL; LOPEZ, NANCY; ZITO, EDGARDO; LOVOTTI, LUIS.
INSTITUCION:	DEPARTAMENTO DE INGENIERIA. UNIVERSIDAD NACIONAL DEL SUR.
DIRECCION:	AV. ALEM 1253 - (8000) - BAHIA BLANCA - BUENOS AIRES

IMPLEMENTACIÓN COMPUTACIONAL A LAS SOLUCIONES CLÁSICAS DEL PROBLEMA DEL REGATEO

Autores:

Cavalié, Pablo Andrés (*)
Welch, Daniel Alberto (*)
Quintas, Luis Guillermo ()**

(*) Departamento de Informática

(**) Departamento de Matemática - IMASL (Instituto de Matemática Aplicada San Luis)

Facultad de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
Ejército de los Andes 950 - (5700) San Luis
E-Mail: LQUINTAS(@UNSL.EDU.AR

ABSTRACT

En el siguiente trabajo se presentan algoritmos que fueron implementados para encontrar las soluciones clásicas para el Problema del Regateo. El objetivo fundamental fue facilitar el trabajo e investigadores matemáticos y economistas para comparar y analizar el comportamiento de dichas soluciones ante la presencia de diferentes conjuntos factibles.

Para ello fue necesaria el desarrollo de algoritmos que adecuaran el conjunto factible previo al cálculo de las soluciones: algoritmos de convexificación y de comprensión.

Antes de la formulación de los algoritmos se da una breve explicación teórica del área de aplicación la cual se encuadra dentro de la Teoría de Decisión, específicamente la Teoría de Regateo.

1 EL PROBLEMA DE REGATEO

La teoría de regateo surge de un artículo fundamental realizado por Nash [NA1950]. Nash se interesó en la siguiente situación: dos agentes tienen acceso a cualquiera de las alternativas: en algún conjunto factible sobre las cuales deben tomar una decisión. Si se ponen de acuerdo en una alternativa particular, eso es lo que obtienen. Si no se ponen de acuerdo, terminan en una alternativa preespecificada en el conjunto factible llamada el punto de desacuerdo. El objetivo de Nash fue desarrollar una teoría que ayudara a predecir como los agentes en situaciones de esta clase establecerían un compromiso entre sus preferencias conflictivas.

Nash especificó una cierta clase de tales situaciones conflictivas o problemas a las que les realizó su análisis e investigó sus soluciones, es decir, estableció reglas para computar para cada problema en la clase, una alternativa posible para ese problema e interpretarla como el compromiso alcanzado por los agentes. También formuló una lista de propiedades, o axiomas, que pensó que las soluciones deberían satisfacer y estableció la existencia y unicidad de una solución satisfaciendo todas las propiedades. Así entonces, Nash estableció la fundamentación de la rama de teoría de juegos conocida hoy en día como la teoría axiomática de regateo.

1.1 Dominios

Definición es la clave de pares (S,d) donde s es un subconjunto de R y d en un punto de R^n , tal que

- (1) S es convexo y compacto,

(2) $d \in S$, y

(3) existe $x \in S$ con $x > d$

S es el conjunto factible: cada punto x de S es una *alternativa factible*; la coordenada de x da los niveles de utilidad de von Neumann - Morgenstern [NM1944] obtenidos por los n agentes a través de la elección de una acción conjunta.

d es el punto de desacuerdo o status-quo.

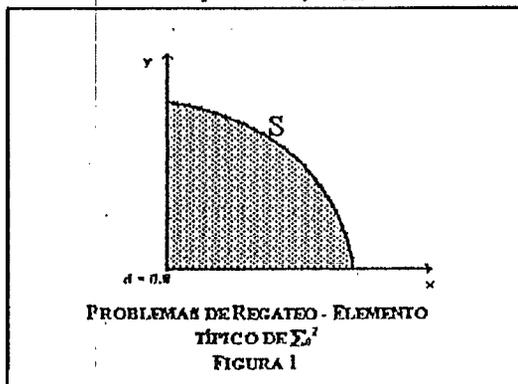
El par (s,d) es un problema de regateo o simplemente un problema

La interpretación de (S,d) es la siguiente: los n agentes pueden obtener algún punto de S si ellos acuerdan unánimemente en este. Si ellos no lo hacen obtienen d .

Se define ahora, el segundo dominio principal Σ^2 . Un ejemplo de un elemento de Σ^2 es representado en la Fig. 1. Es obtenido de $c\Sigma$ imponiendo tres restricciones adicionales. La primera es la siguiente

Definición: $(S,d) \in \Sigma_d^n$ es d -comprensivo si para todo $x, y \in \mathbb{R}^n$ con $d \leq y \leq x$ y $x \in S$, $y \in S$.

Las otras dos restricciones son que $d = 0$ y $S \subseteq \mathbb{R}_+^n$.



Para resumir, se tiene

Definición: Σ_0^n es la subclase de Σ_d^n de pares (S,d) tal que además de cumplirse las condiciones (1), (2) y (3) de la definición 1.2.1, verifican lo siguiente

(4) $d = 0$

(5) $S \subseteq \mathbb{R}_+^n$, y

(6) para todo $x \in S$ y para todo $y \in \mathbb{R}_+^n$, con $0 \leq y \leq x$, $y \in S$.

Si se cumplen (4), (5) y (6) se dice que S es 0 -comprensivo, o simplemente *comprensivo*.

Σ_0^n es una clase de problemas de particular importancia para economistas.

1.2 Soluciones

Dado un dominio de problemas de regateo, una solución definida sobre el dominio es una función que asocia con todo problema en el dominio una técnica alternativa factible de aquel problema; esta alternativa es interpretada como el compromiso alcanzado por los agentes (o recomendado a ellos, dependiendo del contexto),

El valor tomado por la solución, cuando se aplica a un problema particular, es la **solución** resultado del problema.

Sea algún dominio de problemas de n personas.

Definición: Una solución sobre S es una función $F: Z \rightarrow R^n$ tal que para todo $(S, d) \in S$, $P(x, d) \in S$

Existen esencialmente dos formas para evaluar cuán buena es una solución. Una es examinar cómo se comportan las soluciones acerca de lo que la intuición dice. La otra es formular propiedades generales, o axiomas, que las soluciones podrían o deberían satisfacer y chequear si **existen soluciones que** satisfacen todos los axiomas deseados. Si son impuestos demasiados axiomas, estos podrían ser mutuamente incompatibles. Entonces, la forma natural para proceder es buscar por la **lista maximal de axiomas** que sean compatibles. Este paso generalmente conduce a teoremas identificando una **solución particular** como la única que satisface una cierta lista de axiomas. El primero de tales teoremas de caracterización fue probado por Nash. El teorema de Nash es el modelo, después del cual, la mayoría de los trabajos subsecuentes usaron como patrón.

1.3. Principales soluciones, principales propiedades

En este capítulo, son definidas las principales soluciones dando una **caracterización axiomática**. Se presenta primero la solución de Nash. La prominencia de esta solución fue notable desde su introducción en 1950 y fue prácticamente la única hasta mediados de los 70's. En los últimos diez años, se ha producido un desarrollo considerable de nuevas soluciones. Hoy en día, dos soluciones, la de Kalai-Smorodinski [KS1975] y la Igualitaria juegan un rol igualmente importante **que** la solución de Nash. Las tres soluciones han sido generalizadas también en una variedad de direcciones.

Se analizarán primero las soluciones de Nash, Kalai-Smorodinsky e Igualitaria.

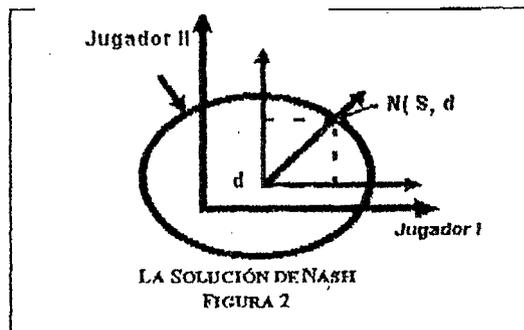
Luego se describirán una de las más recientes soluciones introducidas: la solución de Perles-Maschler (PM1981), y la más antigua: la solución Utilitaria.

1.4. La solución de Nash

Nash formuló una lista de propiedades que él intuyó **que** las soluciones deberían satisfacer y estableció la existencia y unicidad de una solución satisfaciéndola. Esta solución es ahora llamada la **solución de Nash**.

Se define ahora la solución introducida por Nash. Dado $(S, d) \in E_c^2$, $I(S, d) = \{x \in R^2 \mid x \geq d\}$,

Definición La solución de Nash N , es obtenida poniendo, para cada $(S, d) \in E_c^2$, $N(S, d) \in I(S, d)$ el máximo del producto $(x_1 - d_1)(x_2 - d_2)$ para $x \in I(S, d)$.

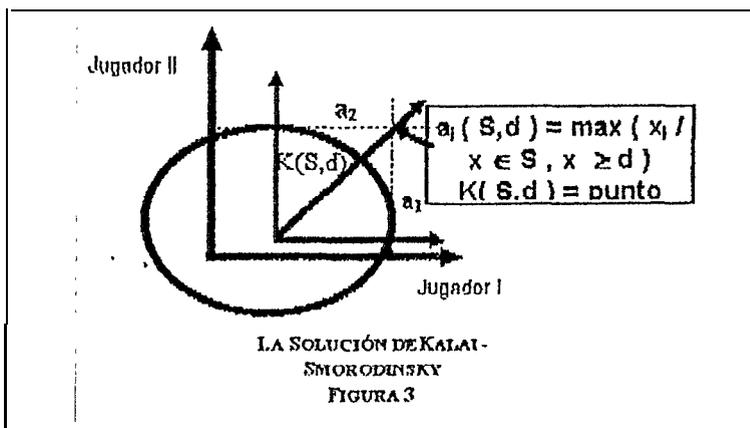


denota un producto. $(x_1 - d_1)(x_2 - d_2)$ denota el producto $(X_1 - d_1)(X_2 - d_2)$

1.5. La Solución de Kalai - Smorodinsky

En 1975, Kalai y Smorodinsky propusieron la siguiente solución

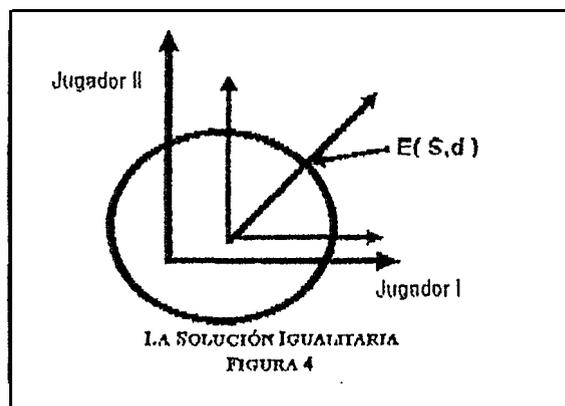
Definición: Dado $(S, d) \in C^2$, $S \subseteq \mathbb{R}^2$, $n(S, d) = \max \{x_i / x_i \geq d_i\}$ para $i = 1, 2$. Entonces la solución de Kalai - Smorodinsky, K , es definida poniendo $K(S, d)$ igual al máximo punto de S sobre el segmento que conecta d con $a(S, d)$.



1.6. La Solución Igualitaria

La solución Igualitaria selecciona el punto maximal de S en que las ganancias de utilidad de los agentes desde d son iguales.

Definición La solución Igualitaria, E , es obtenida poniendo, para cada $(S, d) \in C^2$, $R(S, d)$ a ser el punto maximal del con unto $\{x \in S / x_i - d_i = x_j - d_j\}$.



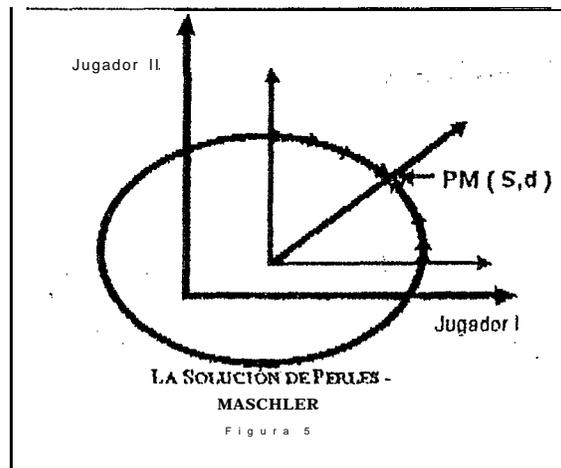
1.7 La Solución de Perles - Maschler-

En contraposición con todas las soluciones encontradas anteriormente, la solución introducida por Perles y Maschler (1981) contiene definiciones y teoremas que son de una dificultad técnica significativamente mayor comparado con los , resultados obtenidos. Sin embargo; la solución tiene un número de propiedades atractivas que hacen interesante su estudio.

No se presentará una definición formal de esta solución.

Una forma práctica para encontrar la solución de Perles Maschler es la siguiente:

Se parte de las mejores situaciones para cada uno de los jugadores, a igual velocidad y en direcciones opuestas a través de In frontera del conjunto factible. El punto de encuentro, será la solución

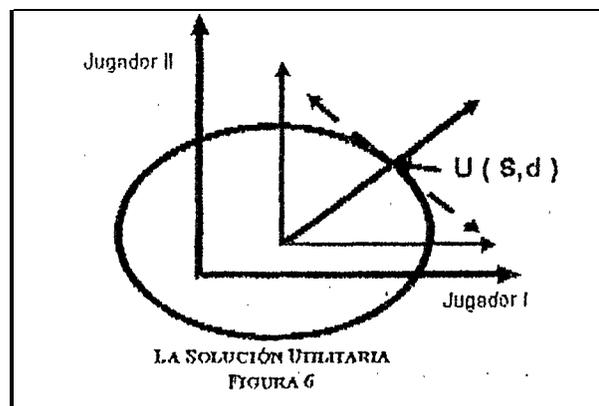


1.8 La Solución Utilitaria

Definición: La solución Utilitaria, U , es definida poniendo, para caso $S \in Cd$, $U(S,d)$ igual al máximo de $x_1 + x_2$ sobre E .

Si el máximo de $x_1 + x_2$ sobre E no es: único, el punto medio del segmento de máxima, es un mecanismo de desempate ("He-breaker") natural. La solución resultante será referida como la **solución Utilitaria U** .

La solución Utilitaria es la solución más antigua, que se remonta a escritos de la escuela **utilitaria** del siglo XIX.



2 ALGORITMOS

Los algoritmos que se presentarán para el cómputo de las soluciones clásicas, suponen como entrada un conjunto de puntos, al cual determine unívocamente la cápsula d-comprensiva del conjunto factible. Además se requiere también que el punto de desacuerdo d sea igual al punto $(0,0)$,

Por tales situaciones se dan a continuación una serie de algoritmos que permitan convertir cualquier conjunto de puntos uno con tales características.

Algoritmo de traslación al origen

Entrada: $d = (d_1, d_2)$: punto de desacuerdo
 C : conjunto de puntos a trasladar
Salida: C' : conjunto de puntos trasladados

M e t o d o :

Var $c = (c_1, c_2)$ /* variable para almacenar **un** punto cualquiera */

Q $c \in C$ hacer

$c_1 = c_1 - d_1$

$c_2 = c_2 - d_2$

Definiciones y algoritmos previos al algoritmo de d-comprensión

Definición Sea S un conjunto de puntos cualesquiera, la cápsula convexa para S será la **figura** geométrica determinada por un subconjunto mínimo C de puntos de S , $C \geq 3$, tal **que todo** punto perteneciente a S pertenece a la Figura. Se dirá que C determina la cápsula convexa para S .

Sean c_1 y c_2 puntos de S con coordenadas (x_1, y_1) y (x_2, y_2) respectivamente, tales que para todo $z = (x, y)$ perteneciente a S , $x_1 \leq x \leq x_2$ y $y_1 \leq y \leq y_2$. (Es decir, c_1 y c_2 son los puntos de S cuyas coordenadas X son la mínima y máxima respectivamente).

Definición Se denominará casco convexo de S a la **cápsula CONVEXA** de S , siendo S_c un subconjunto de S , tal $(C, \{c_1, c_2\} \cup \{z = (x, y) \mid y_1 \leq y \leq y_2, \forall x \in [x_1, x_2]\}) \cap C = \{c_1, c_2\}$.

definición: Se llamará base convexa de S a la cápsula convexa de S_B , siendo S_B un subconjunto de S , tal $q = \{c_2\} \cup \{z = (x, y) \in S \mid y_1 \leq y \leq y_2\}$ con $c_1 = (x_1, y_1)$ y $c_2 = (x_2, y_2)$.

Aunque no se va a **demonstrar**, se puede ver claramente que la unión de los puntos que determinan el casco y la **base**, S_c y S_B convexas para S respectivamente, es igual al conjunto de puntos que determinan la cápsula **convexa** de S . Es decir, $S_c \cup S_B = C$.

Por lo **tanto**, será **equivalente** encontrar S_c y S_B y luego efectuar la unión, a encontrarse el **conjunto C** que determina la cápsula convexa para S .

Algoritmo para la obtención del casco convexo

Entrada: List Ent de puntos originales

Salida: List Casco Lista de puntos que determinan el casco convexo

Método:

InS U (Elem(1, ListEnt), ListCasco) (I)

$i \leftarrow 1$

Mientras $i < \text{Car}(\text{List Ent.})$ hacer (II)

Si $\text{pend}(\text{Elem}(i, \text{ListEnt}), \text{Elem}(i+1, \text{ListEnt})) >$

$\text{pend}(\text{Elem}(i, \text{ListEnt}), \text{Elem}(i-2, \text{ListEnt}))$

entonces

InsU (Elem (i-t-1, ListEnt), ListCasco)

sino

InsU (Elem (i+2, ListEnt), ListCasco)

Mientras Car (ListCasco) > 2 **n** (III)

(pend (Ejem (Car (ListCasca), ListCasco),

Elem (Car (ListCasco)-2, ListCasco))

pend (Elem (Car (ListCasco), ListCasco),

Elem (Car (ListCasco)-1, ListCasco)))

hace

Eliminar (Car (List.Casco)-1, ListCasco)

I n c (i)

Explicación del algoritmo

Funciones utilizadas:

Insu(Punto,Lista) : Inserta el punto P unto al final de la lista Lista.

Elem(i,Usler) : Devuelve el punto ubicado en la posición i en la lista Lista,

Car(Lista) : Devuelve: la cardinalidad de la lista Lista.

Eliminar(i,Lista): Elimina el punto ubicado en la posición i en la lista Lista.

Pend(x1;y1 ,x2y2) : Devuelve la pendiente de la recta determinada por los puntos (x1 ,y1) y (x2,y2),

El primer paso del algoritmo (I) es insertar el primer punto de la lista de entrada (c1) en la lista

resultado: ListCasco.

LA primera iteración (II), consiste en recorrer **ListEnt** y **formar** ternas de puntos (A,B,C) para con ellos comparar las pendientes de los segmentos AB y AC, incluyendo B en la lista resultado si AB tiene mayor pendiente que AC, caso contrario C es insertado en la lista.

Luego, en la segunda iteración (III) se toman los tres últimos puntos de la lista resultado (A, B, C) y se comparan las pendientes de los segmentos CA y CB, eliminando B si la pendiente de CA es mayor que la de CB.

Algoritmo para la obtención de la base convexa

Este es análogo al anterior, la diferencia radica en que la comparación de pendientes se realiza por menor (<) en lugar de por mayor (>). Además, dado que el objetivo final será **encontrar el conjunto d-**comprensivo, bastará con encontrar solamente el casco convexo.

Algoritmo para la obtención de la cápsula convexa.

Entrada: *ArrPtos* arreglo conteniendo los puntos originales.

Salida: *ArrPtos* arreglo conteniendo los puntos que determinan la cápsula convexa.

Método:

Var

ListEnt, ListCasco, ListBase, ListSalida: Lista de TPuntos

PasarArreglo-Lista (ArrPtos, ListEnt)
OrdenarX(ListEnt)
CalcularCasco(ListEnt, ListCasco)
CalcularBase(ListEnt, ListBase)
UnirListas(ListCasco, ListBase, ListSalida)
PasarLista-Arreglo(ListSalida, ArrPtos)

Explicación del algoritmo

Funciones utilizadas:

PasarArreglo_Lista(Arreglo, Lista): Pasa los elementos de una estructura estática *Arreglo* a una *Lista* dinámica.

OrdenarX(Lista): Ordena la *Lista* de puntos en orden ascendente según las abscisas.

CalcularCasco(Entrada, Salida): Invoca al algoritmo para calcular el casco convexo pasándole en *Entrada* la lista de puntos originales y esperando en *Salida* el resultado.

CalcularBase(Entrada, Salida): Invoca al algoritmo para calcular la base convexa pasándole en *Entrada* la lista de puntos originales y esperando en *Salida* el resultado.

UnirListas(Entrada1, Entrada2, Salida): Concatena la lista *Entrada2* a *Entrada1* y lo devuelve en la lista *Salida*.

PasarLista_Arreglo(Lista, Arreglo): Pasa los elementos de una estructura dinámica *Lista* a un *Arreglo*.

Este algoritmo no hace desarrolla ninguna tarea nueva, sino solamente se encarga de organizar la invocación de otros algoritmos.

Las invocaciones a *CalcularBase* y *UnirListas* no serán necesarias si el objetivo es encontrar la cápsula d-comprensiva. Simplemente se podrán saltar y en la última sentencia deberá reemplazarse el parámetro *ListSalida* por *ListCasco*.

Algoritmo de d-comprensión

Entrada: C: Arreglo de puntos.

Salida: C modificado a ser d-comprensivo.

Método:

Var

MaxX, MaxY: Reales

$MaxX \leftarrow (x / \forall (x_i, y_i) \in C \ x \geq x_i \wedge (x, y_i) \in C) /* \text{valor máximo que toma } x */$

$MaxY \leftarrow (y / \forall (x_i, y_i) \in C \ y \geq y_i \wedge (x_i, y) \in C) /* \text{valor máximo que toma } y */$

Agregar a C los siguientes puntos

(MaxX, 0)

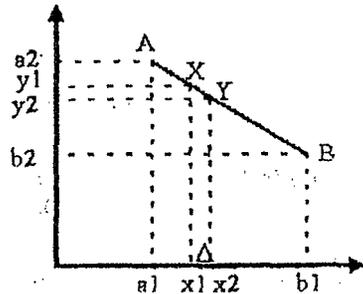
(0, MaxY)

(0, 0)

Invocar el algoritmo para obtención de la cápsula convexa con C como su entrada

Ecuación para encontrar los puntos comprendidos en un segmento

Varios de los algoritmos para el cálculo de las soluciones necesitarán recorrer los puntos de un segmento determinado por dos puntos conocidos. A partir de una discretización del problema se da a continuación la ecuación que permite encontrar las coordenadas del siguiente punto buscado en un segmento dados el punto anterior, los dos puntos que determinan el segmento y, la variación de la x (Δ), la cual responde a la discretización elegida



$$\Delta = x_2 - x_1$$

Sea $A = (a_1, a_2)$, $B = (b_1, b_2)$ y p la pendiente del segmento \overline{AB} igual a $\frac{b_2 - a_2}{b_1 - a_1}$ y sea $X = (x_1, y_1)$.

Se desea encontrar el punto $Y = (x_2, y_2)$ tal que Y pertenezca al segmento \overline{AB} y $x_2 = x_1 + \Delta$

Sea la ecuación de una recta igual a

$$y = px + d$$

dónde d es el término independiente igual a

$$d = -pa_1 + a_2$$

Reemplazando

$$y = px - pa_1 + a_2$$

Sacando factor común la pendiente queda

$$y = p(x - a_1) + a_2$$

como $x_2 = x_1 + \Delta$ y la pendiente del segmento \overline{AB} es igual a la del segmento \overline{XB} : p

$$y_2 = p\Delta + y_1$$

Algoritmo para el cálculo de la solución de Nash

Entrada: Puntos: Estructura que contiene los puntos que forman el conjunto factible d -compleensivo ordenados ascendentemente según las x .

CantPuntos: Variable entera indicando la cantidad de puntos en Puntos.

A: Número real que indica la variación de las abscisas.

Salida* SolNsw: Estructura de tipo TPunto conteniendo la solución de Nash.

Método:

Var

i : Integer

p : Real

Aux: TPunto

SolNash \leftarrow Puntos[1]

Aux \leftarrow Puntos[1]

$i \leftarrow 1$

Mientras $i < \text{CantPuntos}$ hacer (I)

```


$$p \leftarrow \frac{\text{Puntos}[i+1].y - \text{Puntos}[i].y}{\text{Puntos}[i+1].x - \text{Puntos}[i].x}$$

Mientras Aux  $\neq$  Puntos[i+1] hacer (II)
  Aux.x  $\leftarrow$  Aux.x +  $\Delta$ 
  Aux.y  $\leftarrow$  p $\Delta$  + Aux.y
Si Aux.x * Aux.y > SolNash.y * SolNash.x
  SolNash  $\leftarrow$  Aux
Inc(i)

```

Explicación del algoritmo

La idea del algoritmo es: recorre la frontera superior del conjunto Factible buscando el **punto** [X,Y) tal que x*y aca el máximo (rectángulo de área máxima comprendido en el conjunto factible).

La primera iteración (I) recorre los puntos que determinan la cápsula convexa, **los** cuales están almacenados en la estructura **Puntos**. La segunda iteración (II), recorre los puntos comprendidos en el segmento determinado por un par de puntos adyacentes en Puntos; como estos son infinitos el algoritmo recibirá en la variable A la variación que se desee realizar a las x's.

Si bien el pseudocódigo de este algoritmo recorre toda la frontera, se puede probar que por d-comprensividad del conjunto, el producto de las coordenadas de los puntos se va incrementando a través de la frontera hasta que alcanza el producto maximo, y luego decrece; por lo que el algoritmo podría tener un punto de corte ni bien el producto comience a decrecer.

Algoritmo para el cálculo de la solución de Kalai - Smorodinsky

Entrada: Puntos: Estructura que contiene los puntos que forman el conjunto factible ordenados ascendentemete según las x.

Cant. Puntos: variable entera indicando la cantidad de puntos en Puntos.

A: Número real que indica la variación de las abscisas.

Sslida: SolKS: Estructura de tipo **TPunto** conteniendo la solución de Kalai - Smorodinski

Método:

```

Var
  i, MX, MY: Integer
  p, pi, q: Real
  Aux: TPunto
SolKS  $\leftarrow$  Puntos[1]
Aux  $\leftarrow$  Puntos[1]
i  $\leftarrow$  1
MX  $\leftarrow$  MaxX(Puntos)
MY  $\leftarrow$  MaxY(Puntos)
 $p \leftarrow \frac{MY}{MX}$ 
Mientras i < CantPuntos hacer
   $q \leftarrow \frac{\text{Puntos}[i+1].y - \text{Puntos}[i].y}{\text{Puntos}[i+1].x - \text{Puntos}[i].x}$ 
  Mientras Aux  $\neq$  Puntos[i+1] hacer
    Aux.x  $\leftarrow$  Aux.x +  $\Delta$ 
    Aux.y  $\leftarrow$  q  $\Delta$  + Aux.y

```

$$p_i \leftarrow \frac{Aux.y}{Aux.x}$$

si $|p_i - p| < \left| \frac{SolKS.y}{SolKS.x} - p \right|$ entonces

SolKS \leftarrow Aux

Inc(i)

Explicación del algoritmo

Primero encuentra las rectas en donde x e y toman sus valores máximos MX y MY. El punto (MX,MY) y el punto (0,0) determinan una recta de pendiente $\frac{M-Y}{M-X}$

Luego recorre la frontera de forma similar al algoritmo para calcular Nash hasta encontrar el **punto** cuya pendiente sea lo mas cercana p.

En realidad, la pendiente de la solución debería ser igual a p. pero como la variación de las x's se realiza en intervalos A, escoge el punto aproximado.

Algoritmo para el cálculo de la solución Igualitaria

Entrada: Puntos: Estructura que contiene los puntos que forman el conjunto factible ordenador ascendentemente según les x.

Cant.Puntos; Variable entera indicando la cantidad de puntos en Puntos.

A; Número real que indica la variación de las abscisas.

Salida SolEgallt: Estructura de tipo TPunto conteniendo la solución Igualitaria

Método:

Var

i : Integer
 p,p,q: Real
 Aux: TPunto

SolEgalit \leftarrow Puntos[1]
 Aux \leftarrow Puntos[1]
 i \leftarrow 1
 p \leftarrow 1 /* pendiente de la recta $x = y$ */
 Mientras i \leq CantPuntos hacer

$$q \leftarrow \frac{Puntos[i+1].y - Puntos[i].y}{Puntos[i+1].x - Puntos[i].x}$$

Mientras Aux.Puntos[i+1] hacer

Aux.x \leftarrow Aux.x + A
 Aux.y \leftarrow q Δ + Aux.y

$$p_i \leftarrow \frac{Aux.y}{Aux.x}$$

si $|p_i - p| < \left| \frac{SolEgalit.y}{SolEgalit.x} - p \right|$ entonces

SolEgalit. \leftarrow Aux

Inc(i)

Explicación del algoritmo

En forma analoga al algoritmo para calcular la solución de Kalai -Smorodinsky, recorre la Frontera buscando el punto que con el origen determinen la recta de pendiente mas cercana a 1. **Es** decir el punto intersección de la frontera del conjunto con la recta $y = x$.

Algoritmo para calcular la solución de Perles- Maschker

Entrada: Puntos; Estructura que contiene los puntos que forman el conjunto factible ordenados ascendentemente según las x

CantPuntos: Variable entera indicando la cantidad de puntos en Puntos.

Salida: SolPM: Estructura de tipo TPunto conteniendo la solución de Perles - Machler

Método:

Var

d, d_1, d_2, d_1 : Real
 i, n : Integer

$$d_1 \leftarrow \sum_{i=1}^{\text{CantPuntos}-1} \sqrt{(Puntos[i].x - Puntos[i+1].x)^2 + (Puntos[i].y - Puntos[i+1].y)^2} \quad (I)$$

$n - - 2$

Repetir (II)

$$d \leftarrow \sum_{i=1}^n \sqrt{(Puntos[i].x - Puntos[i+1].x)^2 + (Puntos[i].y - Puntos[i+1].y)^2}$$

Inc(n)

Hasta $\frac{d}{2}$

Si $d = \frac{d}{2}$ - entonces T (III)

SolPM - Puntos[n-1]

sino

$$(IV) \quad d_1 \leftarrow \frac{dr}{2} - \left(d - \sqrt{(Puntos[i].x - Puntos[i+1].x)^2 + (Puntos[i].y - Puntos[i+1].y)^2} \right)$$

$$(V) \quad d_2 \leftarrow \sqrt{(Puntos[i].x - Puntos[i+1].x)^2 + (Puntos[i].y - Puntos[i+1].y)^2}$$

$$(VI) \quad \text{SolPM}.x \leftarrow Puntos[n-2].x + \left(\frac{Puntos[n-1].x - Puntos[n-2].x}{d_2} \right) d_1$$

$$(VII) \quad \text{SolPM}.y \leftarrow Puntos[n-2].y - \left(\frac{Puntos[n-1].y - Puntos[n-2].y}{d_2} \right) d_1$$

Explicación del algoritmo

La idea del algoritmo es encontrar el punto de la frontera del conjunto, cuya distancia a través de la frontera desde la mejor opción de alguno de los jugadores sea igual a la mitad de la distancia de la curva comprendida entre las mejores opciones de ambos jugadores.

El primer paso del algoritmo es calcular entonces la distancia total de la frontera: DT (Y)

Luego en la iteración (II) recorre la estructura Puntos acumulando las distancias para determinar el punto o el segmento en donde se encuentra la solución

Si encuentra un punto de Puntos cuya distancia es DT/d (III) lo retorna como la solución de Perles-Mascler; caso contrario encuentra un par de puntos adyacentes en Puntos, $A = (a1,a2)$ y $B(b1,b2.)$ que determinan el segmento en el que se haya la solución.

Sea $S = (s1,s2)$ el punto solución a encontrar y sea $O = (o1,o2)$ el punto que representa la mejor opción para el jugador Y.

Entonces $d(O,S)$ debe ser igual a $D/2$, $d(O,A)$ se puede calcular, y la distancia de A a S será $d1 = -DT/2 - d(O,A)$ (IV)

$d2$ es la distancia entre A y B (V)

Con esos datos, $S = (s1,s2)$ se puede calcular con las siguientes ecuaciones

$$s1 = a1 + \frac{b1 - a1}{d2} d1 \text{ (VI)}$$

$$s2 = b1 - \frac{b1 - b2}{d2} d1 \text{ (VII)}$$

Algoritmo para calcular la solución Utilitaria

Entrada: Puntos: Estructura que contiene los puntos que forman el conjunto factible ordenados ascendentemente según las x.

Cant.Puntos: Variable entera indicando la cantidad de puntos en Puntos.

A Número real que indica la variación de las abscisas,

Salida: SolUtilIt Estructura de tipo TPunto conteniendo la solución Utilitaria

Método.-

Var

ListaCandidatos: Lista de TPunto

MaxSuma, i: Integer

p: Real

Aux: TPunto

MaxSuma = 0

i = 1

Mientras i < CantPuntos Hacer

$$p \leftarrow \frac{\text{Puntos}[i+1].y - \text{Puntos}[i].y}{\text{Puntos}[i+1].x - \text{Puntos}[i].x}$$

Aux ← Puntos[i]

Mientras Aux ≠ Puntos[i+1] Hacer

Aux.x ← Aux.x + Δ

Aux.y ← p Δ + Aux.y

Case Aux.x + Aux.y > MaxSuma (1)

= : InsertU(ListaCandidatos,Aux)

> : Vaciar(ListaCandidatos)

InsertU(ListaCandidatos,Aux)

MaxSuma ← Aux.x + Aux.y

< : Descartar

EndCase

solUtilit - MitadLista(listaCandidatos) (II)

Explicación del algoritmo

Recorre la frontera del conjunto factible de manera análoga a como lo hace el algoritmo para **calcular** Nash, pero en lugar de escoger el punto de mayor producto, elige el de mayor suma.

En este puede darse que exista más de un punto que tengan la suma mayor, en **cuyo caso estos serán** adyacentes y el algoritmo o en (I) los va almacenando en una lista de soluciones candidatas.

Al finalizar (II) escoge el punto de la mitad de la lista de candidatos que equivale al punto medio del segmento, o en caso de tener **solución** única al único punto que estará almacenado en dicha lista.

3 CONCLUSIONES Y POSIBLES EXTENSIONES

En la actualidad, la Teoría de Juegos está en su apogeo, Una gran cantidad de investigadores de todo el mundo, de ramas de matemática y economía están abocados en esa temática.

Nuestro objetivo fue aprovechar las ventajas de la informática para brindar apoyo tanto a investigadores, como a docentes y alumnos, en el área específica de Teoría de Juegos denominada Teoría **de** Regateo:

Gracias a este trabajo, estas personas podrán fácilmente, comparar y analizar el comportamiento de las soluciones clásicas al problema.

Quedan como futuras extensiones las siguientes tareas:

- Además de las soluciones clásicas, existen otras no tan conocidas **que se** podrían llegar a estudiar como por ejemplo: Solución **Pago** a Paso (Step by Step **Solution**), **Soluciones** Integrales (Integral Solutions), Soluciones de Ordenamiento (Ordering Solutions)
- Variante de las soluciones **clásicas**.

4 BIBLIOGRAFIA

- [NA1950] Nash, J. F. (1950) "The Bargaining Problem". *Econometrica*, 28:155-162.
- [PM1981] Perles, M. A. and M. Maschler (1981) "A Super-Additive Solution for the Nash Bargaining Game". *International Journal of Game Theory*, 10:163-193.
- [KS1975] Kalai, E. and M. Smorodinsky (1975) "Other Solutions to Nash's Bargaining Problem". *Econometrica*, 43:513-518.
- [NM1944] von Neumann, J. and O. Morgenstern (1944) "Theory of Games and Economic Behavior". Princeton: Princeton University Press, Second Ed., 1947.
- [TW1994] Thompson, W. (1994) "Cooperative Models of Bargaining". *Handbook of Game Theory with Economic Applications*, Vol. 2, North Holland.
- [TW1986] Thompson, W. (1986) "Bargaining Theory, the Axiomatic Approach". University of Rochester, Rochester N.Y. 14627.
- [WC1982] Wieder, C. M. (1982) "Elementos de Matemáticas Prácticas". Ed. Tesis.
- [HA1986] Howard Anton (1986) "Introducción al álgebra lineal". Ed. Limusa, novena edición.

ANALISIS DE REFINAMIENTOS DEL EQUILIBRIO DE NASH USANDO IMPLEMENTACIONES COMPUTACIONALES

Claudia Necco y Luis Quintas

Dirección: Instituto de Matemática Aplicada San Luis- Facultad de Ciencias Físico-Matemáticas y Naturales - Universidad Nacional de San Luis - Ejercito de los Andes 950 (5700) - San Luis - ARGENTINA

Fax: 0652-30224

Telex: 58125 UNSL-AR

Correo Electrónico: ayesha@unsl.edu.ar

Teléfono: 0652-20803

Tópicos: Teoría de Juegos - Automatas - Optimización y Simulación

Palabras claves: Juegos Repetidos-Automatas - Equilibrios - Estrategias Dominadas

Resumen

En este trabajo se realiza un análisis comparativo de distintas situaciones conflicto, que se modelan como juegos repetidos, bajo supuestos de racionalidad acotada y usando implementaciones computacionales de refinamientos del Equilibrio de Nash. Para implementar las estrategias de cada jugador se usan autómatas de tamaño prefijado. Los equilibrios analizados fueron el Subjuego Perfecto y el Forgiving. Se usan criterios de dominación para eliminación de estrategias. El pasaje del Equilibrio de Nash o Subjuego Perfecto a Forgiving resulta crucial para la eliminación de equilibrios no intuitivos.

1 .- INTRODUCCIÓN

Cuando agentes económicos se hallan involucrados en relaciones a largo plazo Su comportamiento difiere substancialmente de aquel que se observa cuando solo se trata de una situación de corto plazo. Estas situaciones se modelan como Juegos Repetidos. un tipico ejemplo lo constituyen firmas que interactúan en un mercado. Dichas firmas suele presentar conductas "cooperativas" aún cuando no haya ningún compromiso establecido entre ellas para llevar a cabo tal cooperación. La posibilidad! de futuras acciones de castigo para conductas demasiado agresivas permiten establecer cooperación sobre bases estrictamente no-cooperativas (esto es bajo la suposición que cada firma soto intenta maximizar sus beneficios).

En este modelo se estudia la posibilidad de que existan errores. Esto se acerca mas a situaciones reales donde efectivamente existe la posibilidad de que tales errores ocurran .

El sistema desarrollado provee una herramienta útil para estudios empíricos de situaciones de conflicto estratégico. Este trabajo es el resultado del análisis comparativo de distintas situaciones clásicas de conflicto.

2. IMPLEMENTACION DE AUTOMATAS

Existe un tamaño mínimo para un autómata que implementa una estrategia dada y este tamaño corresponde a la cardinalidad del conjunto de estrategias inducidas por f , (Ver Kalai-Stanford(1988) Aumann[1981]).

La siguiente es una descripción formal de un **Autómata Full** (para el jugador i -ésimo):

$$M = \langle M_i, m^0, B_i, T \rangle \text{ con } M_i \subset N$$

donde:

* M_i es el conjunto de estados.-

* m^0 es el estado inicial.-

* $B_i: M_i \rightarrow A_i$, es la función de Comportamiento (para cada estado prescribe una acción), donde A_i es el conjunto de estrategias disponibles para el jugador i -ésimo,

* $T: M_i \times M_{-i} \rightarrow M_i$, donde $A_{-i} = \prod_{j \in N, j \neq i} A_j$ es el conjunto de perfiles de acciones, es la

función de Transición, la cual describe como el autómata cambia de estados.

Si la función de Transición se restringe a:

$$T_i: M_i \times A_{-i} \rightarrow M_i, \text{ Con } A_{-i} = \prod_{j \in N, j \neq i} A_j,$$

se dice que M_i es un **Autómata Exacto**. Un autómata exacto solo toma en cuenta las acciones de los otros jugadores y no las propias acciones.

En este trabajo se simulan juegos con $[A]=2$, $A=\{ C, D \}$ donde $C= Cooperar$, $D= no Cooperar$.

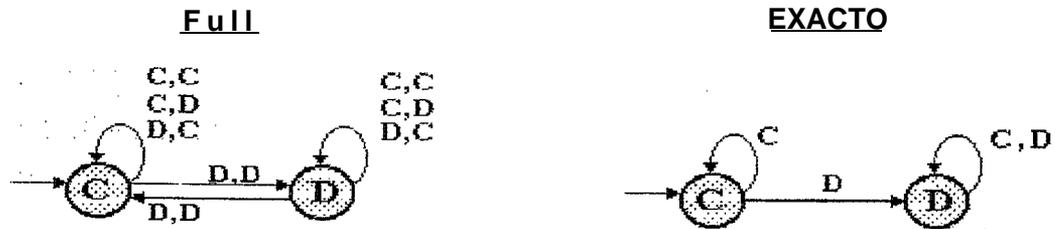
Un autómata Full o Exacto, puede representarse por medio de un vector de longitud $2n$, donde n es la cantidad de estados del autómata. Las primeras n componentes de dicho vector contienen: la función de comportamiento de cada estado en orden creciente, las últimas n componentes contienen la función de Transición de cada estado para toda acción 'a' (es decir una fila de la tabla de la función).

En este trabajo se utilizaron dos implementaciones distintas de la representación vista en el párrafo anterior para los autómatas. Una para la generación y almacenamiento de los autómatas, que se denominó de **bajo nivel**, y otra para hacerlos jugar, denominada de **alto nivel**.

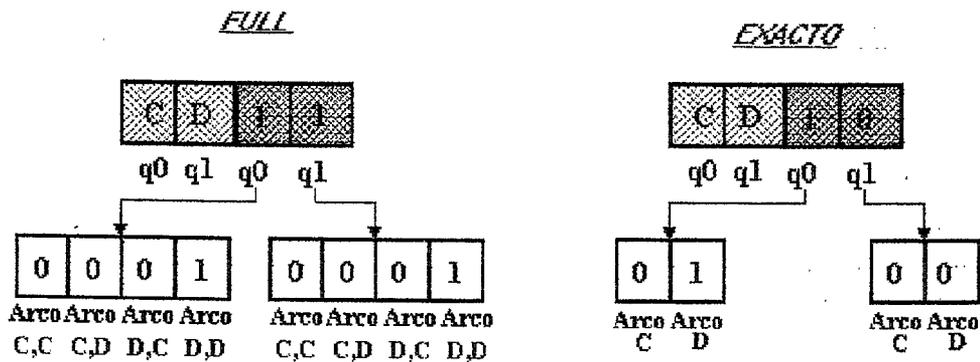
La codificación de bajo nivel se desarrolló de manera tal de optimizar el uso de memoria, ya que es la que se utiliza para soportar los autómatas generados en memoria principal. La codificación de alto nivel fue desarrollada para facilitar la manipulación de los autómatas por programa.

La figura que sigue muestra dos autómatas, uno full y otro exacto, de dos estados, junto con los diagramas de Moore y las codificaciones de bajo y alto nivel que les corresponden:

Los **Autómatas:**



Codificación de Bajo Nivel:



Codificación de Alto Nivel:

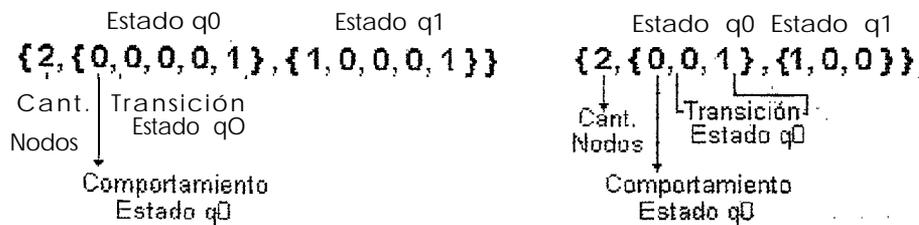


Fig. 1

Luego de generados todos los autómatas full y todos los autómatas exactos: de hasta dos estados, se simula el Juego Repetido de dos jugadores.

3. SIMULACIÓN DEL JUEGO USANDO AUTÓMATAS.

Un juego repetido modela sucesivos encuentros de agentes que enfrentan una situación de conflicto estratégico. En cada etapa, los agentes tienen que tomar decisiones;

esto es elegir estrategias (comportamientos) y obtienen pagos asociados con tales elecciones (ver figura,2.a.)

Dados dos autómatas, hacerlos jugar es implementar una serie de decisiones que se toman en un juego repetido. Esto es, comenzar con el par de comportamientos correspondientes a los estados iniciales de cada autómata, que llamaremos una jugada, y teniendo en cuenta esa jugada cada jugador se moverá a través de la función de transición de su autómata, después de la transición se vuelven a registrar los comportamientos y esto determinará la próxima jugada. De esta forma se obtendrá una sucesión de pares de comportamientos a los que se puede asociar una sucesión de pagos.

El uso de automatas finitos en un juego repetido infinitas veces ocasionará que a partir de cierto momento algunos estados del autómata se repitan en forma cíclica y 'la sucesión de acciones por ellos prescripta, se tornara periódica. Esta sera llamada fase cíclica y es de esta fase de donde se obtendrán, las ganancias de cada jugador por medio de un promedio (existe también una fase inicial que deberá sesgarse). Veamos un ejemplo-de lo expuesto.

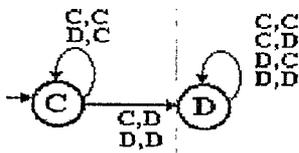
Sea la Matriz de pagos (correspondiente al Juego de los Prisioneros):

		Estrategias del Jug.2	
		C	D
Estrategias del Jug.1	C	2, 2	-1, 3
	D	3, -1	0, 0

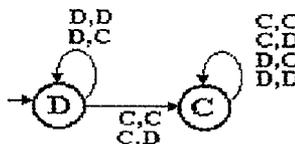
Fig 2-a

Sean los automatas Full:

Jugador 1:



Jugador 2:



Secuencia de comportamientos:
(acciones)

Jug.1 :	C	D	D	D	D	D
Jug.2 :	D	D	D	D	D	D

Estado Inicial

Secuencia de Pagos:

Jug.1 :	-1	0	0	0	0	0	0
Jug.2 :	3	0	0	0	0	0	0

Fig 2.b

Para cada jugador, la primera componente del par de acciones que figura en cada arco es la que corresponde a su propia acción, Es decir, la jugada inicial es C para el Jug.1 y D

para el Jug.2; esto implicará que el Jug. 1 debe moverse a través del arco C,D, y el Jug.2 a través del arco D,C.

Para analizar la conveniencia del uso de determinados autómatas que implementen estrategias en el juego Repetido, se juegan entre si, de a pares, todos los autómatas full y exactos de cada jugador, **y se construyen las Matrices de Ganancias del Juego de los Autómatas full y exactos respectivamente.** En cada matriz, la posición (i,j) contiene el par (p_{1i}, p_{2j}) donde la primera componente es la ganancia del jugador 1 y la segunda es la ganancia del jugador 2 correspondientes al juego de ambos con las estrategias implementadas por el autómata i-ésimo para el primer jugador y j-ésimo para el segundo.

4. EQUILIBRIOS

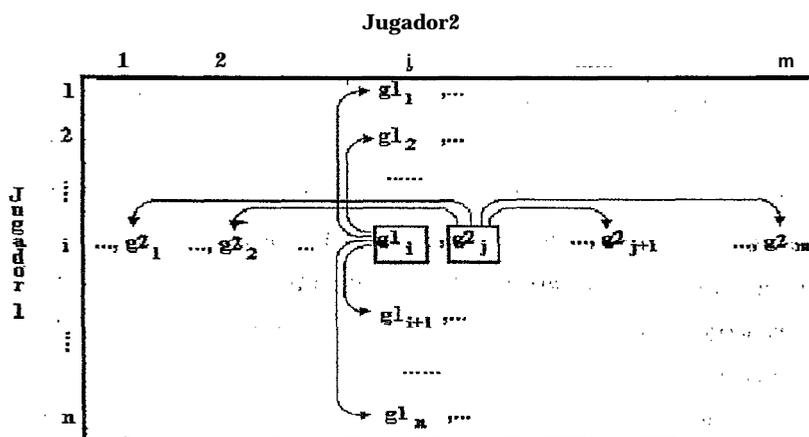
En este trabajo es posible computar tres tipos de Equilibrios: Nash, Subjuego Perfecto y 'Equilibrio Forgiving.

El equilibrio de Nash es muy importante ya que los otros dos equilibrios que se calculan en este trabajo como son refinamientos del mismo se calculan sobre los pares en equilibrio de Nash.

4.1. Equilibrio de Nash

4.1.1. Equilibrio de Nash sobre autómatas EXACTOS

Buscar un Equilibrio de Nash en 'la matriz de ganancias de los exactos significa buscar un par **Ganancia Jug.1, Ganancia Jug.2 en el** que se cumple que la primera componente es mayor o igual a toda otra ganancia del Jugador 1 en esa columna de la matriz y que la segunda componente es mayor o igual a toda otra ganancia del Jugador 2 en esa fila. Gráficamente :



i, j son estrategias en Equilibrio de Nash \iff $g1_i \geq g1_k, \forall k, (g1_k, g2_j)$ con j constante $\wedge k \neq i$
 $g2_j \geq g2_k, \forall k, (g1_i, g2_k)$ con i constante $\wedge k \neq j$

Fig.3

Tener en cuenta que cuando se llega a este punto en la ejecución de la aplicación, es posible que ya se haya procesado **DOMINACIÓN**. Esto significa que es posible que la

matriz de ganancia actual tenga menos filas y/o columnas que la generada originalmente, por lo tanto, cuando en el párrafo anterior se dice filas y/o columnas, esto se refiere a las filas y/o columnas no **eliminadas**.

4.1 .2. Equilibrio de Nash sobre autómatas FULL

El equilibrio de Nash sobre los autómatas Full, como ya se indicó, se realiza a partir de los resultados obtenidos sobre los autómatas exactos. Como se explicó en el punto 4, la dominación se calcula sobre la matriz de ganancias de los autómatas Exactos y estos resultados también deben hacerse extensibles a los autómatas Full.

En el caso de Dominación Fuerte y Regular la relación entre una fila/Exactos y n-filas/Full no presenta problemas, ya que el tipo de relación (mayor estricto o mayor igual con por lo menos un mayor estricto) asegura que si la dominación se calcula sobre la matriz de los autómatas Full, toda la familia de autómatas Full que corresponden a un exacto (las n-filas) será eliminada. En estos casos calcular Dominación sobre la matriz de ganancias de los autómatas exactos es equivalente a calcularla sobre la matriz de ganancias de los Full. Por lo que en este caso, para cada par de autómatas exactos e_1, e_2 en equilibrio, se consideran todos los pares de autómatas full f_1, f_2 posibles, tales que, f_1 tiene como autómata exacto equivalente a e_1 y f_2 a e_2 . Como se puede observar y por lo que se dijo anteriormente, para cada par de autómatas Exactos, se obtendrán $n_1 \times n_2$ pares de autómatas Full /donde n_1 es la cantidad de autómatas Full a los que les corresponde el primer autómata del par de Exactos y n_2 es la cantidad de autómatas Full a los que les corresponde el segundo autómata exacto del par).

En el caso de Dominación Débil, el orden de eliminación es muy importante, ya que el operador relacional que se utiliza es el mayor o igual. Eliminar estrategias dominadas en el caso de la matriz de ganancias de los exactos, significa que si una fila (columna) domina a otra, la segunda será eliminada. Pero recordar que a cada fila (columna) de la matriz de Exactos le corresponden n-filas (n-columnas) de la matriz de los Full, (que no necesariamente están consecutivas además); teniendo en cuenta, la equivalencia de los resultados no se puede asegurar (el orden de eliminación es fundamental en este caso). En este caso; para cada par de autómatas exactos e_1, e_2 en equilibrio, se hace lo siguiente: se toma un representante de todos los autómatas Full cuyo autómata Exacto es equivalente a e_1 y se hace lo mismo para e_2 por lo que la cantidad final de pares de autómatas Full en equilibrio

4.2. Equilibrios Subjuego' Perfecto

Este tipo de equilibrios, considera conductas que inducen equilibrios en cada subjuego. En el contexto de juegos' repetidos esto se define como Sigue: Dada Una estrategia del jugador i -ésimo f_i y una historia h (esto es, una sucesión finita de jugadas, denotamos f_i / h la estrategia definida por: $f_i / h (h') = f_i (h.h')$ para otrà historia h' ; donde $h.h'$ 'es' la concatenación de h y h' (esto es, la historia que consiste de poner la historia h luego de la historia h'). Para la historia vacia tendremos que $e.h = h.e = h$ para cada historia h . Intuitivamente f_i / h implica implementar la estrategia f_i luego de la historià h . Una **estrategia f es un Equilibrio Subjuego Perfecto para et juego repetido ("G"; "G"; Gt) si para toda historia h , f / h es un Equilibrio de Nash.** (Selten [1975] y Kalai[19871].

En otras' palabras, luego de observada la historia h , f_i / h es una estrategia a la que se puede asociar un autómata con igual estructura que el de f_i pero en el cuál' el estado inicial podría ser otro. O sea $(f_i / h$ para $h \in H$ se puede interpretar como el Conjunto mínimo de estados íde un autómata que implementa f_i .

Prácticamente, dados dos autómatas, para verificar que estén en equilibrio Subjuego Perfecto, se debería generar H (el conjunto de todas las 'historias, las de longitud 0, longitud 1, longitud 2, . .y así siguiendo), hacer jugar a los autómatas con cada historia, para cada historia ver en que estado quedó cada autómata, buscar para cada autómata el autómata equivalente pero con estado inicial igual al estado en que quedó después de lugar, Y para esos dos nuevos autómatas controlar si existe Nash.

Antes de clarificar esto con un ejemplo,' acotemos el para toda **historia** de la definición anterior. El Conjunto H puede ser refinado por la reflexión de equi-respuesta, ya que cualquier historia compatible con un autómata en particular debe dejar el 'autómata necesariamente en algún estado del mismo. Además se, puede probar que alcanza con que H contenga las historias de longitud menor o igual a $n \times m$ (donde n es' la cantidad de estados del Primer autómata y m es la cantidad de estados' del segundo autómata), para cubrir todas las posibilidades.

Veamos el ejemplo. Dados los autómatas:

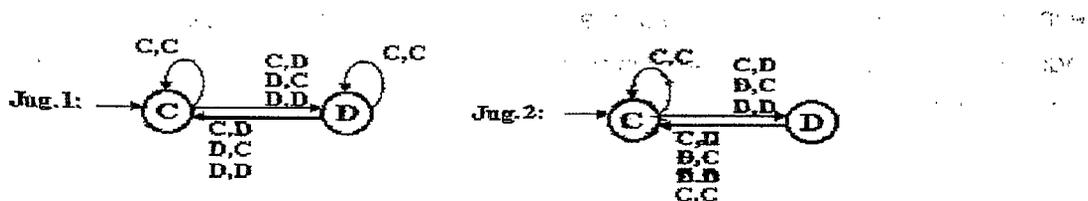


Fig 4.

Si se los hace jugar con una historia de longitud cero ($Long(h)=0$) quedan en sus estados iniciales. Si juegan con $h=\{(D,D)\}$ ($Long(\{(D,D)\})=-1$), se obtiene la combinación de estados $q1, q1$ para el Jugador 1 y el Jugador 2 respectivamente. Si juegan con $h=\{(D, D), (C,C)\}$, se obtiene la combinación de estados $q1, q0$ para el Jugador 1 y el Jugador 2 respectivamente. Si juegan con $h=\{(D, D), (C, C), (D, D)\}$, se obtiene la combinación $q0, q1$ para el Jugador 1 y el Jugador 2 respectivamente. En este ejemplo se obtuvieron todas las posibles combinaciones de estados para esos autómatas. Ahora para cada combinación de estados válida se deben buscar los autómatas que implementan las mismas estrategias que los autómatas que se verificaron pero con estado inicial igual al estado que marca la combinación a chequear (la idea es que se descarta una cierta historia inicial).

Si todos los pares de autómatas están en equilibrio de Nash, los autómatas originales están en equilibrio Subjuego perfecto,

Como la historia de longitud cero siempre me deja en los estados iniciales de los autómatas, la combinación $q0, q0$ siempre se obtendrá como válida, no importa cuales sean los autómatas elegidos. Esto significa que el par de autómatas sobre el que se vá a chequear el equilibrio Subjuego Perfecto debe estar en equilibrio de Nash.

Volviendo al ejemplo, para que el par de autómatas de la figura.4 esté en equilibrio Subjuego Perfecto deben cumplirse las siguientes condiciones:

- * el par de autómatas de la figura 4 debe estar en equilibrio de Nash.
- * el par de autómatas correspondientes al autómata que implementa la misma estrategia que el autómata del jugador 1 de la figura 4 pero con estado inicial igual a $q1$ y el autómata del Jugador 2 de la figura 4 deben estar en equilibrio de Nash. (esto corresponde a la combinación de estados $q1, q0$)
- * el par de autómatas correspondientes al autómata que implementa la misma estrategia que el autómata del jugador 2 de la figura 4 pero con estado inicial igual a $q1$ y el autómata del Jugador 1 de la figura 4 deben estar en equilibrio de Nash. (esto corresponde a la combinación de estados $q0, q1$)
- * el par de autómatas correspondientes al autómata que implementa la misma estrategia que el autómata del jugador 1 de la figura 4 pero con estado inicial igual a $q1$ y el autómata que implementa la misma estrategia que el autómata del jugador 2 de la figura 4 pero con estado inicial igual a $q1$ deben estar en equilibrio de Nash. (esto corresponde a la combinación de estados $q1, q1$)

Los autómatas :

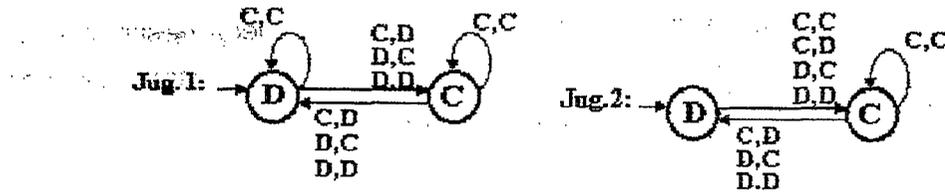


Fig 5.

implementan, la misma estrategia que los autómatas de la fig.4, respectivamente, si se hubiera considerada en estos otro estado inicial (en 'este caso, de autómatas. de dos estados sólo hay una posibilidad).

En, el ejemplo anterior, se dieron todas las combinaciones posibles de estados entre dos autómatas de dos estados, esto es, q_0, q_0 ; q_0, q_1 ; q_1, q_0 ; q_1, q_1 . Notar que hay pares de autómatas en los que alguna combinación no se da. Es decir, no existe historia tal que quede en alguno de los pares posibles.

4.3. EQUILIBRIO FORGIVING

Sea $G=(A,u)$ un juego finito de n jugadores. Sea $G^\infty = (F,u)$ la repetición infinita del juego G evaluado con el criterio del promedio.

Para una estrategia f_1 del jugador i , sea f_i / h la estrategia inducida después de la historia h .

Una estrategia f es un Equilibrio Forgiving de G^∞ (Kalai-Neme-Quintas[1993]) si se cumplen las siguientes dos condiciones:

Para cualquier n -upla de historias h_1, \dots, h_n

a) $(f_1 / h_1, \dots, f_n / h_n)$ es un Equilibrio de Nash para G^∞

b) Hay una historia consistente h (consiste de un juego finito -la longitud de h - de la estrategia $(f_1 / h_1, \dots, f_n / h_n)$), tal que $(f_1 / h_1 h, \dots, f_n / h_n h) = f$

Si la estrategia f es implementada por autómatas finitos, la condición a) significa que Si Se cambia el estado inicial del autómata esta será un equilibrio de Nash para cualquier n -upla de estados iniciales. Esto puede interpretarse por ejemplo, como que los jugadores podrian tener algunos errores de monitoreo y por lo tanto observar historias diferentes. La idea es que después de cualquiera de tales historias, el juego permanezca en equilibrio.

En particular para el caso $n=2$, dos estrategias implementadas por dos autómatas finitos están en equilibrio Forgiving, si para todas las combinaciones posibles de estados, los autómatas equivalentes están en equilibrio de Nash y juegan en fase (esto es, en todos los nuevos pares en equilibrio se obtiene. la misma ganancia que en los dos originales a evaluar).

Nótese que la diferencia más significativa entre este equilibrio y el equilibrio Subjuego Perfecto (ESP) se encuentra en el párrafo anterior, en el segundo puede pasar que no todas las combinaciones de estados sean válidas, por lo tanto es necesario generar historias y registrar solo las combinaciones de estados posibles.

5.- COMBINACIÓN'

Después que y autómatas juegan, tanto Los Full como los Exactos, y se obtienen las correspondientes matrices de ganancias del juego, es posible verificar distintas relaciones entre las filas/columnas de las mismas.

Si las ganancias del Jug.1 en la fila i son "mejores" que las ganancias del mismo jugador en la fila j para todas las columnas de dicha fila, esto quiere decir que la estrategia que implementa el autómata utilizado para obtener las ganancias de la fila i domina a la estrategia implementada por el autómata utilizado para obtener las ganancias de la fila j .

Es factible para los usuarios de este sistema eliminar las estrategias dominadas y calcular equilibrios sobre los resultados del juego reducido por tales eliminaciones.

Se implementaron tres tipos de Dominación, Fuerte, Regula y Débil. (Owen[1982]).

Dependiendo del tipo de Dominación, el orden de eliminación puede producir resultados distintos. Para cada tipo de Dominación, es posible eliminar en uno de tres órdenes posibles: Filas / Columnas, Columnas / Filas, Random.

6. EJEMPLOS ILUSTRATIVOS DE ALGUNAS SITUACIONES CLASICAS ANALIZADAS

Después de generar para cada jugador los autómatas de hasta dos estados del correspondiente Juego Repetido, se los hace jugar a todos entre si de a pares y con las ganancias obtenidas en el Juego con las distintas matrices de pagos, se construyen las matrices de Ganancia de los Autómatas Exactos y la de ganancia de los Autómatas Full. Se calculan los distintos Equilibrios implementados por este trabajo y se observan los siguientes resultados:

6.1. Dilema de los Prisioneros.

Dada la matriz de pagos:

	C	D
C	2, 2	-1, 3
D	3, -1	0, 0

*** EJECUCIÓN 1 ***

Dominación: Ninguna

Cantidad de Eliminaciones: 0 (Cero)

<u>Equilibrio:</u>	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	32		
Pares en equilibrio (Full) :	11.393	3.935	3.913
Pagos en equilibrio:	(0,0) (1,1) (2,2)	(0,0) (1,1) (2,2)	(0,0) (2,2)

Gráfica de Pagos en Equilibrio :



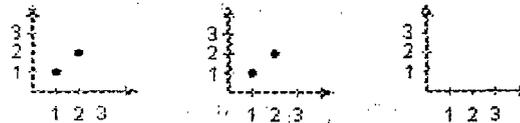
*** EJECUCIÓN 2 ***

Dominación: Regular.

Cantidad de Eliminaciones: 26 (veintiséis) filas/columnas

<u>Equilibrio:</u>	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	14		
Pares en equilibrio (Full) :	3.584	784	768
Pagos en equilibrio:	(1,1) (2,2)	(1,1) (2,2)	(2,2)

Gráfica de Pagos en Equilibrio :



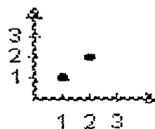
*** EJECUCIÓN 3 ***

Dominación: Débil

Cantidad de Eliminaciones: 28 (veintiocho) filas/columnas

<u>Equilibrio:</u>	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	1 4		
Pares en equilibrio (Full) :			
Pagos en equilibrio:	(2,2) (1,1)	(*) (*)	(*) (*)

Gráfica de Pagos en Equilibrio :



(*) El resultado varía dependiendo del autómatas Full elegido como representante.

*** EJECUCIÓN 4 ***

Dominación: Fuerte. **Cantidad de Eliminaciones:** 0 (Cero)

Equilibrios: Se mantienen los mismos resultados que en el caso de la EJECUCIÓN 1 ya que no se han eliminado estrategias.

6.2. Juego de Coordinación 1.

Dada la matriz de pagos:

	C	D
C	1, 1	0, 0
D	0, 0	1, 1

* EJECUCIÓN 1 *

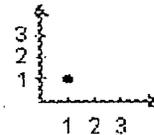
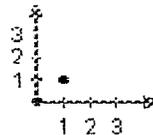
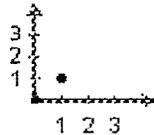
Dominación: Ninguna

Cantidad de Eliminaciones: 0 (Cero)

Equilibrio:

	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	76		
Pares en equilibrio (Full) :	25.825	13.139	3.937
Pagos en equilibrio:	(0,0) (1,1)	(0,0) (1,1)	(1,1)

Gráfica de Pagos en Equilibrio :



* EJECUCIÓN 2 *

Dominación: Fuerte

Cantidad de Eliminaciones: 0 (Cero). Equilibrios: (*)

EJECUCIÓN 3 *

Dominación: Regular.

Cantidad de Eliminaciones: 0 (Cero). Equilibrios: (*)

* EJECUCIÓN 4 *

Dominación: Débil.

Cantidad de Eliminaciones: 0 (Cero). Equilibrios: (*)

(*) Se mantienen los mismos resultados que en el caso de la EJECUCIÓN 1 ya que no se han eliminado estrategias.

6.3. Juego de Coordinación 2.

Dada la matriz de pagos:

	C	D
c	1, 1	0, 0
D	0, 0	0, 0

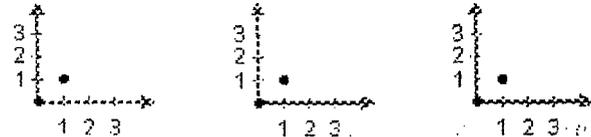
*** EJECUCIÓN 1***

Dominación: Ninguna

Cantidad de Eliminaciones: 0 (Cero)

<u>Equilibrio:</u>	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	32		
Pares en equilibrio (Full) :	11393	3935	3913
Pagos en Equilibrio:	(0,0) (1,1)	(0,0) (1,1)	(0,0) (1,1)

Gráfica de Pagos en Equilibrio :



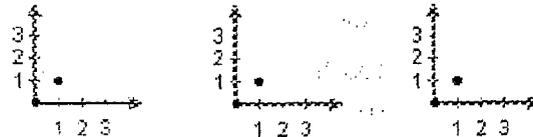
*** EJECUCIÓN 2 ***

Dominación: Regular.

Cantidad de Eliminaciones: 23 (veintitres) filas/columnas

<u>Equilibrio:</u>	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	8 0		
Pares en equilibrio (Full] :	26.849	13.410	13.121
Pagos en equilibrio:	(0,0) (1,1)	(0,0) (1,1)	(0,0) (1,1)

Gráfica de Pagos en Equilibrio :



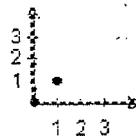
*** EJECUCIÓN 3 ***

Dominación: Débil

Cantidad de Eliminaciones: 24 (veinticuatro) filas/columnas

<u>Equilibrio:</u>	<u>Nash</u>	<u>Subjuego Perf.</u>	<u>Forgiving</u>
Pares en equilibrio (Exactos):	76		
Pares en equilibrio (Full) :	76	0	0
Pagos en equilibrio:	(0,0) (1,1)		

Gráfica de Pagos en Equilibrio :



*** EJECUCIÓN 4 ***

Dominación: Fuerte.

Cantidad de Eliminaciones: 0 (Cero)

Equilibrios: Se mantienen los mismos resultados que en el caso de la EJECUCIÓN 1 ya que no se han eliminado estrategias.

7 CONCLUSIONES .

El Equilibrio Forgiving, a pesar de ser un refinamiento del Equilibrio Subjuego Perfecto y por lo tanto tener más restricciones que este último, algorítmicamente resultó más fácil de implementar. Los tiempos de ejecución del calculo del Equilibrio Forgiving son ademas mucho menores.

En el estudio de los ejemplos clásicos, se observó una notable reducción en la cantidad de estrategias en equilibrio al pasar del Equilibrio de Nash a Equilibrios Subjuego Perfecto y una reducción menor al pasar de Equilibrio Subjuego Perfecto a Equilibrio.. Forgiving. Sin embargo, en el calculo del equilibrio Forgiving desaparecen algunos pagos **contra intuitivos que antes estaban en equilibrio.**

Respecto a Dominación, se pudo observar que con Dominación Fuerte (en general) no se reduce el conjunto de estrategias, lo que si sucede con las dominaciones Regular y Débil, las que a 'veces reducen además el conjunto *de* pagos en *equilibrio*. De todas formas se observó que la eliminación de estrategias dominadas no necesariamente produce un efecto notable en la reducción de la cantidad de estrategias en equilibrio, lo que si sucede al utilizar distintos tipos de equilibrios. Esta última afirmación surge del analisis comparativo de juegos clásicos, ya que inicialmente al trabajar con el Dilema de las Prisioneros, los mejores resultados respecto a la reducción del conjunto de estrategias se obtenían con la combinacion de dominación regular y Equilibrio Forgiving (ver Necco C., Quintas [1996]).

Un hecho significativo es el resultado obtenido para los ejemplos 6.1 y 6.2 (ver ejemplo)' con Equilibrio Forgiving, donde el único pago que resulta en equilibrio es el correspondiente a la cooperación plena. Esta selección de pagos en equilibrio, que intuitivamente es la más deseable, es muy difícil de obtener bajo otras condiciones de racionalidad no cooperativas.

Por último, cabe señalar que un mejor entendimiento de las consecuencias del uso de automatatas en problemas de decisión permitira en muchos casos un mejor uso de los recursos disponibles, una simplificación de las estrategias que hagan uso de tales recursos y una (mas amplia (y confiable) delegación de decisiones a programas de computación.

BIBLIOGRAFÍA

Aumann R. [1981], "Survey of Repeated games". In Essay in Game Theory and Mathematical Economics in Honor of Oskar Morgenstern, Bibliographisches Institut Mannheim, Wein Zurich, 11 -42.

- Kalai E. [1987]**, "Artificial Decisions and strategic complexity in repeated games", Proceeding of the International Workshop on Game Theory, Columbus, Ohio.
- Kalai E. , Neme A. and Quintas L [1993]**, "Forgiving Equilibrium", Anales XXII JAIIO, Tomo2, pags241 a 257.
- Kalai E. and Stanford W. [1988]**, "Finite Rationality and interpersonal complexity in repeated games", Econometrica 56 (2), 397-410.
- Necco C., Quintas L. [1996]**, "Implementación de autómatas Full para el estudio de soluciones del tipo Subjuego Perfecto en Juegos Repetidos". Anales XXV JAIIO, pag. 319 a 331.
- Owen G. [1982]**, "Games Theory", New York ,Academics Press.
- Quintas L. and Silvestri M. [1994]**, "Aplicación de autómatas en procesos de eliminación de estrategias dominadas en Juegos Repetidos", Anales XXIII JAIIO, pag 39 a 53.
- Selten R. [1975]**, "Reexamination of the perfectness concept for equilibrium points in extensive form games". Int. J. of Game Theory 4, 25-55.

SPF

UN AMBIENTE DE PROGRAMACION PARA FP

Ana M. Funes. Luis G. Quintas, Carlos Kavka

*Departamento de Informática - Fac. de Cs. Físico, Mat. y Nats.
Universidad Nacional de San Luis
Ejercito de los Andes 950 - Sari Luis
e-mail: afunes@unsl.edu.ar

*IMASL - Fac. de Cs. Físico, Mat. y Nats.
Universidad Nacional de San Luis
Ejercito de los Andes 950 - San Luis
e-mail: lquintas@unsl.edu.ar

Departamento de Informática - Fac. de Cs. Físico, Mat. y Nats.
Universidad Nacional de San Luis
Ejercito de los Andes 950 - San Luis
e-mail: ckavka@unsl.edu.ar

Resumen

El presente trabajo expone las distintas etapas en el desarrollo de un ambiente integrado de programación para un lenguaje funcional del tipo de los lenguajes FP propuestos por Backus[1978]. Ejecuta en computadoras con procesadores, Intel Y compatibles, bajo el ambiente MS-Windows. El ambiente SPF cuenta con un editor de textos, un traductor del lenguaje fuente a código intermedio y un intérprete o ejecutor de dicho código, además de un sistema de ayuda en línea con hipertextos.

1. Introducción

El sistema, denominado SPF por Sistema de Programación Funcional; "es un ambiente computacional que permite editar, compilar y ejecutar programas escritos en un lenguaje puramente aplicativo, libre del uso de variables, del tipo de los lenguajes FP (Functional Programming) propuestos por Backus[1978].

Fue desarrollado como parte de un proyecto más grande, el proyecto SGEV (Sistema Generador de Esquemas de Votación), consistente en el desarrollo de un sistema para la especificación y cálculo de esquemas de votación por medio de programas escritos en este lenguaje, ya que al ser estas funciones, pueden, fácilmente, ser expresados en términos de programas 'aplicativos. Esto resulta de utilidad para completar estudios teóricos en temas de Teoría de Elección Social. Otra motivación importante fue la de poder contar con una herramienta didáctica para la enseñanza de grado y 'posgrado en el área de lenguajes.

SPF ejecuta en computadoras con procesadores Intel y compatibles, bajo el ambiente MS-Windows y fue programado en lenguaje Borland C++ 3.1 haciendo uso de ObjectWindows, una biblioteca, de clases orientada a objetos, que encapsula el compartimiento que las aplicaciones Windows comúnmente tienen.

SPF cuenta con un editor de textos, para la escritura de los programas' en FP o de cualquier otro documento, un compilador que traduce a código intermedio los programas sintácticamente. Correctos o que informa al' usuario los errores detectados oportunamente, y un ejecutor que interpreta, en la 'etapa de ejecución del programa funcional, el código intermedio generado para el mismo. Cuenta, además con un sistema de ayuda en línea con hipertextos que hace uso del sistema de ayuda de Windows.

2. El sistema SPF

2.1. El lenguaje'

SPF es un ambiente computacional integrado para editar, compilar y ejecutar programas escritos en un lenguaje que es una extensión del lenguaje propuesto por Backus[19781] Tales extensiones comprenden el agregado de las funciones primitivas $<$, $<=$, $>$ y $>=$ más la forma funcional **readi**, que se definen a continuación. Las restantes funciones se definen de la misma manera en que se encuentran definidas en la propuesta original.

- $>$:
 $>:x \equiv x = \langle x, y \rangle \ \& \ x > y \rightarrow T;$
 $x = \langle x, y \rangle \ \& \ x <= y \rightarrow F; \text{Bottom}$
 $>$ compara por mayor dos elementos de una secuencia.

- $>=$:
 $>=:x \equiv x = \langle x, y \rangle \ \& \ x >= y \rightarrow T;$
 $x = \langle x, y \rangle \ \& \ x < y \rightarrow F; \text{Bottom}$
 $>=$ compara por mayor o igual dos elementos de una secuencia.

- $<$:
 $<:x \equiv x = \langle x, y \rangle \ \& \ x < y \rightarrow T;$
 $x = \langle x, y \rangle \ \& \ x >= y \rightarrow F; \text{Bottom}$
 $<$ compara por menor dos elementos de una secuencia.

- $<=$:
 $<=:x \equiv x = \langle x, y \rangle \ \& \ x <= y \rightarrow T;$
 $x = \langle x, y \rangle \ \& \ x > y \rightarrow F; \text{Bottom}$
 $<=$ compara por menor o igual dos elementos de una secuencia.

- readi:
readi x:y = z
 donde x es un átomo string parámetro opcional de la forma funcional, y es cualquier objeto y z es un átomo:entero ingresado por el usuario.

E j e m p l o s
readi 'ingrese el valor de la cuota': <1,2> = valor entero ingresado por teclado por el usuario al momento que SPF abra una caja de diálogo con la leyenda "ingrese el valor de la cuota".

readi : <1,2> = valor entero ingresado por el usuario al momento que SPF abra una caja de diálogo sin leyenda alguna.

2.2. Etapas de desarrollo

El desarrollo del sistema SPF se dividió en cuatro etapas consecutivas bien diferenciadas. La primera correspondió a la construcción de la interfaz principal de la aplicación y del editor, la segunda a la construcción de un traductor de FP a código intermedio, la tercera a un intérprete del código Intermedio y la cuarta y última al desarrollo de un sistema de ayuda en línea.

En la primera etapa se construyó la interfaz principal del sistema haciendo uso de lo que en Windows se denomina MDI. MDI es una interfaz estándar para las aplicaciones Windows que permite que el usuario trabaje con muchos documentos abiertos simultáneamente. Un documento, en nuestro caso, corresponde a una ventana de edición donde el usuario podrá escribir texto, en particular SU programa en lenguaje FP. Esta MDI es también la encargada de administrar el manejo de mensajes del menú de opciones, respondiendo a las opciones que le corresponden y enviando los mensajes que no reconoce a las interfaces locales. En cuanto al editor de textos, se hizo uso de la clase TFileWindow de Object/Windows que provee el comportamiento y los datos necesarios para soportar ventanas de edición de archivos, permitiendo las operaciones típicas de archivos (abrir, guardar, guardar como, etc.) y las de los editores (cortar, pegar, copiar, etc.).

Para la segunda etapa de desarrollo del sistema se tuvo en cuenta la metodología propuesta por Aho et al [1987] para la construcción de un compilador, que se basa en el *modelo de análisis y síntesis*, el cual realiza una descripción funcional de un compilador, dividiendo su funcionamiento en etapas o fases bien diferenciadas:

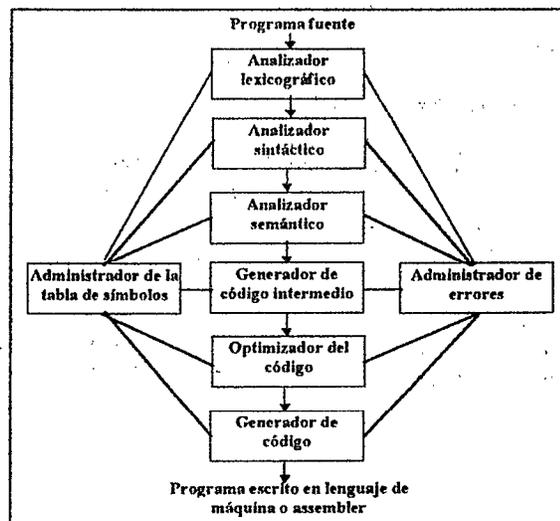


Fig. 1 Fases de un compilador

Las tres primeras fases conforman la 'porción del análisis dentro del modelo de análisis y síntesis. Las actividades de administración de la tabla de símbolos y manejo de errores, como se ve en la figura 1, interactúan con todas las otras fases. Las últimas tres etapas conforman la etapa de síntesis del compilador.

Respecto a nuestro compilador cabe aclarar que, en este modelo, se llega hasta la fase de generación de código intermedio, que produce como salida no un programa escrito en lenguaje de máquina o en lenguaje assembler, sino una representación intermedia del programa fuente. Esta representación intermedia es interpretada por un programa llamado el Ejecutor, implementado en la tercera etapa de desarrollo.

Se comenzó con la construcción del traductor de lenguaje FP a código intermedio, implementándose para esto el analizador lexicográfico, luego el analizador sintáctico, las rutinas de recuperación de errores y, por último, el generador de código intermedio.

Dado que no se trata de un compilador propiamente dicho ya que no genera código objeto, o de máquina sino código intermedio (Aho et al [1979][1987]), la tercera etapa correspondió a la construcción de un intérprete para ese código intermedio. En esta tercera etapa, se construyó, además, la interfaz adecuada para el ingreso de los datos de entrada a un programa FP.

En la última etapa, se desarrolló un sistema de ayuda en línea con: hipertextos, creada usando herramientas provistas por Borland para este fin que permiten generar archivos del tipo (archivos de ayuda de Windows).

2.2.1. El Traductor

El traductor del lenguaje FP implementado es de una pasada (Aho et al [1979][1987]), construido sobre la base de un analizador sintáctico descendente recursivo (Aho et al [1979][1987]) que interactúa, a medida que avanza el análisis sintáctico, con el analizador lexicográfico - encargado de aislar y devolver al analizador sintáctico las unidades sintácticas presentes en el programa fuente - y con el generador de código intermedio, mientras no se hayan detectado errores. El administrador de errores, presente en las tres tareas anteriores, se encarga, ante la presencia de un error, de informar el tipo y lugar del error, y de buscar un estado estable desde donde proseguir con el análisis sintáctico y lexicográfico. El chequeo de tipos se realiza en tiempo de ejecución, momento en el cual se conoce la naturaleza de los datos de entrada a los que se les aplicará la función implementada por el programa.

La tabla de símbolos es la encargada de absorber aquellas características que no son tipo 2 en los lenguajes de programación, características que no pueden ser expresadas por medio de una gramática tipo 2 o libre del contexto. Dado que el lenguaje FP es un lenguaje aplicativo puro, libre de variables, no posee declaraciones de estas últimas, por lo tanto el administrador de la tabla de símbolos se ocupará solamente de mantener los nombres de las funciones definidas por el usuario dentro de un programa, para así poder controlar aquellos aspectos tipo 1 que hacen a la redefinición de funciones y al uso de identificadores de funciones no definidas.

2.2.1 .1. El analizador lexicográfico

El analizador lexicográfico se encuentra en la primera fase del desarrollo del traductor. Su principal tarea es la de leer los caracteres de entrada y producir como salida una secuencia de **tokens** (Aho et al [1987]) que son usados por el analizador sintáctico.

Para esto, como primer paso se construyó una gramática libre del contexto, generadora del lenguaje anteriormente descrito, a partir de la cual se identificaron las distintas unidades sintácticas o tokens que conforman el lenguaje y que serían reconocidas y devueltas por el analizador lexicográfico al analizador sintáctico para su tratamiento. Esta gramática luego fue usada en la etapa posterior, para la construcción del analizador sintáctico. A continuación se da dicha gramática expresada por medio de una BNF:

<programa>	::= <definición> {<definición> }*
<definición>	::= def <identificador> = <composición de FF>
<composición de FF>	::= <forma funcional> { . <forma funcional> }*
<forma funcional>	::= <función> <constructor> <condicional> <constante> <insertar> <aplicar a todos> <binario a unario> <repetir> <leer entero>
<función>	::= <identificador> <entero>[r] tail id atom eq = < > <= >= null reverse distl distr length + - * % trans and or not apndl aphdr tlr rotl rotr
<constructor>	::= [<composición de FF> { , <composición de FF> }*]
<condicional>	::= <composición de FF> → <composición de FF> ; <composición de FF> (<composición de FF> → <composición de FF> ; <composición de FF>)
<constante>	::= const <objeto>
<insertar>	::= / (<composición de FF>) / <forma funcional>
<aplicar a todos>	::= @(<composición de FF>) @ <forma funcional>
<binario a unario>	::= bu <composición de FF> <objeto>
<repetir>	::= while <composición de FF> <composición de FF> (while <composición de FF> <composición de FF>)
<leer entero>	::= readl [<cte string>]
<objeto>	::= <átomo> <secuencia>
<átomo>	::= <entero> <cte string>
<secuencia>	::= <lista de objetos> <>
<lista de objetos>	::= <objeto> { , <objeto> }*

Fig. 2 BNF del lenguaje FP

Para la construcción del analizador lexicográfico se tuvieron en cuenta las siguientes observaciones:

* las palabras claves - símbolos terminales en la gramática, escritos en negrita - y los **lexemas** (Aho et al [1987]) correspondientes a los no terminales <nombre de archivo>, <identificado>, <entero> Y <cte string> conforman el conjunto de cadenas de caracteres que el analizador lexicográfico es capaz de reconocer

* los blancos no son significativos;

* las palabras claves son reservadas, por lo tanto no pueden ser utilizadas como identificadores;

* los comentarios encontrados durante el análisis son descartados, estos deberán ir encerrados entre llaves ().

El analizador lexicográfico fue construido sobre la base de un autómata finito determinístico (AFD) enriquecido en sus estados con acciones semánticas. El AFD es el siguiente:

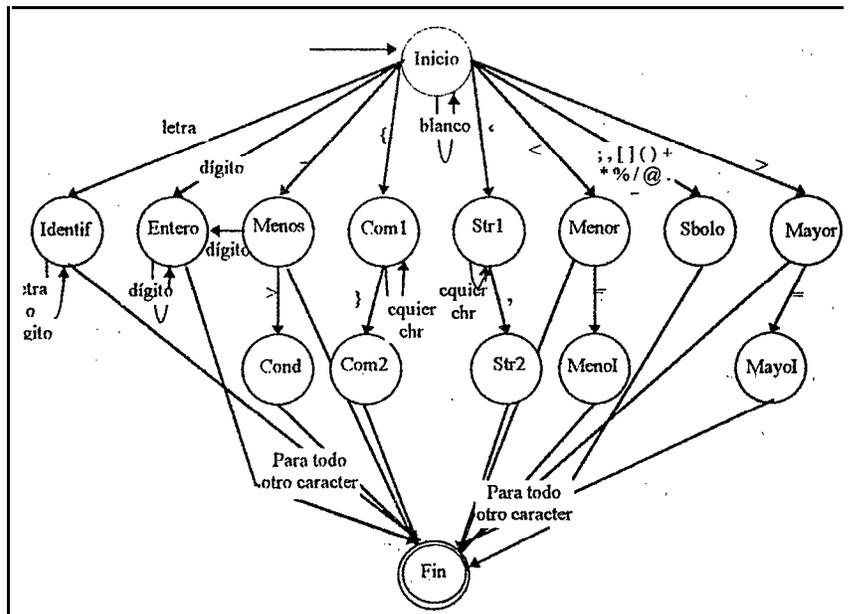


Fig. 3 AFD usado para la construcción del analizador Lexicográfico

Se definió una clase, a la que se llamó TScanner, capaz de contener todos los 'datos y el comportamiento' necesarios para soportar al analizador lexicográfico. Para su implementación se siguió un método sistemático que a cada estado le corresponde un segmento de código. Si existen, arcos saliendo del estado, entonces su código lee un carácter y selecciona el arco a seguir; si es posible. Se usa una función miembro NextChar() para leer el próximo carácter de la entrada, avanzar el puntero al carácter y devolver el carácter leído (Aho et al [1987]).

2.2.1.2. El analizador sintáctico

El análisis sintáctico o parsing es el proceso de determinar si una cadena de tokens puede ser generada por una gramática dada (Aho et al [1987]).

Una vez finalizado el desarrollo del analizador lexicográfico se comenzó con la implementación del analizador sintáctico trabajando sobre la base de la gramática dada en la figura 2. Se construyó así un tipo de analizador sintáctico top-down conocido como descendente *recursivo* consistente en un conjunto de procedimientos mutuamente recursivos encargados de llevar adelante el análisis. Más precisamente, se implementó una forma especial de parsing descendente recursivo, llamado parsing predictivo, en el cual se asocia un procedimiento con cada no terminal de la gramática y el símbolo de lookahead no ambiguo leído sobre la entrada determina el procedimiento asociado a un no terminal que va a ejecutarse (Aho et al [1987]).

Para poder implementar este tipo de parser, es necesario que la gramática sea LL(l) (Aho et al [1979] [1987]). Nuestra gramática presenta un problema con la producción

<condicional> ::= <composición de FF> + <composición de FF> ; <composición de FF>
 (<composición de FF> - <composición de FF> ; <composición de FF>)

donde <composición de FF> puede comenzar con cualquiera de los símbolos con que comienza una función. Este problema fue resuelto sobre el código de la siguiente manera: si el símbolo siguiente a la composición de formas funcionales es '-', entonces estamos seguros de que se trata de un condicional, en cuyo caso se asumirá como condición del condicional sólo la última función que aparezca en la composición. Si comienza con el paréntesis '(' tampoco se puede saber si se trata de un condicional o de una forma funcional repetitiva **while**, pero esto es inmediatamente resuelto cuando se lee el próximo símbolo sobre la entrada y se verifica si se trata de un **while** o no. En el caso que no lo fuese, estamos frente a un condicional cuya condición puede estar formada por la composición de cualquier número de funciones.

Se definió la clase TParserDialog, encargada tanto del análisis sintáctico como de crear y visualizar la Caja de diálogo donde se le muestra al usuario el resultado del proceso de compilación.

2.2.1.3 - La Tabla de Símbolos

La administración de la tabla de símbolos se encuentra a cargo del analizador sintáctico. Dado que el lenguaje FP es un lenguaje aplicativo puro, la tarea del administrador de la tabla de símbolos se vio reducida a los controles de redefinición de funciones y uso de un identificador de función no definido.

Una vez finalizada la construcción del analizador sintáctico, se pudo identificar claramente dentro de este programa los puntos donde el administrador de la tabla de símbolos actuaría.

Por un lado, se identificó, dentro de la rutina encargada del no terminal de la gramática <definición>, un punto donde el administrador de la tabla de símbolos debe recuperar del par (código, valor) -devuelto por el analizador lexicográfico al parser- el valor. En este punto, el código corresponde a un código de identificador de función y el valor al identificador de la función. El administrador de la **tabla de símbolos** se encargará de almacenarlo, controlando previamente que ya no exista, caso contrario el administrador de errores se ocupará de informar el error de redefinición de función para luego proseguir con el análisis.

Por otro lado, en la rutina correspondiente al no terminal <forma funcional> se pudo identificar otro punto donde también actuaría el administrador de la tabla de símbolos. Cuando el analizador lexicográfico devuelve al parser el código de identificador y el correspondiente nombre del mismo, el administrador de la tabla de símbolos chequeará si ese nombre se encuentra definido, es decir, si se encuentra almacenado en la tabla de símbolos, caso contrario, se informará el error correspondiente al uso de un identificador de función no definido.

La tabla de símbolos fue organizada como una lista encadenada, la que fue llamada **LFDU** por Lista de Funciones Definidas por el Usuario, en la que cada nodo corresponde a una función definida en el programa. La información almacenada en cada nodo de la lista, correspondiente a una función definida en el programa, consiste en un par:

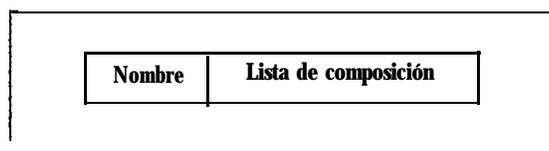


Fig. 4 Información contenida en un nodo de LFDU

donde **Nombre** corresponde al nombre de la función definida y **Lista de composición** es el acceso a una lista, inicialmente vacía, que contendrá las distintas funciones que conforman la parte derecha de la definición. Estas listas son construidas por el generador de código intermedio y serán explicadas con mayor detalle en la etapa de construcción del generador de código intermedio.

De esta forma, por ejemplo para el siguiente programa funcional

```
(Este programa calcula la longitud de una secuencia )
def suma = (null->const 0;/+)
def f = suma . @const 1
```

Fig. 5 Ejemplo de programa

tendremos que la tabla de símbolos, contendrá la siguiente información:

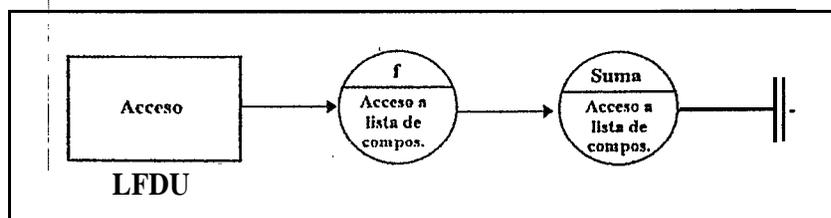


Fig. 6 Tabla de símbolos para el programa de la figura 5.

Como se puede observar en la figura, las funciones en la lista se encuentran almacenadas en orden inverso a como fueron definidas en el programa, es decir que LFDU en este caso es administrada como una estructura de pila, quedando de esta forma, para nuestro ejemplo, en el primer lugar **f**, la función aglutinante, por donde comienza la ejecución del programa.

2.2.1.4.. Detección y recuperación de errores

Un buen compilador debe ser capaz no solo de detectar errores sino de recuperarse de los mismos, es decir, aún ante la presencia de un error ser capaz de continuar con el análisis del programa completo, pudiendo de esta manera detectar todos los errores presentes en el mismo. En este sentido, SPF cuenta con un mecanismo de recuperación que implementa un método donde el analizador sintáctico realiza un salto lo suficientemente pequeño, descartando los símbolos de entrada hasta alcanzar un símbolo "válido" donde recuperarse, entendiéndose como válidos aquellos símbolos pertenecientes al conjunto de símbolos que pueden seguir a aquel en donde se detectó el error.

Se definió la clase TErroresWindow con la capacidad de mantener en memoria, a medida que avanza el análisis, un arreglo con los números de error detectados y la posición dentro del archivo en donde se encontraron. Esta clase contiene además el comportamiento necesario para crear y visualizar una ventana con los errores detectados, en el momento oportuno.

2.2.1.5. El chequeo de tipos

Un grupo muy importante de errores son aquellos relacionados con el chequeo de tipos, es decir, en nuestro caso, el control de la consistencia entre el dominio de aplicación de una función y el argumento al que se le aplicara.

Debido a la naturaleza del lenguaje, estos errores son detectados en tiempo de ejecución, momento en el cual se conoce la naturaleza de los datos de entrada a los que se les aplicará la función implementada por el programa. Provocan que la ejecución del programa sea suspendida.

2.2.1.6. La generación de código intermedio

La última de las etapas en la construcción del traductor correspondió a la implementación del generador de código intermedio.

Si bien un programa fuente puede ser traducido directamente en un lenguaje ensamblador o de máquina, existen algunas ventajas en usar formas de representaciones intermedias, por lo que se optó por una de estas. Por una parte podemos decir que son independientes de un procesador particular facilitando de esta forma la portabilidad; por otro lado, la generación de código eficiente, ya sea ensamblador o de máquina es una tarea complicada ya que se debe elegir un registro particular para mantener el resultado de cada computación, haciendo difícil el uso eficiente de los registros.

Existe un marco notacional para la generación de código intermedio que es una extensión de las gramáticas libres de contexto. Este marco, llamado esquema *de traducción dirigido por la sintaxis* permite que se asocien acciones semánticas o rutinas a las producciones de una gramática libre de contexto. Esas rutinas son las encargadas de generar el código intermedio cuando son llamadas apropiadamente -por el analizador sintáctico de la gramática. Los esquemas de traducción dirigidos por la sintaxis son útiles porque permiten al diseñador del compilador expresar la generación de código intermedio directamente en términos de la estructura sintáctica del lenguaje fuente; otra de sus ventajas es que son independientes del tipo de analizador sintáctico implementado y son fáciles de modificar. Ya que nuevas acciones semánticas pueden ser agregadas sin perturbar las traducciones ya existentes.

La representación intermedia del programa fuente elegida consiste en una estructura multinivel de listas heterogéneas unidireccionales, generada por las acciones semánticas implementadas en llamadas a rutinas que se encuentran embebidos en el analizador sintáctico. Estas rutinas se han implementado en los métodos de la clase *TGenCodigo* y de la clase *Lista*.

Como dijimos en el párrafo anterior, los nodos de estas listas son heterogéneos. Esto se implementó haciendo uso de una solución polimórfica que consistió en definir 'una clase Nodo que contiene los datos comunes a todos los tipos de nodos y clases derivadas que contienen los datos que distinguen a cada uno de los nodos. En la clase base se definieron además los métodos comunes a todos ellos.

Como vimos en el punto 2.2.1.3, cuando hablamos de la tabla de símbolos, a medida que avanza el análisis del programa, el traductor va construyendo LFDU, una lista unidireccional que mantiene el nombre de todas las funciones que se encuentran definidas en el programa fuente. Cada uno de los nodos de LFDU, además de mantener el nombre de la función, sirve de acceso a una lista heterogénea unidireccional con la representación intermedia correspondiente a la parte derecha de esa definición, donde cada uno de los nodos está asociado a cada una de las funciones que se encuentran compuestas entre sí en esa parte derecha. Es decir que la forma funcional composición es la que va encadenando los nodos de la lista, y los representamos gráficamente por los arcos entre los nodos rotulados con un punto. A su vez, los nodos de estas nuevas listas de segundo nivel serán nodos terminales u hojas sólo si se tratase de la representación de una función primitiva, de una definida por el usuario, de la forma funcional **const** o de la forma funcional **readi**; caso contrario, de ese nodo se desprenderá una nueva lista y así sucesivamente. Veamos a continuación cómo sería esta estructura para el caso del ejemplo dado en la figura 5.

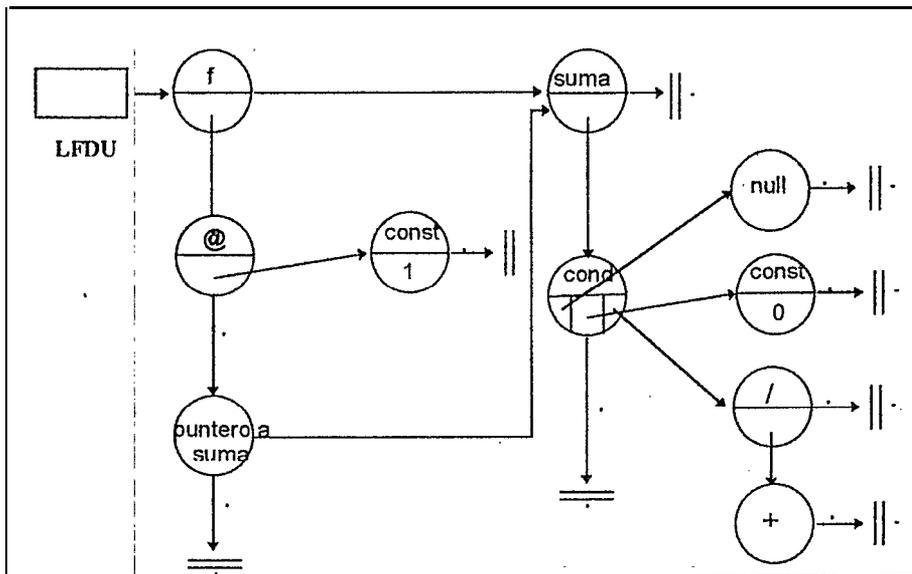


Fig. 7 Representación intermedia del programa fuente del ejemplo dado en la fig. 5

Como se observa en la figura 7, los nodos de primer nivel de LFDU corresponden a cada una de las definiciones. Para esto se definió la clase TDefinición, derivada de la clase *Nodo*, que encapsula el comportamiento y los datos para una definición.

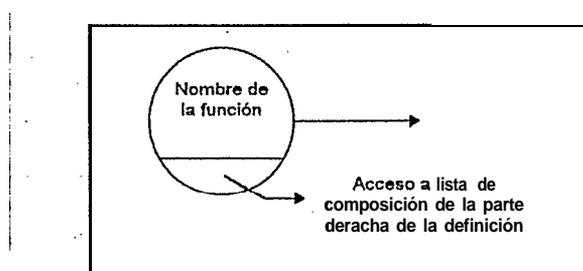


Fig. 8 Información almacenada para las definiciones de funciones

También se puede observar que los nodos se encadenan en orden inverso al de las funciones que se encuentran compuestas entre sí en la parte derecha de la definición. Fue hecho de esta manera para simplificar la evaluación de las funciones en la etapa de ejecución. Además, podemos ver que cada nodo tiene información específica al tipo de función que representa.

En el caso de las funciones primitivas sólo interesa almacenar el nombre de la función, el que se encuentra codificado. La figura 9 muestra la información contenida para el caso de las funciones primitivas. El arco representa la posibilidad de composición con otra función; en caso de que no exista tal composición apuntará a **nil**.

Para soportar este tipo de nodo se definió la clase TFuncionPrimitiva como una clase derivada de la Clase *Nodo*. Sólo se mantiene un código que sirve de identificador de la función primitiva en cuestión.

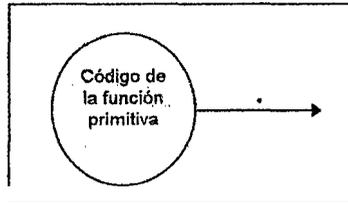


Fig. 9 Información almacenada para las funciones primitivas.

Para el caso 'de las funciones definidas por el usuario se almacena un puntero al nodo de la lista de primer nivel donde se encuentra la definición de la función, más el código para identificar' que se trata de una función definida por el usuario y no de una función primitiva (figura 10).

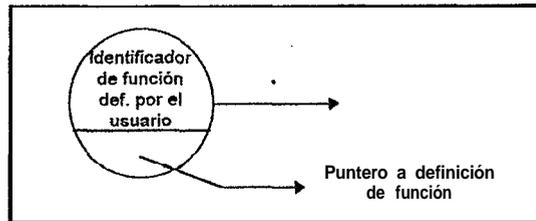


Fig. 10 Información almacenada para las funciones definidas por el usuario

La forma funcional **composición**, como fue mencionado anteriormente, es representada por medio de los arcos entre los nodos asociados a cada una de las funciones que se encuentran compuestas entre **Sí** y conforman la lista, El orden en el que encuentran almacenados los nodos es inverso al de escritura en el programa para facilitar la evaluación de las funciones, dado que por ejemplo para la expresión $f . g : x$ se evaluará primero $g : x$ y luego a este resultado se le aplicará f , es decir la evaluación es $f : (g : X)$ (figura 11).

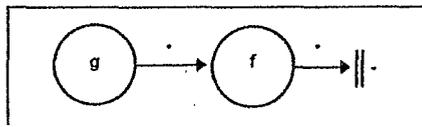


Fig. 11 Información almacenada para la forma funcional composición $f . g$

Para el caso de la forma funcional **const** solamente interesa mantener un puntero al objeto que es parámetro de esta forma funcional (figura 12). Este objeto puede ser un valor entero, un string, un **Valor** booleano o una secuencia. Dado que ya se contaba con la clase *Lista* y la clase *Nodo* se reutilizó este código para implementar las secuencias con el solo agregado de la definición de las clases derivadas de *Nodo*: *TEntero*, *TBoolean* y *TString*.

Para soportar los nodos correspondientes a la forma funcional **const** se definió la clase *TConstante* como una clase derivada de la clase *Nodo*.

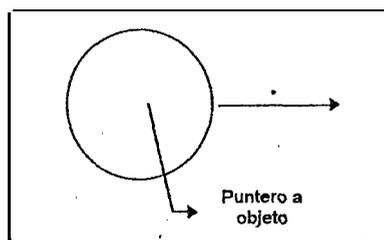


Fig. 12 Información almacenada para la forma funcional const.

Para la forma funcional **readi** solamente basta con mantener un puntero a la zona de Constantes string en donde comienza el mensaje que se visualizará en ejecución al momento de pedir el ingreso de un valor entero (figura 13). Como el mensaje es opcional este puntero puede valer ocasionalmente nil. se definió para soportar este tipo de nodo la clase *TReadint*, derivada de *Nodo*.

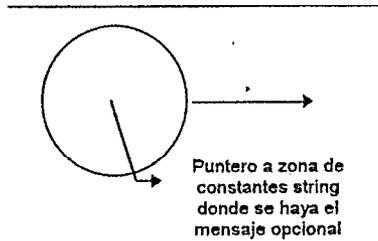


Fig. 13 Información almacenada para la forma funcional read

Las restantes formas funcionales son las que generan nuevos niveles en la estructura multinivel debido a que pueden involucrar la composición de nuevas funciones las que a su vez pueden ser en particular formas funcionales. En las figuras de abajo se muestra la información contenida en los nodos para cada una de ellas.

Para el caso de la forma funcional @ y / solamente se mantiene el acceso a una lista que contendrá la composición de las formas funcionales parámetros de la expresión que denota tales formas funcionales. Se definieron dos clases derivadas de *Nodo* para esto: *TApply* y *TInsert*.

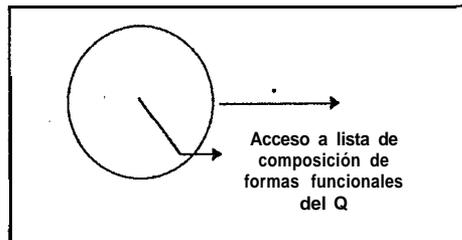


Fig. 14 Información almacenada para la forma funcional @.

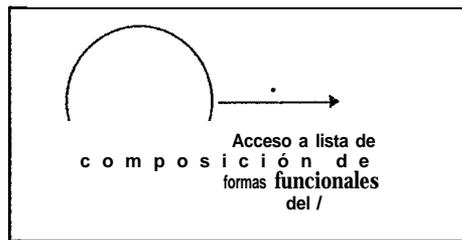


Fig. 15 Información almacenada para la forma funcional /

En la forma funcional condicional distinguimos tres partes a las que llamaremos condición, parte del verdadero y parte del falso. Cada una de estas partes puede involucrar composiciones de funciones por lo tanto del nodo correspondiente a una forma funcional condicional 'se descolgarán tres listas de **composición**. Este tipo de nodo se encuentra soportado por la clase *Tcondicional* que, **COMO** Ya explicamos, es derivada de la clase padre *Nodo*.

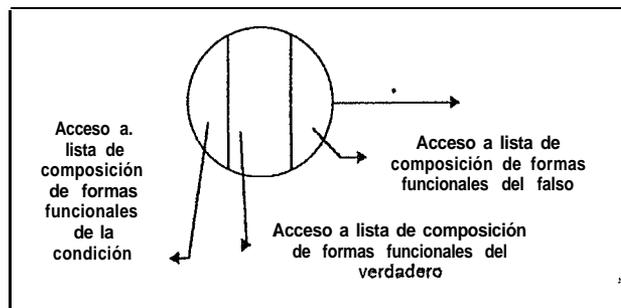


Fig. 16 Información almacenada para la forma funcional condicional

En la forma funcional while tenemos una condición y otra expresión que se aplica o no repetidas veces dependiendo del valor al que evalúe la condición. Cada una de estas dos partes puede involucrar

composición de varias funciones, por lo tanto, de los nodos de este tipo, definidos en la clase TWhile, se podrá acceder a dos listas de composición.

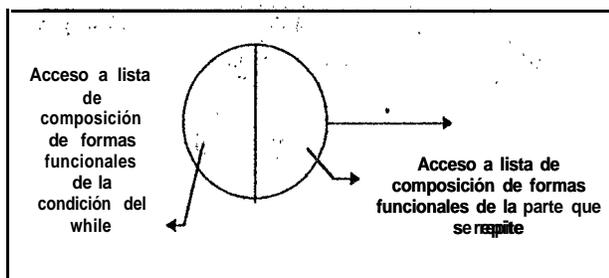


Fig. 17 Información almacenada para la forma funcional while

La forma funcional **constructor** es una expresión en donde aparece una secuencia' de subexpresiones donde cada una de ellas puede a su vez involucrar la composición de varias funciones. De aquí la necesidad de mantener una lista donde cada uno de sus nodos es la cabeza de una lista de composición. Para esto se definió la clase *TConstrucción*, una clase derivada de **Lista**, y a su vez Lista se definió como una clase derivada de **Nodo**.

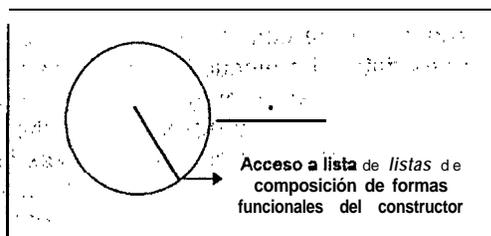


Fig. 18 Información almacenada para la forma funcional construcción.

Por Último nos queda la forma funcional **bu** para la cual se mantiene por un lado el acceso a una lista de composición correspondiente a la expresión funcional parámetro del bu y por otro lado un Puntero al objeto parámetro de esta expresión. Este objeto, al igual que en el caso de la forma funcional **const** puede ser un valor entero, un string, un valor booleano o una secuencia.

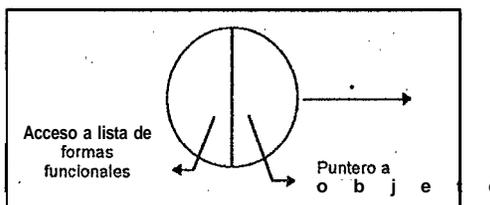


Fig. 19 Información almacenada para la forma funcional bu

2.2.2. El Ejecutor

Una vez finalizada la etapa de desarrollo correspondiente a la generación 'de código intermedio se Pasó a la construcción del programa ejecutor o intérprete del tipo de representaciones intermedias expuesto en el punto anterior. Conjuntamente, en esta etapa, se construyó la interfaz adecuada para el ingreso de los datos de entrada a un programa FP que ya se 'encuentra traducido a la representación intermedia. Dichos datos también son analizados sintácticamente para ver si pertenecen al conjunto de objetos válidos en nuestro lenguaje. En caso de no serlo se le informa al usuario por medio de un mensaje.

Dado que ya se tenía definidas las clases *Lista*, *TEntero*, *TBoolean* y *TString* que permiten la construcción de listas heterogéneas, solamente se agregó una nueva clase- hija de **Nodo**, la Clase **TBottom** para soportar el valor indeterminado **Bottom**.

Supongamos, por ejemplo, que el usuario ingresó los siguientes datos de entrada al programa del ejemplo dado en la figura 5:

<<<1>, <2>, <1,2>, <>>, <<2>, <1,2>, <1>, <>>, <<2>, <1,2>, <>, <1>>>>

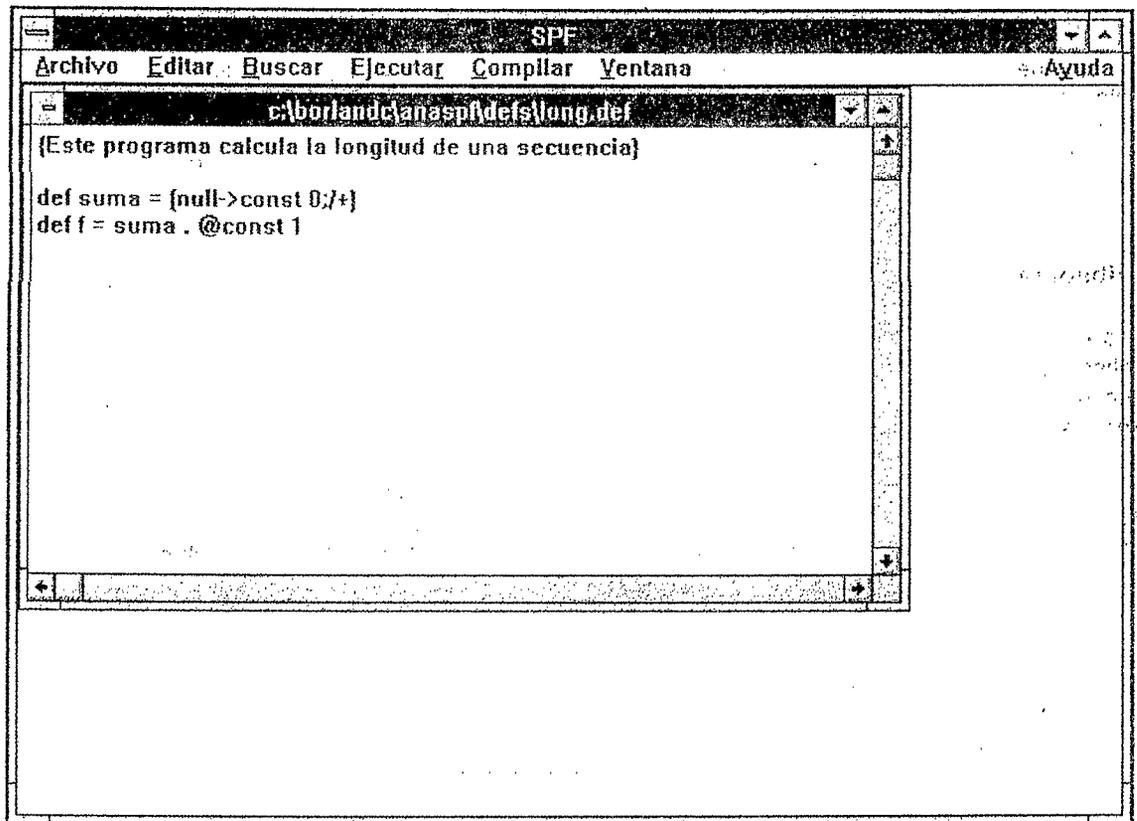


Fig. 21 Ventana de edición conteniendo un programa FP

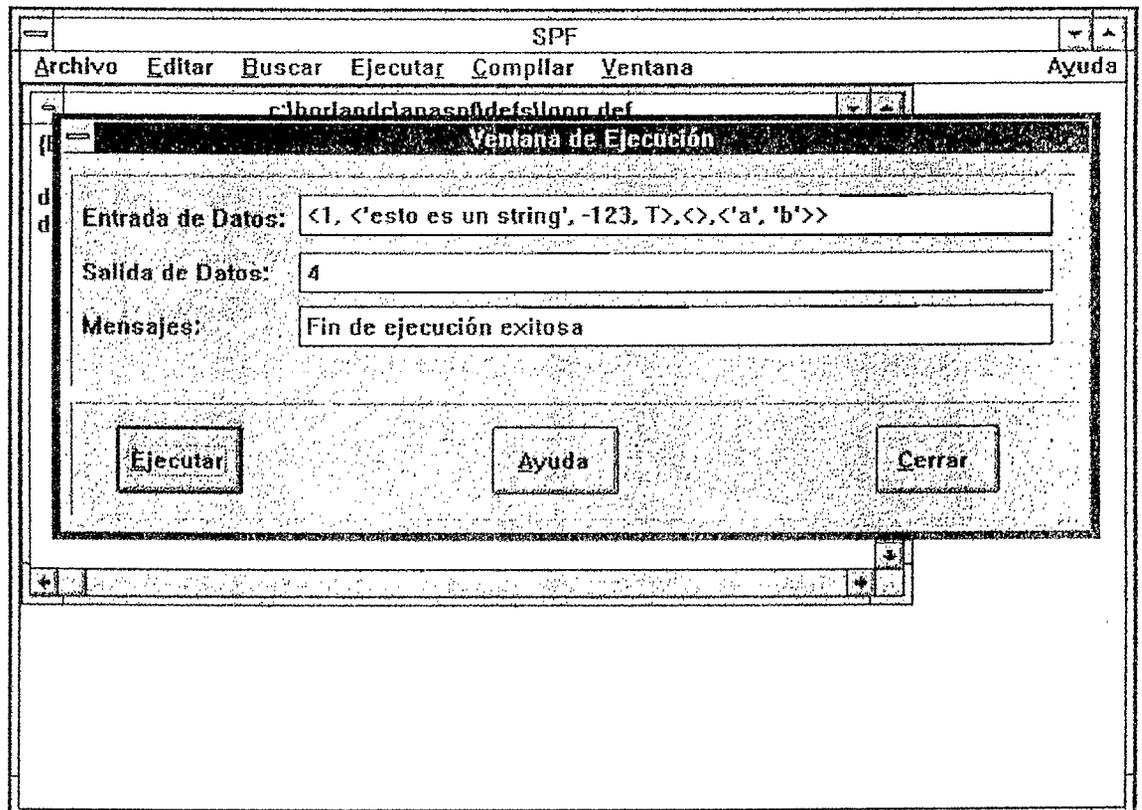


Fig. 22 Ventana de Ejecución

4. Conclusiones y posibles extensiones

Esta herramienta resulta de utilidad tanto en el área de investigación aplicada (Teoría de Elección Social) como en el área educativa en Ciencias de la Computación.

Entre las posibles extensiones de este trabajo está la construcción de un "debugger" para el compilador, que permita al programador, más fácilmente, detectar donde su programa falla siguiendo el curso de la ejecución del mismo.

5. Bibliografía

- Aho A., Sethi R. y Ullman J. - "Compilers. Principles, techniques and tools" - Addison Wesley Publishing Company - 1987.
- Aho A., Ullman J. - "Principles of Compiler Design" - Addison Wesley Publishing Company- 1979.
- Backus J. - "Can programming be liberated from the Von Neumann Style? ..." - Communications of the ACM - vol.21 N° 8 - 1978.
- Faison T. - "Borland C++ 3.1 - Programación Orientada a Objetos" - Prentice Hall. - 1993.
- Manuales de Borland C++ - Versión 3.1 - Borland International - 1993.
- Mc Cord J.- "Developing Windows Applications with Borland C++" - Sams Publishing -1992.
- "Microsoft Windows 3.1 Programming tools" - Microsoft Press - 1992.
- Norton P. y Yao P. - "Borland C++ for Windows" - 1992.

DETERMINACION DE RUTAS OPTIMAS DE DISTRIBUCION PARA LA CIUDAD DE BAHIA BLANCA

R.N. CASAL; RE. CORRAL; N.B. LOPEZ; E. ZITO; L.M. LOVOTTI,

RESUMEN

En este trabajo se estudian las características más relevantes de un sistema logístico de distribución, obteniéndose rutas óptimas mediante la utilización del Algoritmo de Dijkstra, definido dentro de un Sistema de Información Geográfica (SIG) aplicado a la ciudad de Bahía Blanca. Se incorporaron los nombres de las calles, sentido de circulación de las mismas, etc. y explicitaron las intersecciones, con lo cual se obtuvo la malla que define la conectividad de la red y a través de la cual el algoritmo permitió la determinación de las rutas óptimas y su cómputo métrico. Se comprobó la facilidad de empleo del modelo y su potencialidad, por lo que, Su uso les permitiría a las empresas y organismos, notables economías y ventajas diferenciales.

(*) Departamento de Ingeniería, Universidad Nacional del Sur, Av. Alem 1253 - (8000) Bahía Blanca

1. INTRODUCCION

Sin lugar a dudas uno de los aspectos más sobresalientes de los mercados actuales es el alto nivel de competitividad que presentan.

En este contexto, en muchos casos, los clientes perciben escasas diferencias técnicas entre las ofertas, lo que obliga a las empresas a incrementar las ventajas diferenciales a través del concepto de **servicio al cliente**

Tal servicio puede definirse como la provisión *consistente de utilidad de momento y lugar*. Expresado en otros términos significa que los productos aumentan su valor a los ojos de los clientes si los reciben en el momento y el lugar adecuado.

Las empresas! que han logrado el reconocimiento por su excelencia en el servicio y en consecuencia han logrado establecer ventajas diferenciales sobre su competencia, son aquellas en las que se ha fijado un alto nivel de prioridad en la atención al cliente y esto lo consiguen mediante lo que se ha dado en denominar gestión logística. Por consiguiente, cobra relevancia el análisis de aspectos logísticos tan variados como: estudio del tipo de mercadería a distribuir, su almacenaje, niveles de stocks, transportes empleados en la distribución, recorridos realizados, frecuencias de reparto, etc.

En línea con lo anteriormente mencionado se ha desarrollado el presente trabajo; en el cual se tratan aspectos teórico-prácticos útiles para la toma de decisiones en el ámbito logístico de la empresa. Fundamentalmente se **ha centrado** la atención en los temas relacionados con la distribución de productos y la prestación de servicios "puerta a puerta".

En el trabajo se describen conceptualmente las características mas relevantes del sistema de distribución y en base a aspectos propios detectados en el mercado local se ha planteado un modelo práctico de optimización. Este modelo tiene por objetivo facilitar la determinación de rutas óptimas (modelo de ruteo) entre un centro de abastecimiento y los puntos de recepción de pedidos considerando las restricciones de tránsito vigentes en la ciudad de Bahía Blanca. Esto es posible debido a que el modelo consiste de un algoritmo matemático definido dentro de un Sistema de Información Geográfica (SIG).

2. LOGÍSTICA

2.1. Definición ¹

Logística, es la planificación, organización y control del conjunto de las actividades de movimiento y almacenamiento que facilitan el flujo de materiales y productos desde la fuente al consumo para satisfacer la demanda al menor costo, incluidos los flujos de información y control.

2.2. Sistema logístico

El flujo de materiales o productos se descompone en tres sistemas:

2.2.1. Sistema de aprovisionamiento. Comprende las operaciones necesarias para **suministrar** al proceso productivo, las materias primas, las piezas y los elementos requeridos.

2.2.2. Sistema de producción. Comprende el proceso de transformación de los materiales, de ensamble de piezas y elementos, el resguardo de los productos terminados y el correspondiente pasaje al sistema de distribución.

2.2.3. Sistema de distribución. Procede a satisfacer las demandas de los clientes, ya sea directamente o por medio de depósitos intermedios. En este punto quedan comprendidos la prestación de los servicios llamados *puerta a puerta*".

¹ - Tomada de la Revista N°45 de la Asociación Argentina de Logística Empresarial - ARLOG - Junio / 95.

2.3. El problema logístico de distribución . Conceptualización.

En un sentido amplio, la gestión de las actividades logísticas de distribución de la empresa se reduce a un problema de diseño de una configuración en red, la que deberá ser efectiva y eficiente.

El interés del estudio estará centrado en la planificación y ejecución de esa red de distribución que permita a la empresa, tras una petición de productos efectuada por sus clientes, estar con los mismos en forma y tiempo en los lugares adecuados.

Conceptualmente este problema se trata como una red de nodos y arcos o enlaces. Los arcos representan el movimiento o transporte de mercaderías o productos entre los diferentes puntos de almacenamiento o consumo. Los nodos son los locales de venta al menudeo, o la residencia de quienes solicitan el servicio y el local de depósito del mayorista o distribuidor de mercaderías u origen del servicio.

La complejidad del diseño de la red se explica en la existencia de diferentes alternativas en el servicio **de** transporte, distintos tamaños de vehículos y distintos tipos de equipamiento de los mismos, diferentes recorridos y una amplia variedad de productos a repartir.

2.4 El sistema de transporte

“Normalmente, el transporte constituye el costo logístico individual más importante para la mayoría de las empresas”².

Según Richard E. Snyder³ y basado en estudios realizados sobre distintos tipos de industrias, se ha llegado a demostrar que el movimiento de cargas absorbe desde “un tercio a dos tercios de los costos logísticos . . .”.

La organización de los transportes en la empresa depende de la distribución geográfica de los Puntos o nodos a unir (fábrica, depósitos, bocas de expendio o clientes); distinguiéndose dos grandes categorías de problemas:

- . *La elección del medio a utilizar.*
- La programación de los *movimientos*.

-En el caso que nos atañe, la elección del medio de transporte se ve facilitada por cuanto, al tratarse del estudio de distribución de mercaderías o prestación de servicios puerta a puerta dentro del radio urbano de la ciudad de Bahía Blanca, ya están descartados todos los medios diferentes al automotor. En cuanto a la programación de los movimientos constituye el punto central a desarrollar en el resto del trabajo.

2.5. Programación de los transportes

Los transportes de mercaderías tienen por objeto satisfacer demandas geográficamente distintas entre sí desde los lugares donde los productos están disponibles. Constituye la unión entre entidades, de decisión diferentes (clientes y proveedor), donde cada entidad no es directamente responsable de la marcha de la otra. Por lo tanto, si se quiere obtener un sistema de transporte que funcione óptimamente reduciendo costos, es necesario un esfuerzo en común de las dos partes en adecuar su accionar en beneficio del otro.

Si se tiene en cuenta la importancia de los costos del transporte dentro de la estructura de costos logísticos (un tercio a dos tercios del total) es natural que dentro de las pautas establecidas para aumentar el beneficio económico de la empresas se encuentren las destinadas a mejorar la: eficiencia del sistema de transporte.

La cantidad de tiempo que las mercancías están en tránsito, se refleja en el número de envíos que se pueden hacer en un vehículo en un período determinado de tiempo y en el costo total de transporte de

² RH.Ballou - *Logística Empresarial - Control y Planificación* - 1985

³ Richard E. Snyder - *Physical Distribution Costo: a Two-Year Analysis* - Distribution Age

todos los envíos. Para reducir estos costos y mejorar el servicio al cliente, hay que seleccionar rutas y hacer planificaciones que minimicen el tiempo y la distancia del transporte.

En la problemática de selección de rutas óptimas en redes de transporte pueden presentarse básicamente los siguientes casos:

- a) Determinación de rutas óptimas cuando el nodo origen y el nodo destino son únicos y diferentes.
- b) Determinación de rutas óptimas cuando existen múltiples puntos origen y destino.
- c) Determinación de rutas óptimas cuando el punto origen y destino son coincidentes.

Como resultado del relevamiento efectuado en el ámbito local, se han detectado en principio necesidades de optimización de casos que se encuadran bajo el inciso a). Este **rubro que se define habitualmente como "servicios puerta a puerta" se presenta altamente competitivo y en consecuencia requiere contar con un sistema de prestación lo más eficiente posible. Por lo tanto, la selección de trayectos óptimos para los rodados encargados de la tarea de distribución influye directamente en la calidad y el costo del servicio prestado.**

3. PLANTEO DE UN MODELO DE OPTIMIZACION PARA LA PROGRAMACION DE TRANSPORTES DENTRO DE LA CIUDAD DE BAHIA BLANCA.

El modelo propuesto permite definir rápidamente y con precisión la ruta más corta que debe recorrer un despachante desde el centro de aprovisionamiento hasta el punto de entrega del pedido y también el trayecto de regreso, considerando la restricciones de tránsito vigentes en la ciudad de Bahía Blanca.

Se prevé en el futuro incorporar al sistema la definición de la trayectoria más rápida considerando penalizaciones por congestión del tránsito.

3.1. Teoría de redes. Conceptos y Nomenclatura.

Una red (grafo) está constituida por un conjunto de puntos (nodos) y un conjunto de líneas (**arcos** ó enlaces) que conectan pares de nodos y a través de las cuales puede circular algún tipo de flujo. Si el flujo puede circular en una única dirección se dice que el arco es orientado (dirigido) y se indica con una flecha del origen al destino. Si el flujo puede circular en ambas direcciones, el arco es *no* orientado (no dirigido) y se indica simplemente con, una línea que une los nodos en cuestión; este arco no orientado puede ser reemplazado por dos arcos orientados con sentido opuesto.

Una red en la que todos los arcos son orientados recibe el nombre de red orientada; también Se suele hacer referencia a este tipo de redes como *grafo dirigido*.

A la cantidad máxima de flujo que puede circular por un arco se la designa como la capacidad del **arco**. Así, si la cantidad que puede fluir por el arco es infinita, se dice que la capacidad del arco es ilimitada.

Con respecto a los nodos, se distingue entre nodos origen o **fuentes** (aquellos que generan flujo, expresado de otra forma, la cantidad de flujo que sale de los mismos es superior a la que *ingresa*), **nodos destino o sumidero** (aquellos que absorben flujo, en los que la cantidad que ingresa a los mismos excede a la que sale) y **nodos de transbordo** (aquellos en los que la cantidad de flujo que ingresa es igual a la cantidad que sale).

Se define camino (ruta) entre dos nodos a una sucesión de arcos que conectan ambos nodos. Un **camino orientado** del nodo *o* s al nodo *t* es aquél formado por una sucesión de arcos dirigidos todos hacia el nodo *t*. Un **camino no orientado** es una sucesión de arcos dirigidos indistintamente hacia *o* desde el nodo *t*.

Un ciclo es un camino que comienza y termina en el mismo nodo. De igual modo que los caminos, los ciclos pueden ser orientados o no.

Un **recorrido (tour)** es un ciclo que incluye todos los vértices de un grafo.

3.2. Modelo propuesto.

A los efectos de definir rigurosamente la ruta óptima entre dos puntos cualesquiera del mapa de la ciudad de Bahía Blanca se ha implementado un algoritmo típico para los casos donde el nodo origen y el destino son únicos y diferentes denominado algoritmo de Dijkstra dentro de un Sistema de Información Geográfica (SIG), quien es el encargado de suministrar al algoritmo la información necesaria para los cálculos.

La implementación se realiza de manera tal que contempla las siguientes restricciones de tránsito:

- a) sentido de circulación real de las calles y avenidas.
- b) giros permitidos acorde con la semaforización de la ciudad.

3.2.1. El algoritmo de Dijkstra. Descripción

Se considera en esta sección, una red orientada en la que cada arco tiene un peso no negativo (W_{ij}) y donde un vértice se especifica como nodo origen, El peso asignado a cada arco puede ser un valor de distancia, tiempo de viaje, costo, consumo de combustible o cualquier otro atributo. El problema, consiste en encontrar el camino más corto del origen a cualquiera de los nodos de la red, donde la **longitud del camino es** la suma de los pesos de los arcos que integran ese camino.

Este algoritmo, conocido también como **algoritmo de/as etiquetas**, permite encontrar el camino más corto entre un determinado nodo origen s y un nodo destino t o determinar el camino más corto entre el nodo origen y todos los restantes nodos de la red.

Corresponde al tipo de algoritmos conocidos como **ávidos** (greedy), pues siempre escoge el Vértice más cercano al origen de entre aquellos cuyo camino más corto todavía se desconoce (teniendo Presente que se trabaja con pesos no negativos en los arcos).

Se comienza asignando una etiqueta permanente 0 al nodo origen s (pues cero es la distancia de S a sí mismo). A todos los otros nodos se les asigna etiqueta temporaria (distancia infinita), pues aún no han sido alcanzados.

Luego se etiqueta cada nodo v (sucesor inmediato) conectado directamente con el nodo origen s Con rótulos temporarios en los que se indica el peso del arco (s,v) . De entre todos los sucesores inmediatos de S , el vértice con la etiqueta temporaria de menor valor (llámese x), es el nodo más cercano a s . como todos los arcos tienen pesos no negativos, no puede existir un camino más corto de s a x , entonces la etiqueta del Vértice x se convierte en permanente.

A continuación se buscan los vértices conectados directamente al nodo x y se reducen los valores de sus etiquetas temporarias si el camino desde s a cualquiera de ellos es más corto pasando por x que sin pasar por x . Ahora, de todos esos nodos con etiqueta temporaria, se escoge aquél con menor valor de distancia (llámese y), y se convierte su etiqueta en permanente. Este nodo es el segundo nodo más próximo al nodo origen s .

Así, en cada iteración, se reducen los valores de etiquetas temporarias cuando sea posible (seleccionando un camino más corto a través del nodo más recientemente etiquetado en forma permanente), Se Selecciona aquél nodo con etiqueta temporaria de menor valor y se le asigna etiqueta permanente: Se continúa de este modo hasta que el nodo destino t adquiere etiqueta permanente.

A fin de distinguir los nodos etiquetados en forma permanente, de aquellos con etiqueta temporaria, se 'utiliza un vector de Boole, $S[v]$, de orden n . Cuando el i -ésimo nodo adquiere etiqueta permanente, el i -ésimo elemento de este vector cambia de **falso a verdadero**. Se utiliza un arreglo $D[v]$ de **Orden** n para almacenar el valor de las etiquetas de los nodos. Se utiliza otro arreglo $P[v]$ que almacena el

vértice inmediato anterior a v en el camino más corto desde s a t . Se usa una variable **recent** que guarda el último nodo etiquetado en forma permanente.

Suponiendo que la red se da en la forma de una matriz de peso $W=[W_{ij}]$ con pesos ∞ para arcos inexistentes, especificados un nodo origen s y un nodo destino t , el algoritmo de Dijkstra se resume a continuación:

```

INICIO
for all  $v \in V$  do
begin
     $D[v] \leftarrow \infty$ ;
     $S[v] \leftarrow \text{falso}$ ;
     $P[v] \leftarrow -1$ 
end;

{Se etiqueta en forma permanente el nodo  $s$  con 0. El resto de los nodos se etiqueta temporariamente con  $\infty$ . El nodo  $s$  es el último nodo etiquetado en forma permanente}
 $D[s] \leftarrow 0$ ;
 $S[s] \leftarrow \text{verdadero}$ 
 $recent \leftarrow s$ 

ITERACION:
while  $S[t] = \text{falso}$  do
begin
    for todo nodo  $v$  inmediato sucesor de  $recent$  if not  $S[v]$  do
        begin {Se actualizan etiquetas temporarias}
             $newlabel \leftarrow D[recent] + W_{recent, v}$ 
            if  $newlabel < D[v]$  then
                begin {Se cambia la etiqueta de  $v$  si existe un camino más corto a través del nodo  $recent$  y se guarda el nodo que figura en la variable  $recent$  como predecesor inmediato del nodo  $v$  en el camino más corto desde  $s$ }
                     $D[v] \leftarrow newlabel$ 
                     $P[v] \leftarrow recent$ 
                end
            end
        end

    end

    Sea  $y$  el nodo con etiqueta temporaria de menor valor ( $\neq \infty$ )
    { $y$  es el nodo más próximo a  $s$  que adquiere etiqueta permanente}
     $S[y] \leftarrow \text{verdadero}$ ;
     $recent \leftarrow y$ 

end

```

Finalizada la ejecución del algoritmo, la longitud del camino más corto estará dada por el valor $D[t]$. Para reconstruir el camino más corto de s a t , se regresa por los vértices predecesores del arreglo P .

Se presenta a continuación una red de pocos nodos a fin de visualizar cómo trabaja este algoritmo:

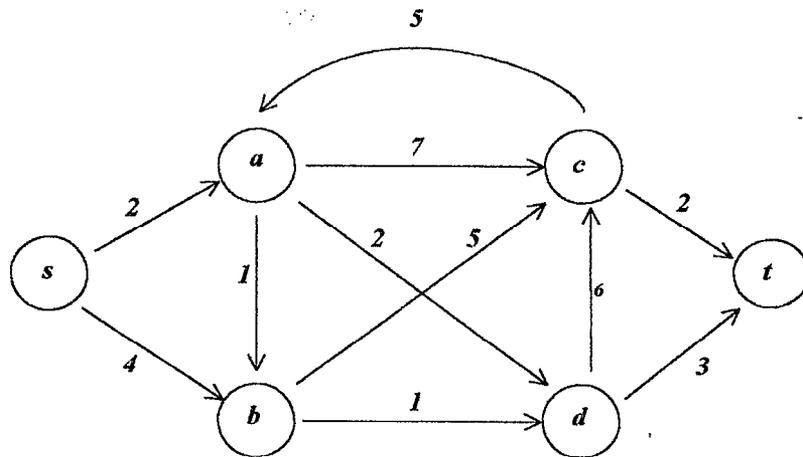


Figura 3.1.

Inicialmente, se asigna al nodo origen s la etiqueta permanente con valor de distancia 0, y a los restantes nodos etiquetas temporadas con valor ∞ . Los nodos directamente conectados a s son a y b ; se reduce el valor de sus etiquetas temporadas de ∞ a 2 y 4, respectivamente. La menor de las distancias corresponde al nodo a , por lo tanto este nodo adquiere etiqueta permanente.

En la iteración siguiente, se consideran los nodos que suceden inmediatamente al nodo a (el último nodo etiquetado en adquirir etiqueta permanente); éstos son los vértices b , c y d y se reducen los valores de sus etiquetas temporadas de ∞ a 3, 9 y 4 respectivamente.

Se continúa de la misma forma hasta alcanzar el nodo destino t . En la **Figura 3.2** se indican las sucesivas iteraciones y los valores que van tomando las etiquetas de todos los nodos. Las etiquetas Permanentes se identifican con un círculo.

	s	a	b	c	d	t
Inicio	0	∞	∞	∞	∞	∞
1ra. Iteración	0	2	4	∞	∞	∞
	0	2	4	∞	∞	∞
2da. Iteración	0	2	3	9	4	∞
	0	2	3	9	4	∞
3ra. Iteración	0	2	3	8	4	∞
	0	2	3	8	4	∞
4ta. Iteración	0	2	3	8	4	7
	0	2	3	8	4	7

Figura 3.2.

En la figura 3.3. se visualiza el líneas de trazo grueso el camino más corto desde el nodo *s* al nodo *t*, determinado a partir de la aplicación del algoritmo de Dijkstra.

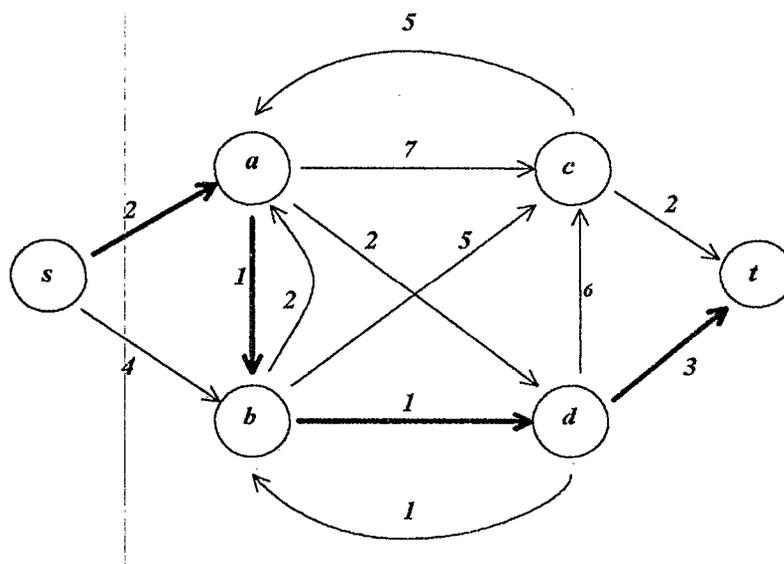


Figura 3.3.

Comentarios

En el ejemplo presentado se finaliza la ejecución del algoritmo cuando al nodo destino *t* se le asignó etiqueta permanente. Sin embargo, se podría haber continuado su ejecución hasta que todos los nodos de la red tuvieran etiqueta Permanente. Así, se podrían haber determinado los caminos más cortos desde *s* hasta todos los nodos de la red. Como en cada ejecución del bucle un nodo adquiere etiqueta permanente, la modificación del algoritmo consiste simplemente en cambiar el bucle **while** por

```
for x ← 1 to n do
```

para poder obtener los caminos más cortos desde *s* a todos los nodos de la red.

Si el problema consiste en encontrar el camino más corto entre todos los pares de nodos, puede aplicarse el Algoritmo de Dijkstra tomando por turno cada vértice como vértice origen; sin embargo, una forma más directa de solución es mediante el algoritmo creado por R. W. Floyd.

3.2.2 El Software utilizado.

Para la aplicación práctica del modelo se ha utilizado el Sistema de Información Geográfica GLOBO (marca registrada), software que se ejecuta sobre Autocad v. 12 para Windows y que ha sido íntegramente desarrollado por una empresa de Bahía Blanca.

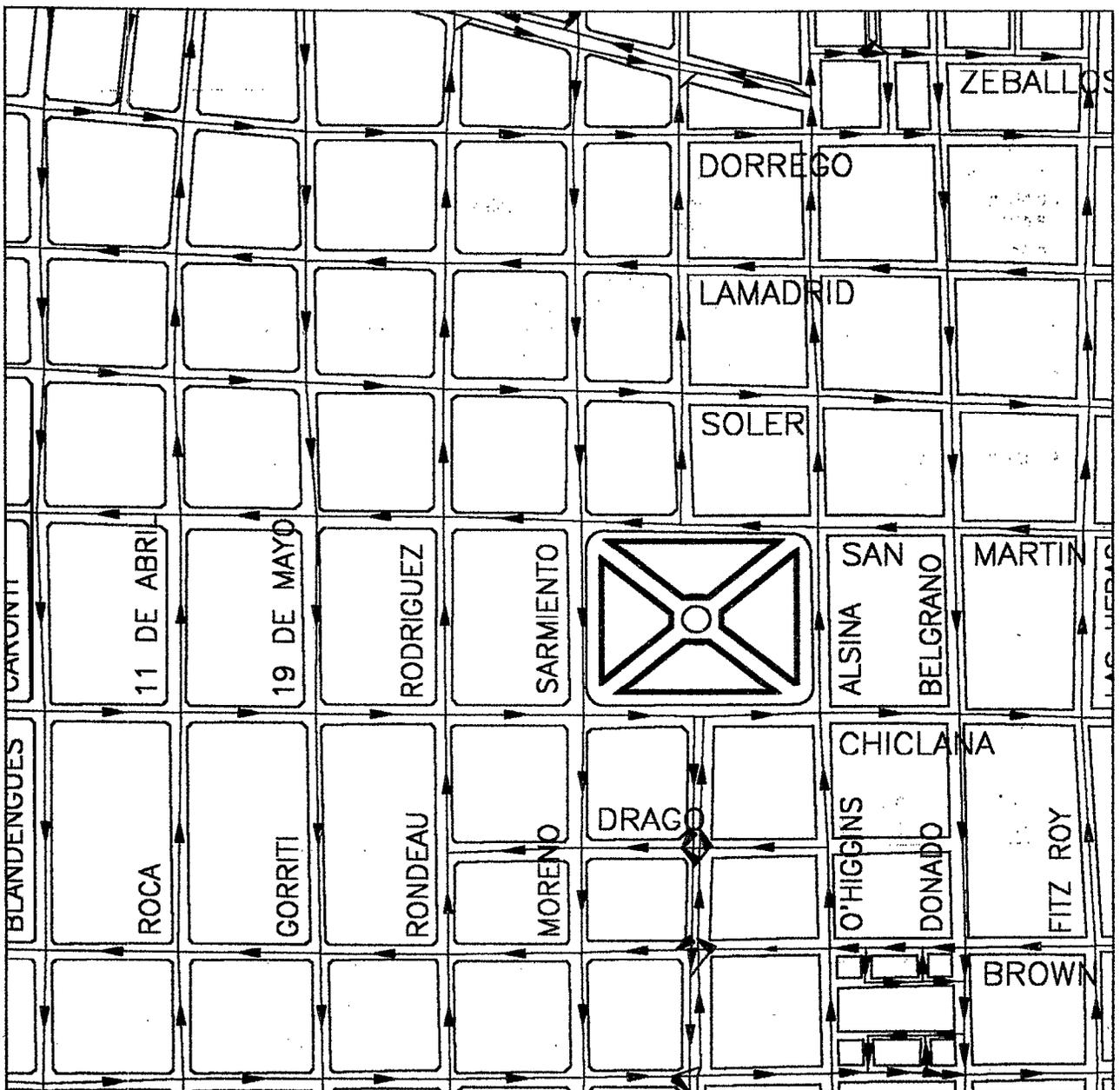
A los efectos de este trabajo se requirió la incorporación dentro del software del Algoritmo de Dijkstra, conjuntamente con un modulo para el ingreso de los datos de conectividad de la red y otro para el trazado de las trayectorias optimas y el cómputo métrico entre dos puntos cualesquiera de la ciudad. El grupo de trabajo tuvo a su cargo la incorporación de la malla de calles de la ciudad con el correspondiente sentido de circulación y demás datos geográficos.

Dicha malla define la conectividad de la red de la ciudad y sus puntos son definidos al momento de dibujar las calles, sus sentidos y sus intersecciones. En la Fig. 3.4 se muestran algunos nodos de la red con sus correspondientes vas de circulación tal como se definen en el sistema.

Cabe destacar que la modularidad del programa y la capacidad de exportar datos y resultados permite en el futuro implementar otros algoritmos más sofisticados que utilicen la misma malla de nodos creada específicamente para este trabajo a partir de la base de datos de calles (nombres y orientaciones de tránsito), incorporando un mayor número de restricciones al problema.

El sistema mencionado permite efectuar también el cómputo lineal de las rutas óptimas obtenidas.

Figura 3.4.



3.2.3. Plano del área seleccionada

A los efectos de este trabajo se ha definido como área de acción la comprendida entre las calles Chile, Sixto Laspiur, Av. La Plata, Av. 1° de Marzo, Av. Alem, Córdoba, Av. Gral. Urquiza, Av. del Trabajo, vías del Ferrocarril y Río Negro, de la ciudad de Bahía Blanca.

No se incluye el plano de la ciudad en el que se observa la red incorporada específicamente para la implementación de algoritmos de selección de ruta óptima dado el tamaño del mismo.

3.2.4. Listado parcial del archivo que contiene la información de la malla de para el área seleccionada en 3.2.3.

En la tabla siguiente, para cada nodo de la red considerada, se indican: las coordenadas (x,y) que dan la posición del nodo respectivo en el plano de la ciudad dibujado sobre Autocad, y el nombre de la(s) calle(s) a que corresponde. Esta tabla es solo una pequeña parte del listado total de calles.

N° Nodo	Coordenada X	Coordenada Y	Código (nombre) de Calles
1	4564511.45	5715077.23	ZEBALLOS
2	4564514.22	5715053.24	PORTUGAL/ZEBALLOS
3	4564613.39	5715186.47	PORTUGAL
4	4564629.92	5715181.89	CORRIENTES
5	4564625.04	5715189.20	CORRIENTES/PORTUGAL
6	4564621.17	5715471.36	H. YRIGOYEN
7	4564632.99	5715455.06	ESPORA
8	4564758.86	5715596.88	H. YRIGOYEN
9	4564759.27	5715609.12	AV. GRAL. URQUIZA
10	4564765.85	5715598.44	AV. DEL TRABAJO/AV. GRAL. URQUIZA/H. YRIGOYEN
11	4564610.19	5715453.15	H. YRIGOYEN/H. YRIGOYEN
12	4564576.62	5713253.01	BERUTTI/MISIONES

A continuación, también para cada nodo, se indica su conectividad con los nodos más próximos según el sentido de circulación.

Conectividad									
1	2		12	847	13	23	25	24	
2	6 2 5	621	13	830	832	24	345	25	
3	1		14	18	15	25	25	24	
4	5	3	15	903		26	560	27	
5	634		16	17		27	575		
6	7		17	634		28	29		
7	6 3 3		18	16		29	692	693	
8	6		19	20	17	30	214	31	
9	1 0	8	20	15		31	215		
10	6 0 6	605	21	782	22	32	177	30	
11	10	7	22	835		33	211	32	

3.2.5. Ejemplos de trayectorias seleccionadas y su cómputo métrico

En las figuras que se incluyen a continuación se indican en color rojo las rutas óptimas determinadas por el programa y el cómputo de la distancia a recorrer por el vehículo para cada uno de los casos considerados.

En el primer ejemplo, se ubicó al centro de distribución en el nodo determinado por las calles Chiclana esq. Belgrano y se fijó que el despachante deba entregar un pedido en el nodo determinado por la intersección de las calles Tucumán y Bravard.

Con el empleo del algoritmo, el software determinó los caminos más cortos, el de ida (Caso 1, pág. 13) y el de regreso (Caso 2, pág. 14).

En el ejemplo indicado como Caso 3 (pág. 15, se ubicó al centro de abastecimiento sobre la calle Sixto Laspiur a metros de calle Brasil. Los centros o nodos de demanda se situaron en: nodo 1: Estomba esq. Moreno, nodo 2: Las Heras esq. San Martín, nodo 3: Avda. Alem a metros de calle Alsina y nodo 4: Estomba esq. Trelew. Se fijó la secuencia del recorrido en función de prioridades tales como horario de habilitado para distribución (por ej. zona céntrica), horario de recepción del pedido y condiciones de tránsito. Luego de visitar todos los puntos en el orden establecido, el camión retorna al depósito de Sixto Laspiur esq. casi Brasil.

En la figura correspondiente a este caso se indica el recorrido óptimo a realizar por el despachante, una vez determinadas las rutas más cortas entre cada par de puntos, así como también el correspondiente cómputo métrico.

Se resume a continuación la información punto a punto de las rutas óptimas que conforman el recorrido de mínima distancia:

- . Primer camino: **Nodo inicial:** Sixto Laspiur a metros de calle Brasil
 Nodo final: Estomba esq. Moreno
 Distancia a recorrer: 3155,83 mts.

- . Segundo camino: **Nodo inicial:** Estomba esq. Moreno
 Nodo final: Las Heras esq. San Martín
 Distancia a recorrer: 711,92 mts.

- Tercer camino: **Nodo inicial:** Las Heras esq. San Martín
 Nodo final: Av. Alem a metros calle Alsina
 Distancia a recorrer: 756,16 mts.

- . Cuarto camino: **Nodo inicial:** Av. Alem a metros calle Alsina
 Nodo final: Estomba esq. Trelew
 Distancia a recorrer: 2500,14 mts.

- Quinto camino: **Nodo inicial:** Estomba esq. Trelew
 Nodo final: Sixto Laspiur a metros de calle Brasil
 Distancia a recorrer: 2132,25 mts.

4. Conclusiones

- La utilización del modelo propuesto facilita la prestación eficiente de servicios de distribución y servicios puerta a puerta; por lo que en la medida que las empresas incrementen su utilización; les permitirá notables economías que se trasladan a los clientes incrementando las ventajas diferenciales mencionadas anteriormente.

- La implementación de un Algoritmo simple como el de Dijkstra en un Sistema de Información Geográfica (GIS), permite determinar con bastante precisión la ruta óptima entre dos puntos cualesquiera de la ciudad de Bahía Blanca, teniendo en cuenta el sentido de circulación real de calles y avenidas, y giros permitidos de acuerdo con la semaforización de la ciudad.

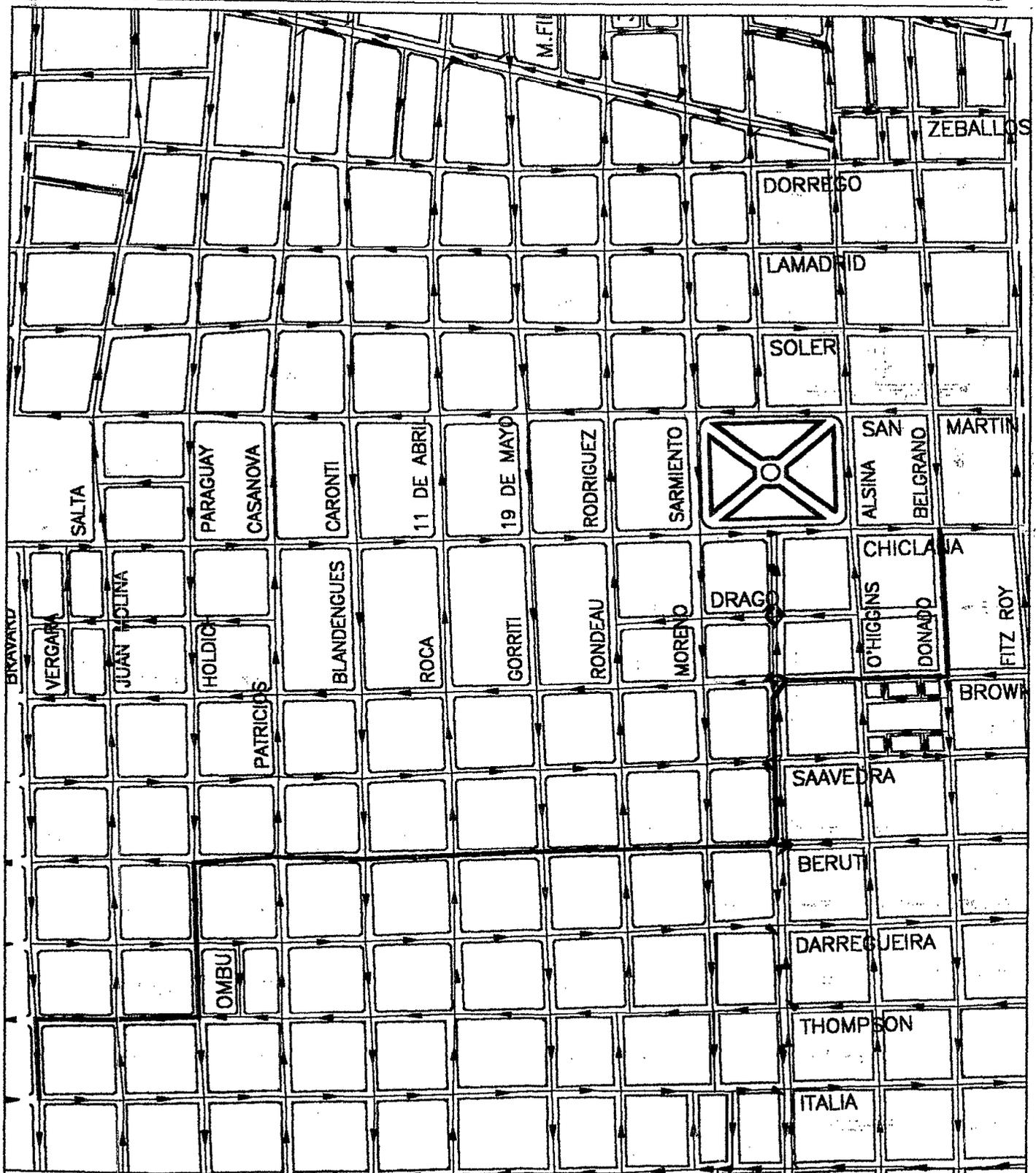
- . La ruta/s resultantes del modelo aseguran mínima distancia y no mínimo tiempo de entrega. Para ello es preciso incorporar al sistema penalizaciones por congestión del tránsito, tarea

prevista para un futuro trabajo. No obstante el aseguramiento de mínima distancia implica consumo mínimo de combustible y menor rodaje de los vehículos.

La modularidad del programa y la capacidad de exportar datos y resultados posibilita en el futuro la implementación de algoritmos más sofisticados que permiten contemplar restricciones tales como: capacidad de los vehículos, puntos intermedios de carga, puntos de carga y descarga, restricciones horarias, restricciones legales, etc.

5. Bibliografía

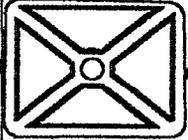
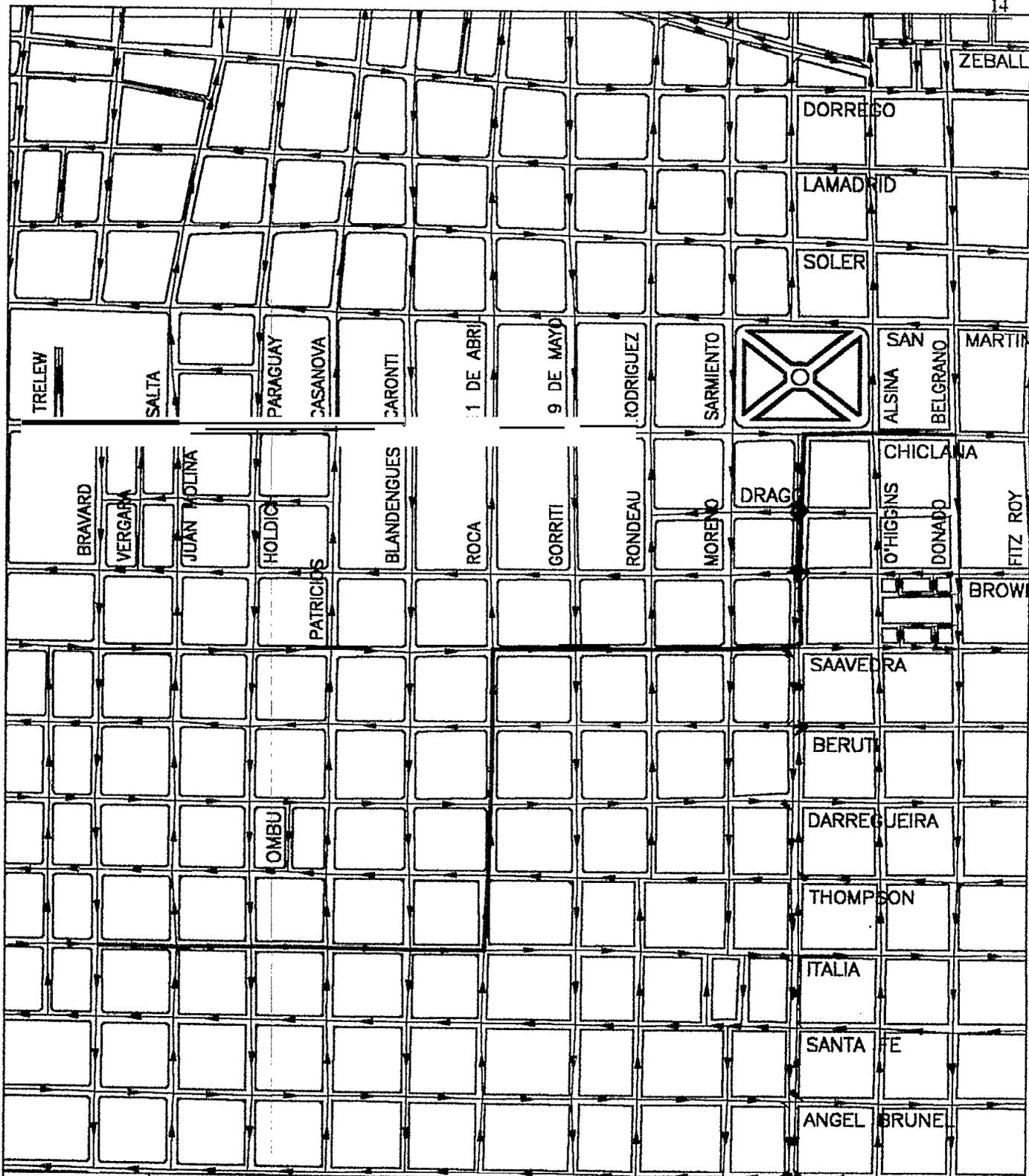
- AHO, A.V., HOPCROFT, J.E. y ULLMAN, J.D.- (1988) *Estructuras de datos y algoritmos*. Addison-Wesley Iberoamericana S.A.
- BALLOU, Ronald (1991) - *Logística empresarial/ - Control y planificación*. Ed. Días de Santos S.A.
- BECKMAN, T.N. y DAVISON, W.R.- *Marketing*. Ronald Press N.Y. 8va. edición
- Boletín informativo de la Asociación Argentina de Logística Empresarial ARLOG
- CHRISTOFIDES, N. y EILON, S. - *Algorithm for vehicle-dispatching problems*. Operational Research Quarterly, vol. 20, N°3
- CHRISTOFER, Martín (1994) - *Logística y aprovisionamiento*, Ed. Folio.
- LA LONDE, B.J. y ZINSZER, P.H. (1976) - *Customer service: Meaning and Measurement*.
- LIGETI, María (1994) - *Corporate reengineering should not stop at the factory door*. Industrial Engineering. Vol. 26, N°8
- LIN, S. y KERNIGHAM, (1973) - *An effective heuristic for the traveling salesman problem*. Operations Research, 21.
- NORMANN, Richard.(1984.) - *Service management: Strategy and leadership in the service business*. Ed. Wiley.
- PERRAULT y RUSS. (Abril 1979) - *Physical distribution service in industrial purchase decisions*. Journal of Marketing.
- POIRIER, David F. y SCOTT SINK, D. (1995) - *Building the distribution system of the future*. Industrial Engineering SOLUTIONS.
- REVISTA de la Asociación Argentina de Logística Empresarial (ARLOG) N° 45 - Junio 1995
- RIGGS, Bruce J (Junio, 1968) - *The traffic manager in physical distribution management*.



norle

CASO Nro: 1
 SELECCION DE RUTA OPTIMA POR
 ALGORITMO DE DIJKSTRA
 FUENTE: AREA ORGANIZACION INDUSTRIAL - UNS

Resumen
 Origen: Chiclana y Belgrano
 Destino: Tucuman y Bravard
 Distancia: 2395.46 mts.



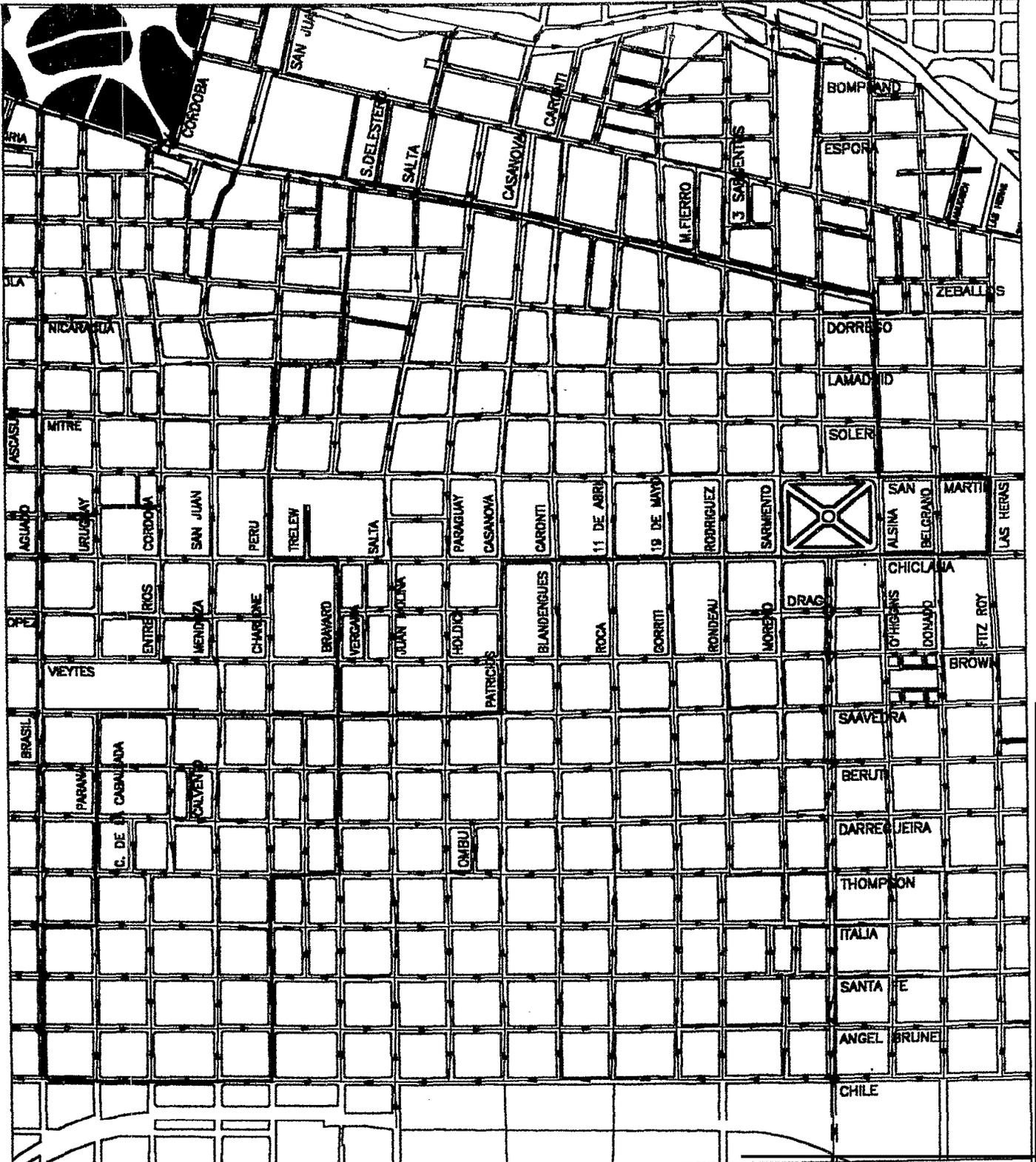
norte

C A S O N r o : 2
 SELECCION DE RUTA OPTIMA POR
 ALGORITMO DE DIJKSTRA

FUENTE: AREA ORGANIZACION INDUSTRIAL - UNS

Resumen

Origen.: Tucuman y Bravard
 Destino: Chiclana y Belgrano
 Distancia: 2392.39 mts.



CASO Nro: 3

SELECCION DE RUTA OPTIMA POR ALGORITMO DE DIJKSTRA

FUENTE: AREA ORGANIZACION INDUSTRIAL - UNS

Resumen

Origen: S.Laspiur-Brasil

Destino: Vrs./ S.Laspiur-Brasil

Distancia: 9256.30 mts.

INGENIERIA DE SOFTWARE / SOFTWARE ENGINEERING

TITULO:	CUANDO LA NORMALIZACION CONDUCE A UN MODELO DE DATOS ILOGICO
AUTOR:	ROSSI BIBIANA, PICA ALEJANDRO; GONZALEZ HECTOR; BALBI, ROMINA.
INSTITUCION:	GRUPO LIDIS. FACULTAD REGIONAL BUENOS AIRES. UNIVERSIDAD TECNOLOGICA NACIONAL.
DIRECCION:	ORTIZ 1222 - (1722) - MERLO - BUENOS AIRES

Bibiana Rossi, Alejandro Pica, Héctor Gouzalet, Romina Balbi

U.T.N. Fac. Regional Buenos Aires

Grupo LIDIS

E-Mail 1: WHEAD@EINSTEIN.COM

E-Mail 2: POSTMASTER@UFAREC.EDU.AR

Domicilio Postal: Ortiz 1222, Merlo C.P. 1722, Buenos Aires, Argentina

INTRODUCCION

Este trabajo enumera los inconvenientes que pueden resultar al tener estructuras de datos en tercera forma normal de acuerdo a las pautas de la bibliografía usualmente consultada para el estudio del tema “Formas Normales”. [Cab 94],[Dat 90],[Dat 95],[Dol 86],[Ken 83],[Mar/S 85],[Smi 85].

El **objetivo de este trabajo** es presentar todas las situaciones problemáticas detectadas en la investigación hasta el momento.

El **presente análisis** se realiza sobre el **modelo conceptual de datos**, no sobre el modelo de implementación que requiere de ajustes según la base de datos a utilizar, las características particulares de velocidad de respuesta y velocidad de procesamiento.

El trabajo consta de dos secciones: FORMAS NORMALES y MODELO ILÓGICO DEL RESULTADO DE LAS FORMAS NORMALES.

En la primer sección se muestra, sobre un ejemplo, los resultados de la aplicación de las formas normales según diferentes autores.

En la segunda sección se enumeran los inconvenientes que ‘presenta el ‘modelo:’ final que será considerado base para la definición de las tablas de una base de datos en una implementación particular.

1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025

FORMAS NORMALES

Se define la estructura de datos Ayudantes de Trabajos Prácticos de Segunda (**ATP2**), considerando una situación particular de relevamiento especificada en las “Aclaraciones”. Los ATP2 son alumnos que se desempeñan como auxiliares docentes en el dictado de alguna materia universitaria. La estructura está expresada según la nomenclatura utilizada por Cabrera. [Cab 94]:

ATP2

Nro facultad regional

Nro legajo ATP2

N o m b r e A T P 2

Fecha nacimiento ATP2

Edad ATP2

Id Asesor Letrado *(1,n)

Cod localidad fac reg

CUIL ATP2

Nro legajo alumno

Cod país nacimiento

Nombre país de nacimiento

Título ATP2 *(1 ,n)

Cod nacional título

Fecha obtención título

Tipo de título

Nombre título

Idioma ATP2 *(1,n)

Te1 particular ATP2 *(0,n)

Cod depto regional

Nom director depto

Te1 interno facultad *(1,n)

Te1 int asignado al depto

Fecha alta línea int

Instalador línea int

Cantidad deptos de facultad

Aclaraciones:

1. Cada facultad tiene un código identificatorio único. No existen dos facultades en una misma localidad.
2. Cada localidad tiene un código identificatorio único.
3. Cada facultad codifica sus departamentos. independientemente de las demás facultades.
4. Todos los departamentos tienen un director. aunque una persona puede ser director de diversos departamentos.
5. Cada facultad asigna distintas líneas telefónicas internas a sus departamentos.
6. Cada teléfono interno de una facultad pertenece a un único departamento.
7. Cada facultad le asigna en forma particular un número de legajo a sus alumnos. Si estos son ayudantes de trabajos prácticos, se les asigna también un número de legajo de ATP2. en forma totalmente independiente de su número de legajo como alumno.
8. Los títulos que puede obtener una persona están codificados a nivel nacional.
9. Los teléfonos particulares de los ATP2 están asignados en forma unívoca. Es decir que no existen diferentes ATP2 con un mismo teléfono.
10. Un ATP2 sólo puede prestar servicios en un único departamento.
11. En cada facultad regional pueden trabajar uno o más asesores letrados identificados unívocamente por un código.

En esta estructura de datos las CLAVES CANDIDATAS son:

- . Nro facultad; regional + Nro legajo ATP2
- . Nro facultad regional + Nro legajo alumno
- Cod localidad fac reg + Nro legajo ATP2
- . Cod localidad fac reg + Nro legajo alumno
- . CUIL ATP2

Partiendo de las claves! candidatas mencionadas que identifican unívocamente y en forma mínima a la estructura **ATP2**, se selecciona como clave principal a la más representativa para el usuario. En este caso, *es Nro facultad regional + Nro legajo A TP2.*

En la estructura de datos se suprime el atributo *Edad* ATP2 ya que no es necesario mantenerlo pues es obtenible mediante la diferencia entre la fecha del día y la *Fecha nacimiento* ATP2. Esta eliminación **no forma** parte de la **1FN**, sino que es un paso previo a la misma y el objetivo es obtener un modelo canónico.[Dat 90],[Dat 95],[Dol86],[Dol 88].

Se normaliza **ATP2** para tener relaciones que cumplan la 1FN, resultando [Cab 94],[Dat 90],[Dat 95],[Dol 86],[Mar/S 85],[Smi 85]:

PRIMERA FORMA NORMAL

<p>ATP2 <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> Nombre ATP2 Fecha nacimiento ATP2 Cod localidad fac reg CUIL ATP2 Nro legajo alumno Cod país nacimiento Nombre país de nacimiento Cod depto regional Nom director depto Cantidad deptos de facultad</p>	<p>TITULO <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Cod nacional título</u> Fecha obtención título Tipo de título Nombre título</p> <p>ASESOR LETRADO <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Id Asesor Letrado</u></p>	<p>IDIOMA <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>I d i o m a A T P 2</u></p> <p>TEL PART ATP2 <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Te1 particular ATP2</u></p> <p>TEL INTERNO FACULTAD <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Te1 int asignado al depto</u> Fecha alta línea int Instalador línea int</p>
--	--	---

Se normalizan las estructuras de datos resultantes de 1FN para tener relaciones que cumplan 2FN resultando [Dat 90],[Dat 95],[Dol 86],[Ken 86],[Mai 93],[Mar/S 85],[Smi 85]:

SEGUNDA FORMA NORMAL

<p>ATP2 <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> Nombre ATP2 Fecha nacimiento ATP2 CUIL ATP2 Nro legajo alumno Cod país nacimiento Nombre país de nacimiento Cod depto regional Nom director depto</p> <p>FACULTAD <u>Nro facultad regional</u> Cod localidad fac reg Cantidad deptos de facultad</p>	<p>TITULO-ATP2 <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Cod nacional título</u> Fecha obtención título</p> <p>TITULO <u>Cod nacional título</u> Tipo de título Nombre título</p> <p>ASESOR LETRADO <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Id Asesor Letrado</u></p>	<p>IDIOMA <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Idioma ATP2</u></p> <p>TEL PART ATP2 <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Te1 particular ATP2</u></p> <p>TEL INTERNO FACULTAD <u>Nro facultad regional</u> <u>Tel int asignado al depto.</u> Fecha alta línea int Instalador línea int</p> <p>TEL INTERNO FACULTAD ATP2 <u>Nro facultad regional</u> <u>Nro legajo ATP2</u> <u>Te1 int asignado al depto</u></p>
---	--	--

Se normalizan las estructuras de datos resultantes de 2FN para tener relaciones que cumplan 3FN, resultando [Dat 90],[Dat 95],[Dol 86],[Ken 86],[Mar/S 85],[Smi 85]:

TERCERA FORMA NORMAL

ATP2

Nro facultad regional
Nro legajo ATP2
CUIL ATP2

ATP2 CUIL

CUIL ATP2
Nombre ATP2
Fecha nacimiento ATP2
Nro legajo alumno
Cod país nacimiento
Cod depto regional
Nom director depto

PAÍS

Cod país nacimiento
Nombre país de nacimiento

FACULTAD

Nro facultad regional
Cod localidad fac reg

DEPTO FACULTAD

Cod localidad fac reg
Cantidad deptos de facultad

TITULO-ATP2

Nro facultad regional
Nro legajo ATP2
Cod nacional título
Fecha obtención título

TITULO

Cod nacional título
Tipo de título
Nombre título

ASESOR LETRADO

Nro facultad regional
Nro legajo ATP2
Id Asesor Letrado

IDIOMA

Nro facultad regional
Nro legajo ATP2
Idioma ATP2

TEL PART ATP2

Nro facultad regional
Nro legajo ATP2
Tel particular ATP2

TEL INTERNO FACULTAD

Nro facultad regional
Tel int asignado al depto
Fecha alta línea int
Instalador línea int

TEL INTERNO FACULTAD ATP2

Nro facultad regional
Nro legajo ATP2
Tel int asignado al depto

MODELO ILÓGICO DEL RESULTADO DE LAS FORMAS NORMALES

En base a las estructuras de datos resultantes de aplicar 3FN se presentando los siguientes problemas:

1.- La estructura

TEL PART ATP2
Nro facultad regional
Nro legajo ATP2
Tel particular ATP2

que se obtuvo como resultado de aplicar 1FN y que cumple con 2FN y 3FN, tiene definida una clave **no mínima**, pues con solo conocer el *Tel particular ATP2* se puede saber de que **ATP2 se** está hablando (ver aclaración 9 de página 2). Es decir que se puede eliminar de la clave, pero no de la estructura de datos, a los atributos *Nro facultad regional* y *Nro legajo ATP2* sin destruir la unicidad de la misma. [Dat 92],[Dat 95],[Mar/S 85].

2.- La estructura

ASESOR LETRADO
Nro facultad regional
Nro legajo ATP2
Id Asesor Letrado

ASESOR LETRADO
Nro facultad regional
Nro legajo ATP2
Id Asesor Letrado

que se obtuvo como resultado de aplicar 1FN y que cumple con 2FN y 3FN

- a) **Define** una “idea o entidad inexistente. para el sistema. Pues los asesores letrados, no se relacionan en forma directa con los ATP2 sino a través de la facultad en la que trabajan. Este tipo de anomalías no es considerada por ninguna de las formas normales conocidas.
- b) Los *Id Asesor Letrado se* repiten por cada uno de los ATP2 que prestan servicios en esa facultad. Es decir, que si hay 325 ATP2 en una facultad, se detallan los asesores letrados, 324 veces de más.
- c) Si se modifica la estructura original reemplazando el atributo repetitivo *Id Asesor Letrado* por el planteo que cada facultad tiene un único asesor (estructura modificada), aplicando 1FN, 2FN y 3FN se obtiene un conjunto diferente de estructuras normalizadas lo cual no es lógico porque el cambio de cardinalidad de un atributo no debería modificar la idea que representa; ni de quién depende:

Opción 1**ATP2 - Estructura original**Nro facultad regional

Nombre ATP2

Fecha nacimiento ATP2

Edad ATP2

Id Asesor Letrado * (1 .n)

Cod localidad fac reg

CUIL ATP2

Nro legajo alumno

Cod país nacimiento

Nombre país de nacimiento

Título ATP2 *(1 .n)

Cod nacional título

Fecha obtención título

Tipo de título

Nombre título

Idioma ATP2 *(1 n)

Tel particular ATP2 *(0-n)

Cod depto regional

Nom director depto

Tel interno facultad *(1 .n)

Tel int asignado al depto

Fecha alta línea int

Instalador línea int

Cantidad deptos de facultad

Opción 2**ATP2 - Estructura modificada**Nro facultad regionalNro legajo ATP2

Nombre ATP2

Fecha nacimiento ATP2

Edad ATP2

Id Asesor Letrado

Cod localidad fac reg

CUIL ATP2

Nro legajo alumno

Cod país nacimiento

Nombre país de nacimiento

Título ATP2 *(1 .n)

Cod nacional título

Fecha obtención título

Tipo de título

Nombre título

Idioma ATP2 *(1 .n)

Tel particular ATP2 *(0.n)

Cod depto regional

Nom director depto

Tel interno facultad *(1.n)

Tel int asignado al depto

Fecha alta línea int

Instalador línea int

Cantidad deptos de facultad

En el ejemplo, se observa que las estructuras difieren en el conjunto de atributos que determina a *Id Asesor Letrado*. Esto no tiene sentido ya que la cardinalidad de un atributo no debería afectar al identificador del mismo.

Tercera Forma Normal - Opción 1**ASESOR LETRADO**Nro facultad regionalNro legajo ATP2Id Asesor Letrado

DEPTO FACULTAD

Cod localidad fac reg

ATP2

Ídem versión original (pág. 4)

ATP2 CUIL

Ídem versión original (pág. 4)

PAIS

ídem versión original (pág. 4)

FACULTAD

Ídem versión original (pág. 4)

TITULO-ATP2

Ídem versión original 1 (pág. 4)

Tercera Forma Normal - Opción 2

DEPTO ASESOR FACULTAD

Cod localidad fac reg

Cantidad deptos de facultad

Id Asesor Letrado

ATP2

Ídem versión original (pág. 4)

ATP2 CUIL

Ídem versión original (pág. 4)

PAIS

Ídem versión original (pág. 4)

FACULTAD

Ídem versión original (pág. 4)

TITULO-ATP2

Ídem versión original (pág. 4)

TITULO

Ídem versión original (pág. 4)

IDIOMA

Ídem versión original (pág. 4)

TEL PART ATP2

Ídem versión original (pág. 4)

TEL INTERNO FACULTAD

Ídem versión original (pág. 4)

TEL INTERNO FACULTAD ATP2

Ídem versión original (pág. 4)

TITULO

Ídem versión original (pág. 4)

IDIOMA

Ídem versión original (pág. 4)

TEL PART ATP2

Ídem versión original (pág. 4)

TEL INTERNO FACULTAD

Ídem versión original (pág. 4)

TEL INTERNO FACULTAD ATP2

Ídem versión original (pág. 4)

3.- La estructura

TEL INTERNO FACULTAD ATP2Nro facultad regionalNro legajo ATP2Tel int asignado al dedo

que se obtuvo como resultado de aplicar 1FN y que cumple con 2FN y 3FN

- a) Define una idea o entidad inexistente para el sistema. Pues los teléfonos internos de un departamento, no se relacionan en forma directa con el ATP2 sino a través del departamento en el que trabaja. Este tipo de anomalías no es considerada por ninguna de las formas normales conocidas. Tal es así que las estructuras de datos **TEL INTERNO FACULTAD ATP2** y **TEL INTERNO FACULTAD** nacen a partir de una misma estructura de datos, pero la primera no logra una buena definición de la realidad del sistema, mientras que no sucede lo mismo con la segunda estructura de datos como vemos a continuación:

TEL INTERNO FACULTADNro facultad regionalTel int asignado al depto

Fecha alta línea int

Instalador línea int

- b) Los *Tel int asignado al depto* se repiten para cada uno de los ATP2 que prestan servicios' en ese departamento. Es decir, que si hay 35 ATP2 en un departamento, se detallan, los teléfonos internos del departamento correspondiente, **34** veces de más.
- c) Si se modifica la estructura original reemplazando el grupo repetitivo *Tel interno facultad* por el planteo que cada departamento tiene un único teléfono (estructura modificada), aplicando la 1FN, 2FN y 3FN se obtiene un conjunto diferente de estructuras normalizadas lo cual no es lógico porque el cambio de cardinalidad de un atributo no debería modificar la idea que representa; ni de quién depende:

Opción 1**ATP2 - Estructura original**

Nro facultad regional
Nro legajo ATP2
 Nombre ATP2
 Fecha nacimiento ATP2
 Edad ATP2
 Id Asesor Letrado * (1 .n)
 Cod localidad fac reg
 CUIL ATP2
 Nro legajo alumno
 Cod país nacimiento
 Nombre país de nacimiento
 Título ATP2 *(1 .n)
 Cod nacional título
 Fecha obtención título
 Tipo de título
 Nombre título
 Idioma ATP2 *(1.n)
 Tel1 particular ATP2. *(0.n)
 Cod depto regional
 Nom director depto
 Tel interno facultad *(1 ,n)
 Tel int asignado al depto
 Fecha alta línea int
 Instalador línea int
 Cantidad deptos de facultad

Opción 2**ATP2 - Estructura modificada**

Nro facultad regional
Nro legajo ATP2
 Nombre ATP2
 Fecha nacimiento ATP2
 Edad ATP2
 Id Asesor Letrado * (1 .n)
 Cod localidad fac reg
 CUIL ATP2
 Nro legajo alumno
 Cod país nacimiento
 Nombre país de nacimiento
 Título ATP2 *(1 .n)
 Cod nacional título
 Fecha obtención título
 Tipo de título
 Nombre título
 Idioma ATP2 *(1 n)
 Tel1 particular ATP2 *(0.n)
 Cod depto regional
 Nom director depto
 Tel int asignado al depto
 Fecha alta línea int
 Instalador línea int
 Cantidad deptos de facultad

Tercera Forma Normal - Opción 1**ATP2 CUIL**

CUIL ATP2
 Nombre ATP2
 Fecha nacimiento ATP2
 Nro legajo alumno
 Cod país nacimiento
 Cod depto regional
 Nom director depto

TEL INTERNO FACULTAD

Nro facultad regional
Tel int asignado al depto
 Fecha alta línea int
 Instalador línea int

TEL INTERNO FACULTAD ATP2

Nro facultad regional
Nro legajo ATP2
Tel int asignado al depto

Tercera Forma Normal - Opción 2**ATP2 CUIL**

CUIL ATP2
 Nombre ATP2
 Fecha nacimiento ATP2
 Nro legajo alumno
 Cod país nacimiento
 Cod depto regional
 Nom director depto
 Tel int asignado al depto
 Fecha alta línea int
 Instalador línea int

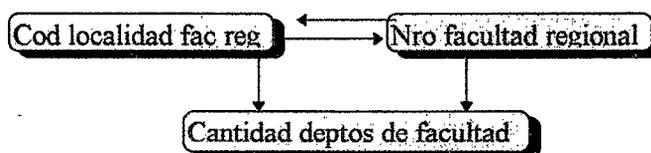
En el ejemplo se observa que las estructuras difieren en el conjunto de atributos que determina a *Fecha alta línea int e Instalador línea int*. Esto no tiene sentido ya que la cardinalidad de un conjunto de atributos no debería afectar al identificador de los mismos.

ATP2
 Ídem versión original (pág. 6)
PAÍS
 Ídem versión original (pág. 6)
FACULTAD
 Ídem versión original (pág. 6)
DEPTO FACULTAD
 Ídem versión original (pág. 6)
TITULO
 Ídem versión original (pág. 6)
TITULO ATP2
 Ídem versión original (pág. 6)
IDIOMA
 Ídem versión original (pág. 6)
TEL PART ATP2
 Ídem versión original (pág. 6)
ASESOR LETRADO
 Ídem versión original (pág. 6)

ATP2
 Ídem versión original (pág. 6)
PAÍS
 ídem versión original (pág. 6)
FACULTAD
 Ídem versión original (pág. 6)
DEPTO FACULTAD
 Ídem versión original (pág. 6)
TITULO
 Ídem versión original (pág. 6)
TITULO ATP2
 Ídem versión original (pág. 6)
IDIOMA
 Ídem versión original (pág. 6)
TEL PART ATP2
 Ídem versión original (pág. 6)
ASESOR LETRADO
 Ídem versión original (pág. 6)

4.- Si consideramos la estructura de datos FACULTAD resultante de la 2FN, junto a' su diagrama de dependencias funcionales, obtenemos:

FACULTAD
Nro facultad regional
 Cod localidad fac reg
 Cantidad deptos de facultad



En la bibliografía citada no se explícita en forma clara que ante una estructura de datos del estilo de **FACULTAD** se deba crear otras. En realidad no se aconseja hacerlo, pero **tampoco se consideran ejemplos de este estilo**. Algunos autores, tales como Cabrera [Cab 94], Kent [Ken 83] y Dolder [Do1 86] sólo' plantean estructuras de datos con **una única clave candidata** sin aclarar el porqué, o bien otros autores tales como Date [Dat 90] y [Dat 95] aclaran que lo hacen *por simplicidad*, basándose en el argumento de la baja aparición de estructuras con más de una clave candidata.

Ante la falta de definición de este tipo de casos, si se quiere enfrentar a una estructura como **FACULTAD** con dichas definiciones, según se tome una u otra se puede arribar a distintas conclusiones.

4.1- Date define [Dat 90] :

Una relación está en 3FN si y sólo si los atributos no-claves son:

a) *mutuamente independientes , y*

b) *completamente dependientes de la clave primaria.*

. Un atributo no-clave es un atributo que no participa en la clave primaria de fa relación considerada.

. Dos o más atributos son mutuamente independientes si ninguno de ellos es funcionalmente dependiente de cualquier combinación de Los demás. Tal independencia implica que cada uno de tules atributos pueden actualizarse independientemente del resto.

Por lo que en la estructura de datos **FACULTAD** se ve que no hay una independencia mutua entre los atributos no-claves, por lo que dicha estructura no cumpliría 3FN. Debido a esto, al aplicar 3FN obtendríamos:

FACULTAD
Nro facultad regional
 Cod localidad fac reg

DEPTO FACULTAD
Cod localidad fac reg
 Cantidad deptos de facultad

4.2- Dolder [Do1 86] dice que una relación está en 3FN si se cumple la *no existencia de dependencias' transitivas (atributos dependientes que son funcionalmente dependientes de otros atributos dependientes)*; con lo que se podría alegar (o no), que no existe una dependencia transitiva de *Cantidad deptos facultad* respecto de *Nro facultad regional* a través de *Cod localidad fac reg* pues *Cantidad deptos facultad* depende 'en forma directa" de *Nro facultad regional* , por lo que la estructura **FACULTAD** podría estar en 3FN (o no). Esto implica que se podrían presentar las siguientes dos opciones:

Tercera Forma Normal - Opción 1:

FACULTAD
Nro facultad regional
 Cod localidad fac reg
 Cantidad depto: de facultad

Tercera Forma Normal - Opción 2:

FACULTAD
Nro facultad regional
 Cod localidad fac reg

DEPTO FACULTAD
Cod localidad fac reg
 Cantidad deptos de facultad

4.3- Martin [MarS 85], da la siguiente definición:

Supongamos que *A*, *B* y *C* son tres atributos o tres colecciones de atributos de una relación *R*. Si *C* es funcionalmente dependiente de *B* y *B* lo es de *A* entonces *C* es funcionalmente dependiente de *A*. Si la correspondencia inversa no es simple, esto es, si *A* no es funcionalmente dependiente de *B* o *B* no es funcionalmente dependiente de *C*, se dice que *C* es transitivamente dependiente de *A*.

Y especifica que una estructura de datos está en 3FN si está en 2FN y además no hay atributos, que no pertenezcan a la clave principal, que tengan una dependencia transitiva respecto claves candidatas.

En el ejemplo que se considera, según la definición de Martin, existe una dependencia transitiva.

Analizándolo se deduce que *Nro facultad regional*, *Cod localidad fac reg* y *Cantidad deptos de facultad* son tres atributos de la relación **FACULTAD**. Como *Cantidad deptos de facultad* es funcionalmente dependiente de *Cod localidad fac reg* (por aclaraciones 1 y 2 de página 2) y *Cod localidad fac reg* lo es de *Nro facultad regional* entonces *Cantidad deptos de facultad* es funcionalmente dependiente de *Nro facultad regional*.

Al analizar la correspondencia inversa simple notamos que Nro facultad regional es funcionalmente dependiente 'de *Cod localidad fac reg* (ver aclaraciones 1 y 2 de página 2), pero como *Cod localidad fac reg* no es funcionalmente dependiente de *Cantidad deptos de facultad* podemos afirmar que *Cantidad deptos de facultad* ES transitivamente dependiente de *Nro facultad regional*.

La definición de Martin llevaría a la conclusión inequívoca de que NO cumple con 3FN la estructura de datos FACULTAD:

FACULTAD
Nro facultad regional
 Cod localidad fac reg

DEPTO FACULTAD
Cod localidad fac reg
 Cantidad deptos de facultad

Como se ve, la bibliografía presenta criterios disímiles de análisis de una misma situación dejando interrogantes como:

- ¿La estructura FACULTAD resultante de 2FN cumple con 3FN?.
- ¿Existe una dependencia transitiva de *Cantidad deptos de facultad* respecto de *Nro facultad regional* a través del atributo *Cod localidad fac reg*?
- ¿No será que la división de la estructura citada en FACULTAD y DEPTO FACULTAD nacen a partir de una confusa interpretación de la 3FN pues los atributos de ambas forman parte de una misma entidad, a la que describen, y desde el punto de vista lógico no tiene sentido separarla en dos?.

:

- 5.- Las estructuras de datos ATP2 y ATP2 CUIL resultantes de 3FN son una variante de la situación anterior, presentando el mismo problema descripto. Aquí, hay una mayor cantidad de atributos y no por ello dejan de presentar el mismo problema conceptual mencionado1

Estructura Original (de 2FN)

ATP2

Nro facultad regional

Nro legajo ATP2

Nombre ATP2

Fecha nacimiento ATP2

CUIL ATP2

Nro legajo alumno

Cod país nacimiento

Nombre país de nacimiento

Cod depto regional

Nom director depto

Tercera forma normal - opción 1:

ATP2
Nro facultad regional
Nro legajo ATP2
CUIL ATP2

ATP2 CUIL
CUIL ATP2
Nombre ATP2
Fecha nacimiento ATP2
Nro legajo alumno
Cod país nacimiento
Cod depto regional
Nom director depto

Tercera forma normal - Opción 2:

ATP2
Nro facultad regional
Nro legajo ATP2
Nombre ATP2
Fecha nacimiento ATP2
CUIL ATP2
Nro legajo alumno
Cod país nacimiento
Cod depto regional
Nom director depto

PAÍS
Cod País nacimiento
Nombre país de nacimiento

6.- Si aplicamos 3FN en la estructura de datos **ATP2** resultante de 2FN tenemos las siguientes dos opciones:

Opción 1: notamos un nuevo problema. En la estructura **ATP2 CUIL** se incluye la idea de los departamentos de la facultad, pero dada la forma de dicha estructura nos veremos imposibilitados de aislar un conjunto de atributos que identifique a la entidad departamento. Por la aclaración 3 de página 2, un departamento está identificado por *Nro facultad regional* + *Cod depto regional*, o sea que estos dos atributos determinan al *Nom director depto*. Pero vemos que *Nro facultad regional* no forma parte de **ATP2 CUIL** por lo que no podremos definir un identificador diferente a la clave primaria que determine a *Nom director depto*. Esta falencia nace en el momento de dividir la estructura de datos ATP2 resultante de 2FN y nos provoca la pérdida de una relación en el modelo de datos.

Nótese que si bien se pierde dicha relación, no perdemos la información de cuál es el departamento en el que trabaja cada ATP2 pues es obtenible mediante un simple algoritmo de búsqueda que involucre las estructuras ATP2 y ATP2 CUIL. Además de que esta estructura no representa en forma óptima el sistema, encontramos un segundo problema que es el de repetir el nombre del director de un departamento por cada ATP2 adicional que trabaje en el mis

Opción 2: el problema que se presenta es tener en la estructura los datos identificatorios del departamento de la facultad en que trabaja el ayudante. Esta estructura que arroja la normalización clásica nos indica que si tenemos 60 ayudantes en un departamento, tendremos 60 veces la información de quien es el director, es decir 59 veces de más.

Nótese que *Cod depto regional* no determina al *Nom director depto* (ver aclaración 3 de página 2) sino que (*Nro facultad regional* + *Cod depto regional*)-*Nom director depto*, por lo que *Cod depto regional* es un atributo no-clave, y *Nro facultad regional* un atributo clave y a causa de esto queda excluido en el campo de las dependencias entre atributos no-claves que se analiza en 3FN.[Dat 90],[Dat 95],[Ken 83].

Un punto particular es que si bien ambas opciones representan al sistema, no logran hacerlo en forma óptima, y hay redundancia! de datos. Es decir que aunque la opción 1 es diferente a la opción 2, se

CONCLUSION

Las Estructuras de Datos resultantes de la 3FN según la bibliografía usualmente utilizada pueden dejar como resultado un modelo ilógico de datos.

Algunos de los problemas enumerados podrían solucionarse aplicando otras formas normales tales como Boyce and Codd, 4FN propuesta por Dolder, 4FN según Fagin, 5FN, 6FN, etc. Otros de los problemas presentados pueden solucionarse realizando modificaciones a la 1FN, 2FN y 3FN Estas propuestas serán la base de futuras publicaciones.

Suele darse como argumento, que la aparición de estas situaciones es de “baja” frecuencia y por lo tanto no es necesario mejorar el proceso de normalización, para lograr estructuras óptimas.[Cab 94],[Dat 90],[Wu 92].

Se debiera hacer un análisis más exhaustivo, para determinar el grado de conveniencia, considerando al menos:

- ¿Cuánto es el porcentual de “baja” frecuencia?
- Si alguna de’ estas situaciones problemáticas aparece, ¿Cuál es el costo de los inconvenientes’ que causan?
- Si el sistema debe ser preciso (control de un proceso industrial, edificios inteligentes, control de viajes espaciales, sistemas de seguridad), estas situaciones problemáticas, ¿afectan la precisión?, ¿cuánto?

Es claro que una vez definido el costo del problema, debe compararse contra el costo de la solución. La idea de la “poca aparición” por sí sola resulta un argumento incompleto. Para poder realizar este análisis es imprescindible conocer cuáles son las situaciones problemáticas y cuáles sus consecuencias, con el fin de poder evaluar los riesgos.

Según Date, en el momento de la definición de la base de datos, el diseñador resuelve estos casos en forma práctica y no llegan a producirse estas situaciones porque se remedian empíricamente.[Dat 90].

Pensar en el desarrollo de software que facilite la tarea de diseño, no es un problema menor. Para ello es necesario el planteo del tema desde el punto de vista del modelo teórico que permita definir el algoritmo correspondiente, y es allí donde el modelo es aplicable parcialmente.

Se desconoce la existencia de un software que considere las situaciones planteadas en el presente trabajo. Un modelo teórico más completo puede servir de base para el desarrollo de un software que le deje al diseñador la elección del nivel de normalización al que desea llegar y que cuantifique y cualifique los riesgos en cada uno de esos niveles, facilitando el diseño de una base de datos.

NOTA: El grupo LIDIS desea recibir aportes y sugerencias de quienes lean el presente trabajo con el fin de intercambiar opiniones sobre los conceptos expresados. Para ello pueden remitirse al domicilio postal o a los e-mail que figuran en la primer página.

BIBLIOGRAFÍA

1. [Cab 94].Cabrera. Jaime. *Profesión Sistemas*. Tomo 1. 3° Edición. - 1994
2. [Dat 90].Date. C.J. *An Introduction to Databases Systems*. 5° Edición. Addison-Wesley Publishing Compaq. - 1990
3. [Dat 92].Date. C.J. y Fagin, Ronald. *Simple Conditions for Guaranteeing Higher Normal Forms in Relational Databases*. ACM Transactions on Databases Systems. Vol. 17. Nro.3 - Septiembre de 1992.
4. [Dat 95].Date. C.J. ,-In *Introduction to Databases Systems*. 6° Edición. Addison-Wesley Publishing Company. - 1993
5. [Die 88].Diederich. Jim y Milton. Jack. *New Methods and Fast Algorithms for Database Normalization*. ACM Transactions on Databases Systems. Vol. 13. Nro. - Septiembre de 1988.
6. [Dol 86].Dolder. Herman E. *Diseño Conceptual e Implementación de Bases de Datos*. DATA S.A. 2 ° Edición. Buenos Aires - 1986.
7. [Dol 88].Dolder. Herman E. y Lubomirsky. Esteban. *DBAID. Sistema Experto para el Diseño de Bases de Datos*. DATA S.A. Buenos Aires - 1988.
8. [Her 9 1].Hernandez. Héctor J. y Chan. Edward P.F. *Constant-Time-Maintenable BCNF Databases Schemes*. ACM Transactions on Databases Systems. Vol. 16. Nro. - Diciembre de 1991.
9. [Ken 83].Kent, W. *A Simple Guide to Five normal Forms in Relational Database Theory*. Communications of the ACM 26.2 (Febrero. 1983). Computer Science Press, Rockville 1993.
10. [Mai 93].Maier. D. *The Theory of Relational Databases*. Prentice Hall International Editorial.
11. [Mar/0 85].Martin, James. *Organización de las Bases de Datos*. Prentice Hall Hispanoamericana - 1985
12. [Mar/S 85].Martin. James.*System Design From Provably Correct constructs*. Prentice Hall International - 1985.
13. [OZS 87].oZsoyoglu. Z. Meral y Li-Yan Yuan. *A New Form for Nested Relations*. ACM Transactions on Databases Systems. Vol.12. Nro. 1 - Marzo de 1987.
14. [Rot 87].Roth. Mark A. y Korth Henry F. *The Design 1NF Relational Databases into Nested Normal Form*. Proceedings of Association for Computer Machines. Special Interest Group on Management. Vol. 16. Nro. 3- Diciembre de 1987.
15. [Smi 85]. Smith. Henry C. *Database Design: Composing Fully Normalized Tables From a Rigorous Dependency Diagram*. Communications of the ACM. Vol.28. Nro. 8- Agosto de 1985.
16. [Ul] 82] Ullman. Jeffrey D. *Principles of Database Systems*. 2° Edición. Computer Science-Press - 1982.
17. [Wu 92]. Wu. Margaret S. *The Practical Need for Fourth Normal Form*. ACM PREST SIGCSE BULLETIN. Vol.24. Nro. 1 - Marzo de 1992.
18. [Yeh 83].Yehoshua. Sagiv. *A Characterization of Globally Consistent Databases and Their Correct Access Paths*. ACM Transactions on Databases Systems. Vol.8. Nro. 2- Junio de 1983.

HERRAMIENTAS / TOOLS

TITULO:	IGNATIUS: UNA HERRAMIENTA PARA CONSTRUIR SISTEMAS DE SUPERVISION DE PROCESOS INDUSTRIALES
AUTOR:	BENITEZ, SILVIA; SEOANE, JUAN; WAINER, GABRIEL; BEVILACQUA, ROBERTO.
INSTITUCION:	DEPARTAMENTO DE COMPUTACION, FACULTAD DE CIENCIAS EXACTAS Y NATURALES. UNIVERSIDAD DE BUENOS AIRES
DIRECCION:	CIUDAD UNIVERSITARIA - PABELLON I - BUENOS AIRES

TITULO:	ALFANEGO: SISTEMA DE RELEVAMIENTO, ANALISIS Y RECOMENDACION DE ACTIVIDADES AGROPECUARIAS
AUTOR:	UZAL, ROBERTO; LUCERO, MARIO; SANCHEZ, ALBERTO; DASSO, ARISTIDES; MONTEJANO, GERMAN.
INSTITUCION:	DEPARTAMENTO DE INFORMATICA UNIVERSIDAD NACIONAL DE SAN LUIS
DIRECCION:	EJ. DE LOS ANDES 950 - 1 ER. PISO - (5700) - SAN LUIS - ARGENTINA

TITULO:	CONTROL ESTADISTICO DE MATERIALES
AUTOR:	CARTAGENOVA, SANDRA; LISNIZER, MARIO.
INSTITUCION:	UNIVERSIDAD F.A.S.T.A. (1) - PHARMA ARGENTINA
DIRECCION:	(1)GASCON 3145 - (7600) - MAR DEL PLATA - BUENOS AIRES

IGNATIUS: una Herramienta para Construir' Sistemas de Supervisión de Procesos industriales

Silvia V. Benitez
silvia.v.benitez@ac.com

Juan J. Seoane
seoane@vnet.ibm.com

Gabriel A. Wainer
gabrielwQdc.uba.ar

Roberto J.G.,Bevilacqua
robevi@dq.uba.ar

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Paballón I - Ciudad Universitaria
Buenos Aires - Argentina
FAX: 783-0729

ABSTRACT

En este trabajo se presenta el diseño e implementación de una herramienta para construir sistemas de supervisión de procesos llamada IGNATIUS. Se propone una descomposición de los sistemas de supervisión en tres niveles de servicio; ISL, PSL y BSL. Los servicios de ISL y PSL varían con cada implementación, pero el BSL incluye los servicios comunes a todos los supervisores. IGNATIUS encapsula estos servicios. La descomposición propuesta limita el esfuerzo de construir sistema de supervisión, reduciendo el ciclo de desarrollo, reduciendo la complejidad y dejando que el usuario se concentre en aspectos de diseño de alto nivel. IGNATIUS provee un entorno de desarrollo que puede ser usado por programadores inexpertos, ayudando a mantener la integridad del sistema, hacer cambios y desarrollar nuevas aplicaciones. Esta aproximación permite construir aplicaciones SCADA complejas reduciendo el costo de desarrollo y mantenimiento.

Palabras Clave: sistemas de supervisión, SCADA, sistemas de tiempo real, herramientas de desarrollo de software.

1. INTRODUCCION

En la actualidad, **los avances y reducción** en los costos de hardware han permitido aumentar el número de computadoras usadas para controlar procesos físicos. Esto condujo a variedad de automatización de procesos donde era necesaria la intervención humana (por ejemplo, operación automática de fábricas, monitoreo y control de aeronaves, control de procesos químicos, control de luces de tráfico en una ciudad, etc.). Esta clase de procesos precisa respuesta en tiempo real.

Un sistema de tiempo real es aquel en el cual los resultados no sólo dependen de la correctitud lógica de los cálculos, sino también del momento en que esos resultados se generan. Si las restricciones de tiempo real no se cumplen, ocurre una falla del sistema que puede provocar consecuencias catastróficas. Luego, en esta clase de sistemas es esencial garantizar que las restricciones de tiempo serán cumplidas [Wai94].

Hay un gran número de aplicaciones en tiempo real en la industria, y un número creciente, de computadoras están controlando procesos industriales. El control de procesos puede definirse como “el ejercicio de una acción planificada, para lo que se considera un objeto satisfaga objetivos particulares” [Miy93]. Un controlador acepta la información que viene de sensores, que se procesa, y sus resultados se envían a un objeto controlado a través de actuadores. La información de salida afecta al objeto controlado.

En muchos casos ocurren condiciones de excepción, y los objetivos del sistema de control no pueden cumplirse. Para prevenir estas condiciones, asegurando una respuesta correcta del sistema, se puede usar un conjunto de programas, llamados Sistemas de Supervisión o SCADA (Supervisory Control and Data Acquisition). Para cumplir con su función, la computadora no sólo debe controlar la operación de la planta, sino también proveer a los ingenieros y gerentes de la planta con una fotografía del estado de los procesos.

Los programas de supervisión están diseñados para coordinar, monitorear y soportar los componentes del sistema. Manejan entrada y salida de mensajes y datos, planifican el flujo e ejecución, otorgan prioridades a los programas de aplicación, y llevan a cabo funciones de mantenimiento. También pueden procesar interrupciones y manejar condiciones de error y emergencia. deben estar diseñados para coordinar las funciones del sistema bajo diversidad de cargas.

Aunque diversidad de cambios han ocurrido en el área de control de procesos, la mayoría de las aplicaciones SCADA no son lo suficientemente flexibles, y tienen un rango de uso muy limitado. Con este escenario en mente, nuestra propuesta trata el diseño e implementación de una herramienta para construir aplicaciones SCADA. Para ello se estudiaron diversas técnicas semiformales de diseño en tiempo real, y la herramienta se ha adaptado a una técnica seleccionada. También se ha seleccionado el software de base para proveer un entorno de desarrollo completo que pueda ser usado por programadores, logrando seguridad, facilidad de mantenimiento y correctitud de las aplicaciones desarrolladas. La herramienta ha sido diseñada para permitir que el usuario construya una variedad de aplicaciones de supervisión con mínimo esfuerzo.

Este trabajo está organizado de la siguiente forma: en la sección 2 se describe una aproximación para construir sistemas SCADA; en la sección 3 se hace una descripción de los componentes de la herramienta, y finalmente en la sección 4 se muestra el uso de la herramienta para construir aplicaciones SCADA.

2. UNA APROXIMACION A LA CONSTRUCCION DE SISTEMAS SCADA

Luego de analizar diversos sistemas SCADA disponibles en la actualidad, se pudieron identificar tres niveles de servicio comunes a ellos:

1; Nivel de Servicio de Interfaz (Interface Service Level ISL): este nivel encapsula todos los programas que proveen la interfaz entre el SCADA y el operador. Provee pantallas gráficas, mímicos, alarmas sonoras, mensajes, etc. La implementación de este conjunto de programas cambia e acuerdo con las plataformas de hardware y software elegidas para el desarrollo.

2, Nivel de Servicio Particular (Particular Service Level - PSL): este nivel encapsula todos los programas que implementan los requerimientos particulares para un SCADA específico, por ejemplo, rutinas de asistencia de alarmas, manejadores de eventos, rutinas de respuesta a comandos de usuario, etc. La implementación de este conjunto de programas cambia para cada SCADA de acuerdo con las características de servicios particulares.

3. Nivel de Servicio Básico (Basic Service Level - BSL): este nivel encapsula todos los programas que implementan los servicios de supervisión básicos que son comunes a todos los SCADA, por ejemplo, representación interna de puntos de la planta, rutinas de detección. de alarmas, almacenamiento histórico, transmisión de mensajes, ejecución de comandos de usuario, planificación de procesos de alto nivel, modelado' de procesos del mundo real, etc. Desde un punto de vista funcional, este nivel de servicio es el mismo para todo SCADA.

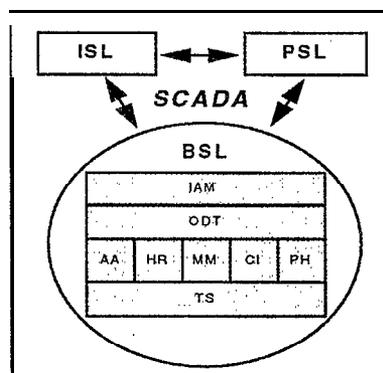


Figura 1. Descomposición de niveles de servicios de un SCADA

IGNATIUS trata de evitar el desarrollo de los servicios. BSL (comunes a todos los SCADA), minimizando de esta forma la tarea de programación. De acuerdo con la clasificación propuesta en [Kap95], esta aproximación actúa como una interfaz con un lenguaje de programación, lo que permite gran flexibilidad ya que es una biblioteca que puede adaptarse para controlar cualquier proceso del mundo real. No es una herramienta orientada al usuario final, sino que se

trata que sea usada por programadores que precisan desarrollar aplicaciones de supervisión específica con un mínimo esfuerzo.

La herramienta fue diseñada para ser acoplada con la mayoría de las técnicas de diseño semiformales para tiempo real existentes, pero está especialmente adaptada para ser combinada con MASCOT [Mas87]. A su vez, se analizaron diversos sistemas operativos existentes, y fue elegido OS/2. La biblioteca de clases fue construida usando C++ y Vispro/C++ como herramienta gráfica (para mayor información acerca de estas decisiones, ver [Ben96a], [Ben96b]).

La herramienta fue implementada como una biblioteca de clases usando el modelo Cliente/Servidor, donde IGNATIUS actúa como un servidor, y los niveles ISL y PSL son los clientes que varían para cada implementación particular. Luego, IGNATIUS puede verse como el mismo BSL que provee servicios a distintos clientes ISL y PSL.

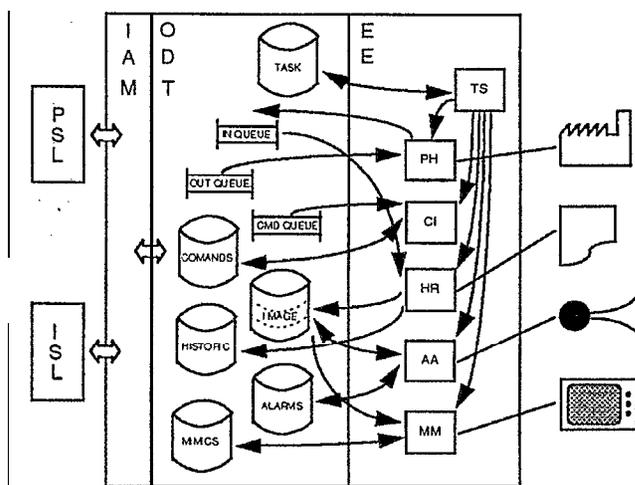


Figura 2 - Interacción de componentes de la herramienta

IGNATIUS provee tres servicios básicos a través de sus clases:

- Tablas de datos (Object Data Tables - ODT): es un conjunto de tablas que almacenan información acerca de los distintos objetos del mundo real, incluyendo la imagen de la planta, alarmas, ports de entrada/salida, mímicos, etc./
- Métodos de acceso a la información (Information Access Methods - IAM): la forma de acceder a los datos almacenados en el ODT es a través de estos métodos. Están implementados como métodos de clases que permiten que el usuario almacene, lea, actualice y borre la información de las ODT. La única forma de acceder a la ODT es a través del IAM.

- . Máquina de Ejecución (Execution Engine - EE): consta de un conjunto de rutinas que inician la ejecución del sistema. Estas ejecutan concurrentemente, cada una en distinto hilo de ejecución, sincronizadas por una rutina de planificación de procesos de alto nivel. La EE está compuesta por los siguientes elementos:
 - a) Planificador de procesos de alto nivel (Task Scheduler - TS).
 - b) Analizador de Alarmas (Alarm Analyzer - AA).
 - c) Registro Histórico (Historic Recorder - HR).
 - d) Administrador de Mímicos (Mimic Manager - MM).
 - e) Intérprete de Comandos (Command Interpreter - CI).
 - f) Manejador de Ports (Port Handler - PH).

3. DESCRIPCION DE CLASES

Como fue explicado, IGNATIUS está construido como una biblioteca de clases implementando objetos para construir aplicaciones SCADA. Está compuesta por las siguiente clases:

3.1. IMAGE

Esta clase representa los puntos de la planta y los métodos para controlarlos. Todos los valores monitoreados son almacenados en un depósito central implementado con esta clase. La imagen está dividida en tres segmentos virtuales que representan valores enteros, analógicos y digitales. La implementación de segmentos virtuales permite que el usuario administre los puntos de la imagen sin importar su almacenamiento interno. La clase almacena información acerca de los puntos que representa, incluyendo sus valores, set-points, estados de alarma, cantidad de actualizaciones, porcentaje de actualización (para almacenamiento histórico).

Hay métodos para agregar, borrar, leer o actualizar un punto de 'una imagen. También se puede aumentar en uno la cantidad de actualizaciones, o modificar el estado de un punto de la imagen. Finalmente se puede devolver el número de puntos de la imagen almacenados en un objeto, devolver el segmento virtual donde está almacenado un objeto, y reconocer estados de alarma.

3.2. ALARM

Esta clase representa los valores de las condiciones de alarma que deben ser evaluados por el EE para detectar escenarios de alarma, así como los métodos para controlar" las condiciones de alarma. También enlaza condiciones de alarma con puntos de la imagen. Estos enlaces permiten que el usuario pueda definir diversas relaciones entre condiciones de alarmas y puntos de la imagen. Las rutinas de asistencia de alarma del PSL pueden analizar la información de escenarios de alarma detectados por el EE para darles el

tratamiento requerido. La clase almacena información para cada una de las condiciones de alarma que representa,

Los métodos asociados permiten agregar, actualizar y borrar condiciones de alarma, leer condiciones, aumentar en uno la cantidad de veces que una condición fue evaluada, calcular los valores mínimos y máximos basados en un valor de set-point y un porcentaje de tolerancia, buscar las condiciones de alarma enlazadas con un punto de la imagen, buscar el punto de la imagen enlazados con una condición de alarma o enlazar una condición de alarma con un punto de la imagen.

3.3. QUEUE /

Esta clase representa los mensajes enviados del SCADA a los dispositivos físicos (In Queue), y los mensajes enviados de los dispositivos físicos al SCADA (Out Queue). También representa los métodos asociados con estos mensajes. Se usa como canal de comunicación entre elementos de control externos (tales como PLCs) y el SCADA.

3.4. HISTORIC

Esta clase lee los mensajes de la In Queue y almacena los registros históricos con información/ del punto de la imagen en el mensaje, basado en un parámetro de actualización especificado por el usuario. Esta clase permite que el usuario recupera información acerca de distintos valores de un punto durante un período de tiempo, proveyendo al usuario una herramienta para analizar la información y detectar cambios o problemas en el entorno. El porcentaje de actualización del punto permite que el usuario especifique la tasa de almacenamiento para el histórico en cada punto de la imagen.

3.5. TASK

Esta clase sincroniza la ejecución de los distintos componentes del EE. Su uso puede extenderse para sincronizar rutinas escritas por el usuario (por ejemplo, una rutina para actualizar los datos en la pantalla). Almacena la información de las tareas componentes del EE. El planificador de alto nivel (TS) es un método responsable de sincronizar los otros componentes usando semáforos. Para hacer esto, la clase almacena el nombre, identificador del proceso, frecuencia de ejecución y tiempo de espera desde la última ejecución (en milisegundos), y el manejador de eventos.

3.6. PROCESO

Esta clase mode a un proceso de planta, por ejemplo: horno de secado, planta de evaporación, reactor, etc. Esta información puede usarla el ISL para mostrar datos del proceso en pantalla.

3.7. MENSAJE

Esta clase encapsula los mensajes enviados por la clase MIMIC al SCADA a través de un pipe del sistema. Cada mensaje contiene información acerca de un punto de la imagen y sus atributos en la pantalla: posición, color, etc. Esta información es usada por el ISL para mostrar los puntos de la imagen del proceso en la pantalla.

3.8. PIPE

Esta clase implementa, a través de pipes del sistema, el canal de comunicación entre la clase MIMIC y el SCADA para transmisión de mensajes. La clase MIMIC escribe MESSAGEs en el PIPE, mientras que el ISL lee estos mensajes para mostrar la información sobre el punto de la imagen, en la pantalla.

3.9. MIMIC

Esta clase conecta procesos del mundo real con puntos de la imagen y sus atributos de pantalla: posición, color, etc. El administrador de mímicos (MM) escribe los atributos de los puntos de en el PIPE con una frecuencia determinada por el usuario.

Las clases MESSAGE y PIPE son partes de esta clase. Se usan con propósitos de implementación, y sirven como vehículo de comunicación entre el MM y el ISL.

3.10. PORT

Esta clase maneja los ports de entrada/salida que se usan para conectar la computadora a los dispositivos físicos (especialmente dispositivos de control industrial tales como PLCs u otros controladores). La clase enlaza dispositivos físicos con los puntos de la imagen, permitiendo que el usuario pueda definir relaciones múltiples entre ellos. Para hacerlo, usa las colas In y Out. También inicializa los ports de entrada/salida.

3.11. COMMAND

Esta clase provee un mecanismo para definir y administrar comandos ingresados por el operador. También valida los comandos y parámetros ingresados para determinar si son correctos. Con esta aproximación, la definición, manejo y validación de comandos se separan de la ejecución del comando. Mientras que esta clase encapsula las citadas funciones, las rutinas de asistencia de comandos deben implementarse en el PSL. La ejecución de rutinas de respuesta a comandos pueden ejecutarse de dos formas: planificadas o inmediatas. En el primera caso, el comando' ingresado se agrega a una cola de comandos y se ejecutan una vez que todos' los comandos previos en la cola han terminado su ejecución. En el segundo caso,

el comando ingresado se ejecuta inmediatamente sin esperar que los comandos en la cola terminen su ejecución.

4. CONSTRUCCION DE APLICACIONES SCADA

Para probar la herramienta y mostrar su uso, se construyeron dos sistemas SCADA diferentes. El primero muestra el uso de la herramienta para construir un SCADA configurable. Permite que el operador configure dinámicamente los recursos, definiendo puntos de la imagen, condiciones de alarma procesos, dispositivos físicos asociados a los procesos, representaciones de pantallas, actualización e puntos de la imagen, manejo de condiciones de alarma, etc. La flexibilidad dada por la configuración dinámica permite usar el SCADA para cualquier implementación de propósito general.

El segundo muestra el uso de la herramienta para construir un SCADA de propósito específico. En este caso, los recursos se definen estáticamente y el operador no tiene capacidad de configurarlos dinámicamente. En ambos ejemplos se consideraron los requerimientos de un proceso de un horno de secado [Ben93].

Basándose en estos requerimientos, se hizo un diseño del sistema usando MASCOT, y se lo implementó usando IGNATIUS y las herramientas de programación visual mencionadas previamente.

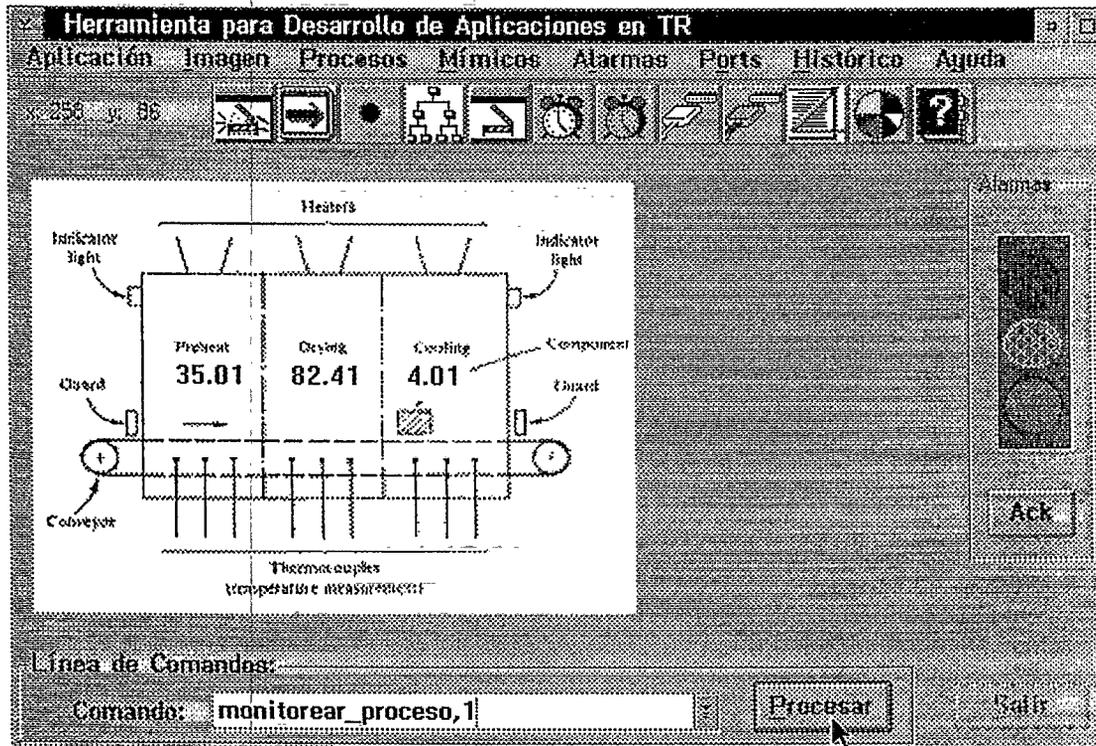


Figura 3 - Primer ejemplo. ventana principal

El primer ejemplo nos permite medir la utilidad de la herramienta para construir un SCADA complejo y también para determinar el esfuerzo necesario para construir un SCADA usando todas las facilidades de la herramienta. El segundo ejemplo permite medir la utilidad de la herramienta para construir un SCADA simple y también determinar el mínimo esfuerzo necesario para construir un SCADA usando la herramienta.

Se probaron los ejemplos compartiendo la CPU con otras aplicaciones, y cambiando el modo de ejecución del SCADA a background. Debido al algoritmo de planificación del sistema operativo y al diseño del TS, el tiempo de respuesta del SCADA siempre estuvo dentro del marco de tiempo, esperado. El algoritmo de planificación con prioridades y remoción evita el uso excesivo de la CPU por parte de procesos que podrían demorar la ejecución del EE. Las rutinas del EE fueron planificadas dentro del TS con distintos intervalos de tiempo. En una primera etapa, estas rutinas fueron ejecutadas con intervalos mayores que un segundo, y los tiempos de respuesta siempre fueron cumplidos. En la última etapa, se obtuvo un intervalo mínimo para permitir que el sistema responda dentro del tiempo esperado considerando una carga de CPU media que puede verse en la Tabla 1.

Rutina EE	Intervalo mínimo con carga media de CPU (en segundos)
AA	0.15
HR	0.45
MM	0.35
CI	0.65
PH	0.55

Tabla 1. Intervalos mínimos del EE

Esto permite que el usuario ejecute otras tareas mientras el SCADA sigue ejecutando, por ejemplo, analizar registros históricos, lanzar nuevos procesos, lanzar una sesión al sistema operativo; 'agregar condiciones de alarmas, procesos, mímicos, etc.,' compartiendo la CPU sin dejar al SCADA fuera de línea.

El tiempo ocupado en diseñar, codificar y probar los ejemplos" fue mínimo. Un total de 120 horas hombre fueron necesarias para el primer ejemplo, mientras que sólo 24 horas hombre fueron usadas para desarrollar el segundo ejemplo. Estos valores reflejan claramente la utilidad de la herramienta y la reducción en el ciclo, de desarrollo, reduciendo la complejidad y dejando que el usuario se concentre en aspectos de diseño de alto nivel.

El esfuerzo de desarrollo es proporcional a la complejidad del entorno del problema en particular. Como se pudo ver en los ejemplos, el esfuerzo es mínimo para entornos de complejidad 'media.'

5. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo hemos presentado una herramienta para proveer un entorno integrado y flexible para desarrollar aplicaciones SCADA. Aunque las técnicas de control no han cambiado demasiado desde el comienzo de computadoras para control de procesos, la supervisión de procesos ha sufrido diversos cambios, y en la mayoría de los casos estas aplicaciones no son lo suficientemente flexibles: su rango de uso puede ser muy limitado, o su complejidad ser excesiva para aplicaciones de control simples.

Pudimos ver que todos los sistemas de supervisión están compuestos por tres niveles. Hemos encapsulado el nivel de servicios básicos (BSL) en la herramienta, permitiendo tener una herramienta poderosa y flexible para construir SCADAs de propósito general y particular.

Se evaluó la utilidad de la herramienta y se determinó el esfuerzo necesario para construir SCADAs con distintos requerimientos de complejidad. Se aplicaron pruebas de desempeño para probar el cumplimiento de las restricciones de las tareas del EE con diferentes intervalos.

El uso de tecnología orientada a objetos simplificó el diseño y desarrollo de la herramienta. La detección de errores también fue simple debido al encapsulamiento de datos y código relacionado. Este paradigma también proveyó reusabilidad y facilidad de mantenimiento.

El sistema operativo OS/2 Warp 3.0 proveyó un entorno de desarrollo estable. Su modo de ejecución protegida evita las caídas del sistema debido a errores involuntarios en el código. Las características de multitarea mejoraron la productividad durante el ciclo de desarrollo.

OS/2 aísla el nivel de hardware del nivel de software de aplicaciones. Las aplicaciones no pueden operar directamente con los dispositivos físicos: deben usarse manejadores de dispositivos. Esto aísla al software de aplicación de las características del dispositivo físico, pero lo hace dependiente del manejador de dispositivo cargado para manejar el dispositivo (por ejemplo, el resultado de una operación de escritura a la consola puede cambiar dependiendo si el dispositivo es o no ANSI). Algunos dispositivos están disponibles para los programas de aplicación sólo si los manejadores de dispositivos apropiados han sido instalados.

Se pudo ver que la descomposición propuesta en tres niveles de servicio y el uso de la herramienta reduce todo desarrollo de un SCADA a la construcción de los niveles de ISL y PSL, reduciendo el ciclo de desarrollo y la complejidad.

La herramienta también permite que el usuario se concentre en aspectos de diseño de alto nivel, proveyendo un entorno de desarrollo completo que puede ser usado por programadores inexpertos, permitiéndoles mantener la integridad del sistema, hacer cambios y desarrollar aplicaciones correctas.

En la actualidad hemos comenzado a usar la herramienta en el curso de Sistemas de Tiempo Real de nuestro Departamento. También será integrada con otros 'proyectos, donde se está usando una versión anterior de la herramienta [Wai93]. Como extensión al trabajo pueden encararse nuevos desarrollos, incluyendo:

- . Migración de la herramienta a otros sistemas operativos.
- . Diseño e implementación de un sistema supervisor usando una arquitectura cliente/servidor distribuida. El BSL residiría en el servidor, y los niveles de ISL y PSL residirían en el cliente.
- . Diseño e implementación del BSL como un servidor único, proveyendo servicios a distintos sistemas de supervisión.
- . Agregado de interfaces a distintas PCs industriales y PLCs.
- . Prueba de la herramienta en entornos de aplicación rudos (incluyendo su embebido en ROM).
- . Desarrollo de otras aplicaciones de supervisión: planillas electrónicas, generador de estadísticas, etc.

6. REFERENCIAS

[Ben93] BENNET, S. "Real-time computer control: an introduction". Prentice-Hall International. 2nd. Edition. 1993.

[Ben96a] BENITEZ, S.; SEOANE, J.; WAINER, G.; BEVILACQUA, R. "Desarrollo e implementación de una herramienta para construir sistemas de supervisión de procesos". Tesis de Licenciatura. Departamento de Computación. Facultad de Ciencias Exactas Naturales. Universidad de Buenos Aires. 1996.

[Ben96b] BENITEZ, S.; SEOANE, J.; WAINER, G.; BEVILACQUA, R. "IGNATIUS: a tool to develop SCADA systems". Submitted to CISS '97, Maryland, Baltimore. 1996.

[Kap95] KAPLAN, G.; HOUSE, R. "Data acquisition software for engineers and scientists". IEEE Spectrum, May 1995. pp. 23-39.

[Mas87] MASCOT. "The official handbook of MASCOT, version 3.1". Computing Division, RSRE, Malvern. 1987.

[Miy93] MIYAGI, P.; PEREIRA RIBEIRO BARRETO, M.; SILVA, J. "Demótica: controle e automacao". Vol II, VI EBAI. 1993.

[Wai93] WAINER, G. "SSDT: una herramienta para desarrollar sistemas de supervisión en tiempo real". Anales de las I Jornadas chilenas de Ciencias de la computación. Octubre 1993. pp. 44-52.

[Wai94] WAINER, G. "Introducción al desarrollo de sistemas de tiempo real". Publicación on-line: "<http://www.dc.uba.ar/people/materias/str>".

Alfanego:
**Sistema de Relevamiento, Análisis y
Recomendación de Actividades Agropecuarias**

Mario Lucero, Alberto Sánchez, Aristides Dasso, Germán Montejano, Roberto Uzal
Dpto. de Informática - U.N.S.L.
Ej. de los Andes 950 1er. Piso (5700) San Luis
TE/Fax: 0652 - 30059
Email: alfanego@unsl.edu.ar

ÁREA: Informática y Comunicaciones

Resumen

El presente trabajo describe un desarrollo que tiene por objeto agilizar y darle seguridad al proceso de diagnóstico y recomendación a los productores rurales y a los profesionales del Area agropecuaria, en las actividades de producción agrícola ganadera. Mediante la selección de recursos naturales, georreferenciados (mapas) y actividades con sus restricciones asociadas para producir como resultado un mapa y un reporte de texto indicando el área donde es posible desarrollar la actividad elegida.

Palabras claves: mapa base, Sistema de Información Geográfico, recursos naturales georreferenciados, variables adicionales, funciones de producción

1. Visión general del Sistema

Contamos, con recursos naturales georreferenciados, restricciones de las posibles actividades y funciones de producción, para realizar el análisis de una actividad. La actividad de diagnóstico realiza el proceso de cruzar la información (actividad deseada vs. ambiente disponible) para obtener la primer salida (es/no es posible) de la actividad propuesta.

El paso anterior puede ser refinado mediante distintas alternativas como: el agregado de variables adicionales (no georreferenciadas previamente, ej. fertilizante) que modifican las condiciones del ambiente, la posterior aplicación de funciones de producción y el análisis de costos y rindes.

1.1 Objetivos del sistema

El término “mapa base”, es utilizado para referirnos a todo mapa que haya sido ingresado desde un Sistema de Información Geográfico (SIG).

1.1.1 A partir de mapas que contienen información relevante para la actividad agropecuaria de una determinada zona, evaluar que posibilidad tiene determinada actividad (previamente definida en cuanto a sus restricciones); de darse con éxito en determinada zona del mapa base.

1.1.2 Seleccionar mapas base para evaluar qué actividad puede darse en algún área de la zona seleccionada.

2. Descripción del proyecto

2.1 Antecedentes

Inicialmente, el diagnóstico se realizaba en una determinada zona, luego el proyecto se amplió, incluyendo el proceso de recomendación y estudio de costos.

El sistema exige contar con la entrada de:

- a) recursos naturales (mapas) georreferenciados,
- b) actividades y sus restricciones,
- c) variables adicionales (no georreferenciadas previamente),
- d) funciones de producción y
- e) análisis de costos y rendimientos.

No es requerimiento del sistema que todas las entradas se encuentren presentes.

2.2 Descripción detallada de aspectos clave del proyecto

2.2.1 Recursos Naturales (mapas)

Recurso es cualquier elemento que el hombre utiliza o puede hacerlo, en su propio beneficio. Los naturales son los que existen, en diversas formas, en la naturaleza, un bosque, el suelo, el agua, son ejemplos de ello. Estos recursos pueden ser renovables o no renovables. Los primeros pueden regenerarse Naturalmente ej. un bosque natural, los segundos por lo contrario se agotan, ej. /explotación minera.

2.2.2 Actividades

Entendemos por actividad a la clase de trabajo que dedica el hombre en una región o área determinada. Así, actividad agrícola, es cuando sobre cierta zona se realizan cultivos agrícolas. Por ejemplo, actividad maíz implica que en determinado lugar se realiza el cultivo de maíz, o actividad trigo implica que en determinado lugar se realiza el cultivo de trigo. Estas últimas son las distintas actividades agropecuarias que son procesadas por el sistema.

Una actividad puede ser posible siempre y cuando existan ciertos elementos básicos para su desarrollo, si uno de ellos faltare la actividad no podría existir; la ganadería no existirá si el agua tiene un alto porcentaje de arsénico, la actividad maíz no podría existir en suelos rocosos, etc. Las

distintas valoraciones que se realicen de cada uno de estos elementos básicos indispensables producirá diferentes grados de éxito de la actividad.

2.2.2.1 Restricciones para una actividad

Cada actividad es descripta por una lista de recursos naturales acotados por valores que influyen en la producción de ella. A esta lista se la denomina **restricciones** para una actividad.

El término restricción se aplica a las condiciones que deben cumplirse para que una actividad pueda ser efectivamente realizada. Si las restricciones son relativamente laxas demarcarán únicamente a la actividad como posible o imposible, si las restricciones son más ajustadas se podrá demarcar no solo la posibilidad de la actividad sino también su nivel o grado de éxito.

2.2.3 Variables Adicionales

Las variables adicionales (no georreferenciadas previamente en el mapa) son restricciones que inciden en el resultado final, pero su ubicación espacial no se encuentra registrado en mapas, por ejemplo la opción de fertilizar o no un cultivo. Son las variables que, el hombre puede manejar aumentando o disminuyendo el contenido de la misma independientemente del nivel que exista naturalmente. Si en el mapa 'base por ejemplo no existe la variable 'nitrógeno disponible; la actividad podrá ser evaluada considerando que este recurso (nitrógeno) no es limitante porque es agregado por el agricultor; lo mismo ocurre con la cantidad de agua, que también puede ser agregada por el agricultor.

2.2.4 Funciones de Producción

La producción o productividad de un cultivo se relaciona con los niveles de ciertos elementos del ambiente que lo condicionan. Por ejemplo el rendimiento de maíz depende de la cantidad de agua caída durante el ciclo del cultivo, de la cantidad de nitrógeno disponible en el suelo, etc. Cuando se tienen valores que cuantifican el efecto de cada una sobre el resultado final (kg./ha de maíz producidos) se tiene una función de producción, esto es la producción del cultivo en función del valor del cuantificador y la variable cuantificada.

Estas son fórmulas o ecuaciones constantes, relacionadas con las restricciones y con las variables adicionales que más afectan a la actividad. Se toman de la literatura o del conocimiento que el profesional posea.

2.2.5 Análisis de Costos y Ganancias

Los costos son los valores en una unidad monetaria dados por el mercado para los distintos insumos necesarios para realizar una determinada actividad.

Las ganancias están dadas por las relaciones entre los valores de los insumos y de los productos. Existe una serie de estimadores del nivel de rentabilidad de /un cultivo como Margen Bruto, Tasa Interna de Retorno, Valor Actual Neto, etc

El análisis de costos y ganancias aporta a las recomendaciones que realiza el sistema, influyendo en las decisiones tomadas por éste al momento de seleccionar una u otra determinada actividad.

2.3 Necesidades de procesamiento de información

2.3.1 Para los recursos naturales georreferenciados

A los fines de este proyecto el término “georreferenciado”, como su nombre lo indica, implica un dato con referencia geográfica, esto es con ubicación precisa en el eje de coordenadas planas (ej. un punto representativo de una perforación petrolera ubicado a los 32 grados de latitud sur y 65 grados de longitud oeste).

Los recursos Naturales Georreferenciados son introducidos al sistema provenientes de algún Sistema de Información Geográfica (SIG). Los mapas son consultado en el sistema, pero no modificados dentro de éste, esto se hace a través de un SIG, ya que no es el objetivo del sistema.

2.3.2 Para las Actividades

Las actividades son ingresadas, actualizadas y almacenadas dentro del sistema.

El/los valor/es de las restricciones que se le asignan a cada ítem es contrastado contra la misma restricción que se encuentra referenciada en un mapa.

El ingreso de una actividad es un proceso independiente de la aplicación de ésta en una ejecución particular del sistema.

2.3.3 Para las Variables Adicionales (no georreferenciadas)

Estas variables que son utilizadas por el usuario final para aumentar o reducir el valor contenido de uno o más determinados recursos naturales, es tenido en cuenta en el momento de realizar las recomendaciones para una determinada actividad.

2.3.4 Para las Funciones de Producción

El procesamiento de estas funciones son ponderaciones que se aplican a los valores d un mapa. Este proceso debe realizarse después de aplicar las

restricciones sobre la actividad seleccionada, produciendo un resultado ponderado por la función de producción.’

Las funciones de producción son ingresadas al sistema por los profesionales.

2.3.5 Para obtener resultados y reportes

Una vez que las entradas se encuentran presentes, existen distintas alternativas para obtener resultados, las cuales son enumeradas a continuación:

- . Asociando los mapas con las restricciones se obtiene un mapa resultante con todas las áreas en las cuales las restricciones se cumplen.
- . Con el agregado de las funciones de producción, se realiza en el área seleccionada, los cálculos que la función indica.
- . A los resultados expuestos en los puntos anteriores, también se le puede agregar variables adicionales (no georreferenciadas) y un análisis de costos y ganancias para mostrar recomendaciones sugeridas para la actividad propuesta.

2.4 Enfoques metodológicos

2.4.1 Fundamentos de la selección de los enfoques metodológicos

2.4.1.1 Primero fueron relevados aspectos procedurales llegando en una etapa posterior a conceptualizarse los Objetos.

Comenzamos utilizando los *Métodos Estructurados* basados en la riqueza de herramientas gráficas en la etapa de análisis, así como también la simplicidad de éstas para ser mostrada a nuestros usuarios en el momento de las entrevistas.

Las características de este modelo son enriquecidas además con una representación modular clara, la casi nula redundancia de información a ser diagramada y la fácil documentación.

Contando ya con un análisis del sistema desarrollado con métodos estructurados, y con lecturas bibliográficas realizadas, esto nos condujo a la utilización de la metodología de Martin-Odell, para la modelización de los objetos.

Esta metodología se basa en el uso de un modelo conceptual único que tiene las siguientes consecuencias:

- Una mayor productividad
- Menos errores
- Una mejor comunicación entre los usuarios, analistas, diseñadores y técnicos
- Resultados de mejor calidad
- Más flexibilidad
- Mayor inventiva.

2.4.1.2 La utilización de una herramienta de implementación estructurada con extensiones, Visual C++, en lugar de una “pura” Orientada a Objetos como Smalltalk.

En primer lugar, nuestra aplicación se ajusta perfectamente para ser desarrollada con las ventajas que otorga el paradigma de objetos y no capitula frente a las desventajas que tiene tal paradigma, por eso parece importante buscar como soporte un lenguaje orientado a objetos.

En segundo lugar del porqué de esta elección se basó en puntos conocidos, como por ejemplo, desde un comienzo deseamos un sistema con ambiente WINDOWS, y casi todos los lenguajes de programación existentes con soporte WINDOWS son Orientados a Objetos, o poseen versiones para soportar dicho paradigma, entonces, para un mejor pasaje y adaptación del análisis y diseño a la implementación, sacábamos ventajas partiendo de una metodología OO.

2.4.2 Características del análisis y definición de especificaciones

El enfoque Yourdon-De Marco es utilizado en nuestro sistema para modelizar el esquema de procesos, en particular los diagramas de flujo de datos (DFD).

Los estudios y propuestas contenidos en el análisis del sistema han sido formalizados según el siguiente esquema:

- 1, Lista de eventos detectados en el ámbito estudiado
- 2 Modelo ambiental del sistema.
- 3 Primer nivel de desagregación del diagrama de flujo de datos.
- 4 Segundo nivel de desagregación del diagrama de flujo de datos.

5 Tercer nivel de desagregación del diagrama de flujo de datos.

Los puntos 3, 4 y 5 contribuyen al armado del modelo de procesos, que integra el modelo de comportamiento el cual es parte del modelo esencial.

Es importante destacar que no fueron seguidos todos los pasos de esta metodología exhaustivamente dado que la complejidad del sistema modelado no lo requería.

2.4.3 Particularidades del diseño del sistema

James Martin y James J. Odell en su libro "Análisis y Diseño Orientado a Objetos" combinan las técnicas orientadas a objetos con las herramientas CASE, la programación visual y diseñar un código reutilizable, construir y almacenar objetos complejos y modelar la forma de comprender la realidad.

En las metodologías tradicionales para la generación de sistemas, los modelos conceptuales utilizados para el análisis difieren de los que se emplean para el diseño.

En las técnicas orientadas a objetos (OO), todos utilizan el mismo modelo conceptual, analistas, diseñadores, programadores y, de modo particularmente importante, los usuarios finales. Todos piensan en los tipos de objetos, los objetos y su comportamiento.

La transición del análisis al diseño es tan natural, que a veces es difícil especificar el punto final del análisis y el punto inicial del diseño.

El análisis de la estructura de objetos se ocupa de definir las categorías de objetos y la forma en que los asociamos. Preguntamos: ¿Qué tipos de objetos hay? ¿Cuáles son sus relaciones y funciones? ¿Qué subtipos y supertipos son útiles? ¿Hay algún tipo de objeto compuesto por otros objetos?.

Cuando el análisis pasa a la etapa del diseño de la estructura de objetos, identificamos las clases (la implantación de tipos de objetos). Se definen las superclases, subclases, rutas de herencia, los métodos a utilizar y se lleva a cabo el diseño en detalle de las estructuras de datos.

El análisis del comportamiento de objetos se ocupa de modelar lo que ocurre a los objetos al paso del tiempo. Nos preguntamos: ¿En qué situación pueden estar las clases de objetos? ¿Qué tipo de eventos cambian estos estados? ¿Qué sucesión de eventos ocurre? ¿Qué funciones resultan de estos eventos y cómo se activan?.

Después sigue la etapa de diseño. En ésta nos preocupamos por el diseño detallado de los métodos, ya sea con técnicas por procedimiento o sin procedimientos. Se crea la entrada para los generadores de código. Se diseña

la pantalla y se diseñan y generan los diálogos. Finalmente, se construyen y desarrollan los prototipos.

2.4.4 Soporte CASE a el/los enfoques metodológicos seleccionados

La herramienta CASE que se utilizó para el desarrollo de la etapa de análisis es System Architect que fue desarrollado por Popkin Software and Systems Inc.

Esta herramienta soporta la técnica utilizada en el presente desarrollo:

- DFD Yourdon/De Marco

Suministra entre otros, los siguientes reportes:

Definición de flujos de datos
Definición de-almacenamiento de datos
Definición de estructuras de datos
Definición de elementos de datos
Definición de procesos

Realiza los siguientes chequeos de consistencia:

Balanceo con la burbuja padre
Balanceo con la burbuja hija
Balanceo horizontal
Chequeo de nombres y definiciones duplicadas

Para la etapa de diseño no se disponía de una herramienta CASE por lo que se realizó el seguimiento de la metodología de Martin-Odell en forma manual.

3. Sistemas de Información Geográfica (SIG)

Los SIG, son sistemas para la entrada, almacenamiento, recuperación, análisis y mostrado de datos geográficamente interpretados. La base de datos está típicamente compuesta de un gran número de representaciones espaciales (frecuentemente llamadas "extensiones") como mapas.

Llamamos imagen a una representación de alguna porción rectangular de espacio. En nuestro caso las imágenes son almacenadas como una colección de códigos de atributos numéricos en formato raster.

En nuestro sistema, las imágenes que usamos son *mapas*.

Nuestro sistema aprovecha la facilidad para la confección y captura de imágenes con que cuentan los SIG tomando un formato raster (BMP) como estándar para entrada, que proveen la mayor parte de estos sistemas.

Raster es comúnmente usado como un término para describir un sistema de representación de imágenes donde la imagen es compuesta de pequeñas celdas internamente uniformes, acomodadas en una grilla. El orden de la imagen almacenada está típicamente dado por líneas de scan, es decir procesadas de izquierda a derecha a lo largo de las líneas, y desde el tope al final comenzando en la primera línea.

3.1 En particular Idrisi

Dado el predominio de Windows, y la riqueza de su ambiente gráfico decidimos comenzar a estudiar Idrisi versión para Windows, el cual nos provee ahora las imágenes necesarias.

Idrisi es un SIG basado en raster y un software de procesamiento de imágenes que sirve para un extenso rango de tareas. Su estructura de archivos es además simple y transparente lo cual lo hace fácil para manejar.

Las imágenes (mapas) BMP importadas de este SIG es *necesario* que se encuentren en formato binario y los datos en formato ASCII.

3.2 Influencia de los Sistemas de Información Geográfica

La influencia está dada en la facilidad de manejo y confección de mapas que tienen los SIG bajo Windows.

Aprovechando estas herramientas, así como su difusión en el ambiente en el cual el sistema va a trabajar, se decidió que todas las imágenes sean ingresadas y modificadas desde el ambiente de un SIG.

Esta determinación fue tomada ya que estos software son expertos en el manejo y captura de este tipo de imágenes como también la consistencia de datos que cada mapa soporta. No consideramos una mejora a nuestro sistema, proveer estas facilidades que son una de las razones de existencia de los SIG.

4. La implementación

4.1 Selección del lenguaje de implementación

Una de las grandes ventajas de trabajar con Windows es que todas las ventanas se comportan de la misma forma y todas las aplicaciones utilizan los mismos métodos básicos (menús desplegables, botones) para introducir órdenes. Por lo tanto decidimos programar en Visual C++.

Las herramientas de Visual C++ reducen el trabajo pesado de la programación. App Studio, AppWizard y ClassWizard reducen significativamente el tiempo que se necesita para escribir código que es específico a la aplicación. Por ejemplo App Studio crea un archivo de cabecera que contiene valores asignados a las constantes de tipo *#define*. App Wizard genera el esqueleto del código para toda la aplicación y Class Wizard genera los prototipos y cuerpos de las funciones para los controladores de mensajes.

De ese forma nos encontramos recompensados por la claridad de programación que introduce el hecho de usar objetos, así como la amplia reutilización, por el lado del código desarrollado por nosotros y la potencia del ambiente visual para Windows, de amplia difusión en este momento y muy agradable a la vista del usuario.

También es importante destacar la facilidad de uso de un ambiente de hipertexto, colores y programas de implementación de interfaz con periféricos (ej.: impresoras).

5. Conclusiones

5.1 Respetto de la naturaleza del sistema

La sistematización del proceso de diagnóstico y recomendación de actividades agropecuarias nace de la necesidad que tiene el usuario que realiza actividades agrícolas ganaderas de poder contar con un soporte ágil, confiable y rápido de apoyo a sus decisiones.

Dado que en este momento se cuenta con suficiente información de recursos naturales en soporte magnético, era necesario implementar la toma de decisión que realiza el experto de la respectiva área.

5.2 Respetto de las Características del proyecto.

El producto final que hemos desarrollado es de amplia aceptación por parte de nuestros usuarios, ya que además de presentar un soporte confiable en cuanto a las decisiones, el ambiente está totalmente desarrollado en **castellano**.

5.2.1 Acerca de aspectos clave del proyecto

Los tópicos abajo desarrollados describen los aspectos claves que encuadra este proyecto.

- . Apoyados en la facilidad que Windows nos suministra, cuenta con un ambiente que soporta múltiples ejecuciones; lo que le facilita al usuario el hecho de poder estar realizando múltiples corridas al mismo tiempo, donde

pueden intervenir los mismos, o distintos mapas y actividades al mismo tiempo.

- . Se le da al usuario la posibilidad de volver hacia atrás en la superposición de mapas sin necesidad de tener que comenzar nuevamente, así como también la información de los mapas que se superponen en cada corrida en particular.
- El hecho de contar con un **zoom** incorporado permite que se puedan visualizar áreas de tamaños tan grande como el país completo, así como también pequeñas, como la ubicación de un determinado lote, ayudados por la simplicidad de consultar los valores reales que se encuentran almacenados en los mapas superpuestos en un punto específico con la única operación por parte del usuario de presionar el botón izquierdo del mouse en el punto que elija.
- . El hecho de contar con un Help Online durante toda la ejecución le da la seguridad al usuario de poder encontrar ayuda en cualquier punto que se encuentre, sin tener necesidad de recurrir al manual del usuario.
- . La actividad de relacionar los distintos mapas con las restricciones asociadas a una actividad, se realiza 'en un ambiente de ventanas totalmente claro, donde el usuario necesita hacer uso del mouse únicamente.
- Una vez terminada una ejecución Alfanego le permite al usuario visualizar al mismo tiempo, los mapas seleccionados, el mapa resultante y el reporte de texto que se construye. Esta facilidad le hace ahorrar tiempo, en cuanto la verificación de la corrida realizada, además de poder realizar su propio análisis en una única pantalla de un monitor.
- . Alfanego permite la impresión de: los mapas base, las actividades y sus restricciones ingresadas al sistema. y los distintos reportes realizados, los cuales pueden ser utilizados como documentación..

Todos los resultados hasta acá expuestos fueron verificados exitosamente, con casos reales, por personal del INTA delegación Villa Mercedes, donde se pudieron ver las facilidades y rapidez que esta herramienta Provee a los usuarios sin necesidad de un fuerte aprendizaje inicial.

5.2.2 Respecto de las necesidades de procesamiento de información

5.2.2.1 Tratamiento de imágenes (mapas)

Como se dijo anteriormente se optó por la utilización de los SIG para la incorporación de las imágenes a nuestro sistema.

Llegamos a esta conclusión luego de leer bibliografía sobre tratamiento de imágenes: “Supercharged Bitmapped Graphics” y “Bitmapped Graphics Programming in C++”, donde encontramos que ya se cuenta con rutinas para conversión en los formatos más estandarizados (ej. BMP, **GIF**, **TIFF**, **PCX**, etc.).

Con relación al tratamiento de los datos que se relacionan a la imagen, éstos tienen que ser creados en forma conjunta con la imagen, por lo que se tomó la misma política que en el párrafo anterior.

Una vez que la imagen y sus datos asociados fueron conseguidos, había que superponer las imágenes e interactuar con los datos.

Para la primera de las operaciones, superponer las imágenes, tuvimos que utilizar no sólo técnicas descritas por el lenguaje sino crear nuestros propios métodos, para lograr ver dos o más imágenes superpuestas en la pantalla.

Una de las técnicas que más empleamos en el tratamiento de imágenes son los *Mapas de bits independientes de dispositivos* (DIBs), que son formatos de mapas de bits alternativos que resuelven el problema de la portabilidad. Cualquier computadora en la que se ejecute Windows puede procesar DIBs, que generalmente se almacenan en los archivos de disco con extensión BMP.

5.2.2.2 Tratamiento de datos

Para el procesamiento de datos hubo que desarrollar un scanner y un parser para validar las restricciones ingresadas por los profesionales al sistema, y cuyo tratamiento se realiza enteramente dentro de éste.

Una vez que las restricciones son validadas se las compara con los datos asociados a la imagen.

El hecho de comparar valores nos llevó a tener que manipular archivos ASCII, usualmente más de uno (uno por cada mapa superpuesto). Como no se cuentan con funciones para acceso directo a posiciones específicas en este tipo de archivos tuvimos que implementarla. Esta función de acceso, realiza tantas búsquedas como accesos sean necesarios a cada archivo, por lo que buscamos la mayor velocidad posible en el manejo de estas estructuras y las limitaciones del lenguaje.

5.2.2.3 Construcción de la imagen (mapa) resultante

Una de las salidas de nuestro sistema es una imagen resultante del proceso descrito en el punto anterior.

Este mapa lo construimos como una imagen BMP, y el código necesario para su realización es parte de las rutinas con que se cuentan en el libro "Progrese con Visual C++" y modificada por nosotros a los fines de nuestro sistema.

5.2.2.4 Construcción de reporte de texto

Se le provee al usuario un listado de reporte con el nombre de la actividad, sus restricciones utilizadas y los mapas asociados.

El archivo mencionado es un archivo de texto y 'posteriormente almacenado.

Además cuenta con las siguientes ventajas:

6. Sugerencias respecto de la continuación de la línea de desarrollo

Luego de la etapa de análisis, en coordinación con nuestros asesores, se decidió comenzar por lo que sería el núcleo del sistema total: el subsistema de diagnóstico.

Si bien se realizó un análisis de todo el proyecto en forma global, se pormenorizaron los aspectos que luego fueron diseñados y programados. Con lo que el diseño provee la futura incorporación de algunas de las posibles extensiones.

6.1 Subsistema de recomendaciones

Este subsistema es uno de los módulos que fue diagramado y cuya realización es dejada como futura extensión.

Para la realización de este módulo se deberían tener los siguientes lineamientos generales:

- a) el subsistema de recomendaciones toma como base los resultados del subsistema de diagnóstico.
- b) este subsistema debería producir recomendaciones en base a un sistema experto con conocimiento y reglas, ingresadas por los 'profesionales, cuyo contenido y forma aún no fue determinado.
- c) si ya se cuenta con el subsistema de análisis de costos y ganancias, al momento de realizar recomendaciones, deben ser tenidas en cuenta.

6.2 Subsistema de costos y ganancias

Este subsistema si bien es independiente del subsistema de recomendaciones, de ser implementado, puede ser significativo par el subsistema de recomendación.

Este subsistema no se diagramó en más detalle dado que aparte del asesoramiento! de los ingenieros agrónomos es necesario el aporte de personas del área finanzas.

La realización de este subsistema puede ser muy importante para los usuarios finales y a su vez da la oportunidad de integrar la aplicación de técnicas computacionales con el área de finanzas.

BIBLIOGRAFIA

Almeida Texeira, Amandio L. de, Edmar Moretti y Antonio Christofolletti. *Introducao aos Sistemas de Informação Geográfica*. Rio Claro, Brasil, Edicao do Autor, 1992.

Department of Geography and Environmental Planning, Towson State University. *IDRISI-L*. e7g4mor@toe.towson.edu

Eastman, J. Ronald. *IDRISI: A Grid-Based. Geographic Analysis System*. Massachusetts, Clark University, Graduate School of Geography, 1988.

Lorup, Eric J. *IDRISI www Tutorial*.
<http://www.fes.uwaterloo.ca/u/gbhall/idrisi/>

Luse, Marv. *Bitmapped Graphics Programming In C++*. EE.UU., Addison-Wesley Publishing Company, 1993.

Martin, James y James Odell. *Análisis y Diseño Orientado a Objetos*. México, Prentice Hall Hispanoamericana, S. A. 1994.

Rimmer, Steve. *Supercharged Bitmapped Graphics*. EE.UU., McGraw-Hill, Inc, 1992

Yourdon, Edward. *Análisis Estructurado Moderno*. 1 ed. México, Prentice-Hall Hispanoamericana, S.A., 1993.

Yourdon, Edward y Larry L. Constantine. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. EE. UU., Prentice Hall Building.

CONTROL ESTADÍSTICO de MATERIALES

Lic. Sandra U. CARTAGENOVA

Lic. Mario E. LISNIZER

El sistema que se presenta a continuación, está siendo utilizado actualmente por el departamento de control de calidad de la empresa PHARMA ARGENTINA dedicada a la producción de productos farmacéuticos. Este tiene la finalidad de disponer de un procedimiento automático tal que a partir de los resultados de los análisis efectuados a las materias primas recibidas : drogas y materiales de empaque (dichos análisis se efectúan previos a la aceptación de las mismas); permita : (1) Llevar control de la situación de cada proveedor y de cada fabricante de aquellos productos utilizados por el laboratorio; (2) Informar acerca de la cantidad, porcentaje y detalles analíticos de entregas erróneas con valores aberrantes, con valores polarizados o con tendencias hacia determinados valores extremos; (3) Determinar para cada materia prima y cada una de sus características (parámetros), su grado de confiabilidad y de riesgo; (4) Ayudar en la toma de decisión al clasificar a proveedores y fabricantes en : Potencia, Normal, Preferencial o Certificado y en la autorización de análisis reducido.

[Definimos " **Análisis Reducido** " a toda disminución de análisis respecto a lo especificado, a efectuar para una materia prima, proceso, producto semielaborado o producto terminado. Dicha reducción puede aplicarse tanto al nivel de muestreo , a la frecuencia de análisis como así también a la cantidad de parámetros a analizar, entendiendo por "**parámetros** " al conjunto de especificaciones a analizarse para cada una de las distintas materias primas .

1.- DESCRIPCIÓN de TAREAS efectuadas en el LABORATORIO

- El proveedor entrega la materia prima a la empresa y en caso de haberla analizado éste en su laboratorio, el certificado con los resultados obtenidos.
- Cada materia prima tiene asociado un conjunto de parámetros o características que deben analizarse para determinar su aceptación o rechazo. Dichos parámetros y las especificaciones que deben cumplirse, varían en cantidad y detalle según la droga o el material de empaque que se esté tratando. Las materias' primas se encuentran agrupadas además en distintos rubros y subrubros ; se puede por lo tanto, obtener también informes según los mismos.

Según cuál sea la materia prima recibida, se analizan entonces los parámetros correspondientes.

- Una-vez obtenidos por el laboratorio PHARMA los resultados de análisis, se completa un formulario formulario (llamado protocolo) en donde figuran los datos del proveedor y del fabricante, el código de materia prima y el rubro al que pertenece, la fecha, de recepción del material la fecha de comienzo del análisis y la de finalización, los datos del analista que intervino, las **especificaciones que debe cumplir** cada uno de los parámetros asociados los.. resultados obtenidos para cada uno de los parámetros analizados por el lab PHARMA como así también los datos de los resultados obtenidos por el proveedor (éste en el caso en que el proveedor acompañe la entrega con el certificado de análisis y el dictamen (Aprobado, Aprobado con Reclamo o Rechazad6 según establezca a partir de la comparación de los resultados obtenidos con los especificados como correctos).

2.- ¿ POR QUÉ SE DESARROLLÓ EL SISTEMA DE CONTROL ?

Existía un proyecto cuyo objetivo final era facilitar una mejora continua en los procesos y en la calidad de los productos a partir del uso de materias primas confiables. Para ello, debía apuntarse a dos temas importantes :

1. Análisis reducido

2. Certificación de Proveedores

Luego de analizar los procesos manuales existentes, se llegó a la conclusión de que eran útiles para alcanzar el objetivo, pero debido a la gran cantidad de materias primas con las que se trabaja (aproximadamente 500 drogas y 1500 materiales de empaque) y la gran cantidad de variables involucradas en el control analítico de cada materia prima, era necesaria una herramienta que pudiera manipular un gran volumen de datos y emitir informes que ayudaran en la toma de decisiones. Es por eso, que se determinó que dicha herramienta debía ser un sistema de computa& que incluyera entre otros, procesos basados en una sólida estructura matemática.

Podemos decir que dicho sistema permite una “**Evaluación Global** ”, ya que se lleva control de todas las materias primas recibidas (drogas o materiales de empaque), de sus proveedores, como así también de los parámetros y rangos de valores obtenidos y especificados.

También podemos clasificarlo como un “**Sistema de Control Estadístico** “**ya que todos** los resultados de análisis son tenidos en cuenta, son comparados con valores especificados y las decisiones se basan en los resultados obtenidos en los procesos estadísticos que éste incluye.

3.- CRITERIOS ESTADÍSTICOS en la Certificación de Proveedores

El sistema informa para, cada proveedor y fabricante y para cada uno de los parámetros de las materias primas asociadas a ellos, su grado de confiabilidad basándose en lo que sigue :

3.1.- ¿ Existen valores dentro de los límites especificados ?

Para cada uno de los parámetros de la materia prima recibida, se analiza si el resultado obtenido en el laboratorio destino (PHARMA Argentina) se encuentra dentro de los límites especificados. Los parámetros podemos clasificarlos en NO Paramétricos (aquellos cuyo resultado es binario, e decir toma un posible valor entre dos . Ejm.: Cumple o No cumple, Positivo o Negativo) o en Paramétricos (aquellos cuyo resultado arroja un valor que puede o no estar dentro del rango de aceptación .

3.2.- ¿ Los valores proyectos por et proveedor son similares a aquellos obtenidos por el laboratorio PHARMA ?

Se analiza si el valor promedio de los resultados de las últimas “n” entregas obtenidos por el proveedor, no difiere significativamente del hallado en el laboratorio destino.[El número “n” se determina a partir del valor que se asocie a la probabilidad de cumplirse las especificaciones

en k partidas sucesivas. Nosotros requerimos que el proveedor tenga una probabilidad mínima promedio de $p=0.95$ (95%) de cumplir una serie de k determinaciones, por lo tanto tomamos las últimas 5 entregas consecutivas correctas]. En el sistema, el valor de “n” a considerarse, se ingresa asociado a cada rubro y subrubro (Cantidad de Lotes para Análisis Reducido). Una vez comparados los resultados de las últimas “n” entregas consecutivas correctas, se determina por medio de la distribución de Student, y el nivel de significación establecido, su confiabilidad.

3.3- ¿ Existen valores aberrantes ?

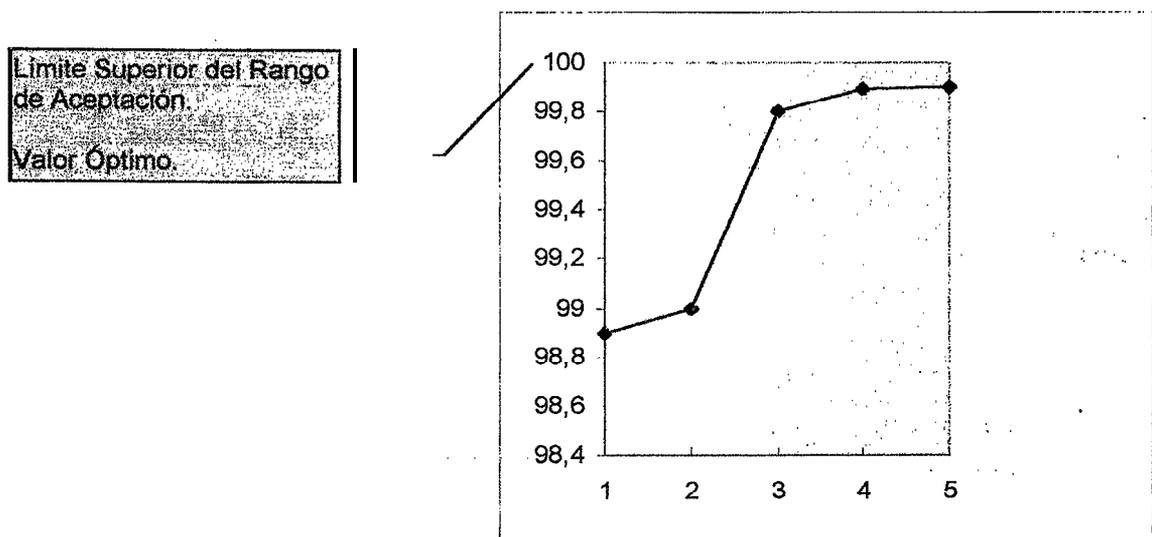
Se determina si existe en la serie de las “n” determinaciones de un parámetro obtenidas por el laboratorio destino, uno o más valores que se comporten diferentes al resto.

3.4.- ¿ Existen valores polarizados ?

El objetivo es estimar si los valores en la serie de las “n” determinaciones de un parámetro, tienden a concentrarse próximos a los extremos de las especificaciones.

3.5.- ¿ Existe tendencia en partidas sucesivas ?

El objetivo es determinar si existe tendencia hacia uno de los extremos en los resultados de los parámetros y en partidas sucesivas. Esto es favorable o desfavorable según la tendencia sea o no hacia un límite deseable u óptimo. Supongamos que un parámetro es aceptable si el resultado obtenido se encuentra en el rango 95 - 100, siendo 100 el valor óptimo; el siguiente gráfico muestra que la tendencia de los valores resultantes es beneficiosa ya que tienden hacia el límite deseable. Si la tendencia fuera en cambio hacia el límite inferior (95), a pesar de que los valores son aceptables, el riesgo proveedor sería mayor pues no tiende hacia el valor óptimo.

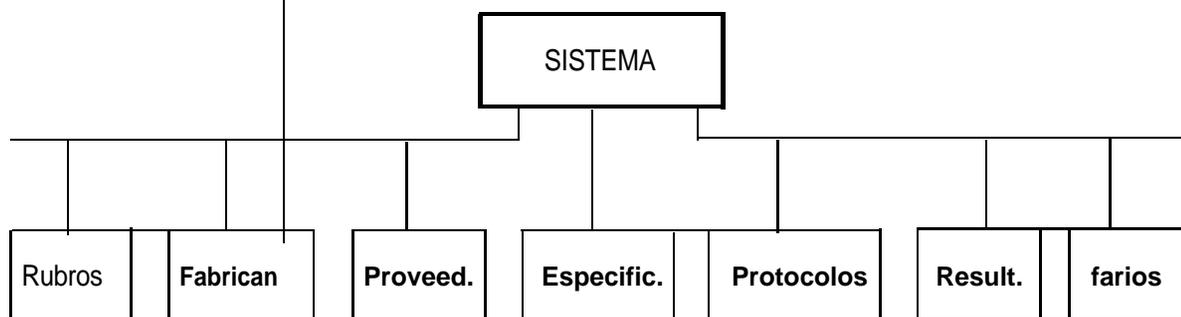


Según las consideraciones planteadas recientemente, se genera un ránking de confiabilidad para cada uno de los parámetros de las materias primas (el informe de ránking puede generarse por materia prima, por proveedor, fabricante o por rubro). El valor del Ránking oscila entre 0 y 5 [cuanto mayor es dicho valor, menor es el riesgo de aceptación] y se calcula según la siguiente tabla :

A evaluar	CUMPLE	NO CUMPLE
Valores dentro del rango especificado	1	0
Valores aportados por el proveedor similares a los determinados por el lab. PHARMA	1	0
Ausencia de valores aberrantes	1	0
NO se detectan valores polarizados	1	0
NO hay tendencia hacia los extremos en partidas sucesivas	1	0
Hay tendencia hacia un límite deseable u óptimo	1	0

4.- DESCRIPCIÓN del SISTEMA ()

Esquema general del sistema :



El diagrama presentado en la página siguiente, representa cómo se encuentran relacionados los módulos y archivos más significativos del sistema :

#	Rubros	Fabricantes	Proveedores	Especific.	proTocolos	resUlt.	Varios	Salir
Sist. de ANÁLISIS de CONFIABILIDAD y CERTIFICACIÓN de PROVEEDORES y FABRICANTES								
PROVEEDORES / Actualización y Consulta								
Código	1468							
Nombre	Proveedor ZZZ							
Rubro	220	Descrip: EXCIPIENTES						
<p>9 Cant. de Protocolos 2 Cant. de Prot. con aspectos INVÁLIDOS</p> <p>22,22 %</p>							A partir de Fecha 27/12/1994	
<input type="button" value="Prim."/> <input type="button" value="Prev."/> <input type="button" value="Sig."/> <input type="button" value="Ult."/> <input type="button" value="Busca"/> <input type="button" value="Alta"/> <input type="button" value="Modif."/> <input type="button" value="Baja"/> <input type="button" value="Impr."/> <input type="button" value="Salir"/>								
Busca y Selecciona un proveedor.								Ins

5.2. La siguiente pantalla, muestra el ingreso o consulta de protocolos de materias primas

#	Rubros	Fabricantes	Proveedores	Especific.	proTocolos	resUlt.	Varios	Salir	
Sist. de ANÁLISIS de CONFIABILIDAD y CERTIFICACIÓN de PROVEEDORES y FABRICANTES									
PROTOCOLOS / DROGAS / Actualización y Consulta									
Actualización y Consulta			Nro. Protocolo	R	316	Aprobado con Reclamo			
Cód. de RUBRO	220	Descrip: EXCIPIENTES							
Código (M.P.)	3422	Descrip:		ETANOL 96 % PV					
		Observ:		PUB. N° 5					
Fecha de Publicación de la Especificación	07/04/1995								
Nro. E.M.	16303	Fecha E.M.	27/12/1994	Nro. REMITO	6473	Fecha REMITO	26/12/1994		
Cód. PROVEEDOR	1468	Nombre: Proveedor ZZZ							
Cód. FABRICANTE	F9	Nombre: Fabricante XXX							
Cant. ENTREGADA	400,000	Unidad de MEDIDA		LT					
<input type="button" value="Parámetros >>"/>									
INICIO de Análisis	27/12/1994	FIN de Análisis		11/01/1995					
Observac.	CERTIFICADO DE ANALISIS. SI LOTE. H 44734								
Dictamen	APROBADO Q								
Analista	E. Collado					Cuaderno	67	Pág.	218
<input type="button" value="Prim."/> <input type="button" value="Prev."/> <input type="button" value="Sig."/> <input type="button" value="Ult."/> <input type="button" value="Busca"/> <input type="button" value="Alta"/> <input type="button" value="Modif."/> <input type="button" value="Baja"/> <input type="button" value="Impr."/> <input type="button" value="Salir"/>									
Actualización y Consulta de Valores Hallados de Parámetros de la Mat. Prima								Ins	

#	Rubros	Fabricantes	Proveedores	Especif.	proTocolos	resUlt.	Varios	Salir
Sist. de ANÁLISIS de CONFIABILIDAD y CERTIFICACIÓN de PROVEEDORES y FABRICANTES								
PROTocolos / DROGAS / Actualización y Consulta								
Actualización y Consulta			Nro. Protocolo	R	316	Aprobado con Reclamo		
Cód. de RUBRO	2.2.0	Descripción	EXCIPIENTES					
Código (M.P.)	3422	Descripción	ETANOL 96 % PV					
		Observación	PUB. N° 5					
Parámetros - Valores Hallados/según Certif.								
PARÁMETRO	Especificación					Hallado	Descripción	
Aspecto	Líquido incoloro, claro, móvil y volátil. Olor característico.					A	B	
Solubilidad	Misible con agua y Solventes Orgánicos.							
Identificación	Cumple					CUMPLE	CUMPLE	
Peso específico	0,812 - 0,816					0,812	0,812	
Acidez	Cumple					CUMPLE	CUMPLE	
Residuo no volátil	Máx 1 mg							
Sustancia insolubles en Agua	Cumple							
Aldehidos y otras sust. org.	Cumple					CUMPLE	CUMPLE	
Alcohol Amílico y sust. carb.	Cumple							
Componentes del Fusel oil	Cumple							
Acetona y 2-propanol	Cumple							
Metanol	Cumple					CUMPLE	CUMPLE	
<input type="button" value="Prim."/> <input type="button" value="Prev."/> <input type="button" value="Sig."/> <input type="button" value="Ult."/> <input type="button" value="Busca"/> <input type="button" value="Ara"/> <input type="button" value="Modif."/> <input type="button" value="Baja"/> <input type="button" value="Impr."/> <input type="button" value="Salir"/>								
Actualización y Consulta de Valores Hallados de Parámetros de la Mat.Prima								Ins

La segunda pantalla recientemente presentada, muestra cómo se ingresan y/o visualizan los valores hallados en el análisis por el laboratorio destino, y a continuación, cómo se lleva control de los valores del certificado de origen

(Como se dijo, no siempre el proveedor o el fabricante entregan el certificado de origen de algunos parámetros. En la pantalla que sigue, podemos ver que del parámetro " Identificación ", no se tiene conocimiento del resultado obtenido por el laboratorio del proveedor o del fabricante).

#	Rubros	Fabricantes	Proveedores	Especif.	proTocolos	resUlt.	Varios	Salir
Sist. de ANÁLISIS de CONFIABILIDAD y CERTIFICACIÓN de PROVEEDORES y FABRICANTES								
PROTocolos / DROGAS / Actualización y Consulta								
Actualización y Consulta			Nro. Protocolo	R	316	Aprobado con Reclamo		
Cód. de RUBRO	2.2.0	Descripción	EXCIPIENTES					
Código (M.P.)	3422	Descripción	ETANOL 96 % PV					
Parámetros - Valores Hallados/según Certif.								
PARÁMETRO	Val.CERTIFICADO	Descripción Certif.		¿Paramétrica?	VALID			
Aspecto	CUMPLE	CUMPLE		NO				
Solubilidad	CUMPLE	CUMPLE		NO				
Identificación				NO	SI			
Peso específico	0,8124	0,8124						
Acidez	0,004	0,004		NO	SI			
Residuo no volátil								
Sustancia Insolubles en Agua				NO				
Aldehidos y otras sust. org.				NO	SI			
Alcohol Amílico y sust. carb.	CUMPLE	CUMPLE		NO				
Componentes del Fusel oil								
Acetona y 2-propanol				NO				
Metanol	CUMPLE	CUMPLE		NO	S I			
<input type="button" value="Prim."/> <input type="button" value="Prev."/> <input type="button" value="Sig."/> <input type="button" value="Ult."/> <input type="button" value="Busca"/> <input type="button" value="Ara"/> <input type="button" value="Modif."/> <input type="button" value="Baja"/> <input type="button" value="Impr."/> <input type="button" value="Salir"/>								
Actualización y Consulta de Valores Hallados de Parámetros de la Mat.Prima								Ins

En las pantallas anteriores, deben ingresarse sólo los valores hallados, la descripción de parámetros tipo de parámetro; y control de validez los determina el sistema según las especificaciones ingresadas (en módulo de **Especificaciones**).

5.3. El que sigue, es el informe de evaluación de la materia prima TEXAPON JAN/VAR que provee "Proveedor ZZZ ". En éste se informa para cada uno de sus parámetros, el grado de confiabilidad asociado :

RESULT. \ Drogas \ Ranking según Mat.Primas

Pharma Argentina

Drogas	RANKING según MATERIAS PRIMAS informe ENERO 1997	10/01/1997
---------------	---	-------------------

Ráking Mat. Prima 3414 | - **TEXAPON JAN/AR**

Proveedor : P23 - **Proveedor ZZZ**

Fabricante : F3 - **Fabric. ARGENTINA S.A.**

Inicio : 15/06/1995

Última Entrega : 26/06/1996

Cant. Total de Entregas : 12

Detalle Ranking

Confiable	Parámetro	Ptos. Ranking	Cant. Analiz.	con Certif.	Últimos 5 valores Consecutivos Correctos PHARMA / Certif.Prov.	* Detalle Ranking			
						C	Ab	Pol	Te
(Conf.)	Aspecto	---	12	12	NO Paramétrico				
	Identificación	0	1	1	Pharma 98,1234 Certif. 98,1237	0			
(Conf.)	PHI	---	12	12	NO Paramétrico				
	Densidad	0	12	5	Pharma 30,3000 30,3000 28,6000 Certif. 25,0000 25,0000 25,9000	0			
	Viscosidad	0	12	12	Pharma 15,0000 16,0000 Certif. 10,0000 11,0000	0			
(Conf.)	Cloruro de Sodio	4	6	12	Pharma 10,0000 11,0000 12,0000 13,0000 14,0000 Certif. 15,0000 10,0000 11,0000 12,0000 13,0000	1	1	1	0
	Sulfatos de Sodio	0	3	12	Pharma Certif.	0			
(Conf.)	Valoración	5	12	10	Pharma 27,0000 27,5000 27,6000 28,2000 25,7500 Certif. 27,3000 27,8500 27,2000 29,0000 26,0000	1	1	1	1
(Conf.)	Control Higiénico	---	10	7	NO Paramétrico				

C = Indica si los valores de certificación aportados por el proveedor son confiables o NO según los valores obtenidos por el laboratorio. Nivel de significación 1,00 a 2colas de distribución y 4 grados de libertad) # **Confiable : C = 1 #**

* Ab : Indica si tiene o NO valores Aberrantes # **SIN valores aberrantes : Ab = 1 #**

* Pol : Indica si hay valores polarizados # **SIN valores polarizados : Pol = 1 #**

* Te = Indica si hay tendencia hacia un extremo o hacia un límite deseable # **SIN tendencia o tendencia correcta = 1 #**

Página : 1

6.- BENEFICIOS

6.1. Selección de mejores proveedores y materiales

Es sabido que no es posible mejorar lo que no ha sido medido, el sistema de Control Estadístico de Materiales trabaja entonces de acuerdo a un sistema de medición” y comparación de resultados analíticos con los de un modelo establecido.

Debido a que la base de datos se alimenta y actualiza constantemente, ‘se tiene acceso continuo a los indicadores de calidad de cada una de las materias primas, lo que determina además el desempeño de los proveedores.

El sistema permite efectuar una clara evaluación de los proveedores y de la calidad de cada una de las materias primas que proveen. Como no siempre se obtiene la calidad deseada de los proveedores que ofrecen los precios más elevados, el sistema de *Control Estadístico de Materiales* nos permite tomar decisiones racionales acerca de la elección de los proveedores, quienes pueden suplirnos de materiales confiables a precios justos.

:

6.2. Disminuir el tiempo dedicado al control de calidad

Ya que los procesos de control estadístico han sido incorporados en el sistema, podemos reducir la cantidad de análisis efectuados por el laboratorio.

6.3. Mejorar el “Control de procesos”

6.4. Establecer una mejor comunicación con los Proveedores

Los diversos informes que ofrece el sistema, permiten mostrarle claramente al proveedor cuál es su situación, qué materias primas deben mejorar su calidad, cuáles son los parámetros que presentan comunmente inconvenientes y cuáles son las materias primas que deben mantener su calidad.

7.- ‘RESUMEN

El Control Estadístico de Materiales es una evaluación global de proveedores y materias primas por medio de un sistema de computación que incluye procesos estadísticos. Los informes emitidos por éste permiten mejorar la calidad.

Sus principales tareas :

1. Establecer si una materia prima cumple o no las especificaciones para una determinación o variable específica.
2. Si la cumple, definir si los análisis del proveedor son confiables respecto a análisis realizados por el laboratorio de destino, con el objeto de decidir si los análisis de laboratorio por parte del proveedor (origen), pueden o no seguir aceptándose en el futuro.

3. A partir de los datos del laboratorio destino, definir si los valores de los diferentes parámetros, aún en el rango de aceptación tienden a estar cerca de los límites de aceptación o rechazo aumentando el "Riesgo Proveedor "
4. En base al análisis de confiabilidad efectuado y al nivel de confiabilidad alcanzado en cada parámetro de las materias primas recibidas por parte de los proveedores, poder decidir si se autoriza el Análisis/ Reducido y si el proveedor se encuentra en la categoría de :Potencial, Aprobado, Preferencial o Certificado.



8.- BIBLIOGRAFIA

- Comparative Data Analysis to Support Local Testing Authorization - D.W. Chapman
- Farmacopea USP III (Design and Analysis of Biological Assays)
- Manuales de referencia
- Reportes técnicos emitidos por J&J Corporate Quality Assurance.
- Reportes técnicos emitidos por Janssen Pharmaceutica Belgium.
- Reportes técnicos emitidos por Ortho Mc. Neil Pharmaceutical, U.S.A.

AUSPICIOS RECIBIDOS AL 24/04/97

Declaraciones de Interés

La Presidencia de la Nación declara: “de Interés Nacional”. Resolución S.G. N° 396.

La Presidencia del Senado de la Nación declara: “Del más alto interés para este cuerpo”. Decreto VSP-132/97

El Gobierno de la Ciudad de Buenos Aires declara: “de Interés para la Ciudad de Buenos Aires”. Decreto Na 376-GCBA-97.

Universidades e Institutos de Enseñanza

Universidad Champagnat

Universidad Nacional de Luján

Universidad Nacional de Río Cuarto

Universidad Nacional de Rosario

Universidad Nacional del Sur

Facultades

Universidad Católica Argentina, Facultad de Ciencias Económicas San Francisco.

Universidad Católica de La Plata, Facultad de Matemática Aplicada.

Universidad de Belgrano, Facultad de Ciencias Económicas.

Universidad de Belgrano, Facultad de Tecnología

Universidad de Buenos Aires, Facultad de Ingeniería

Universidad de Morón, Facultad de Ciencias Económicas.

Universidad Nacional de Entre Ríos, Facultad de Ciencias Económicas.

Universidad Nacional de Jujuy, Facultad de Ciencias Agrarias.

Universidad Nacional de Jujuy, Facultad de Ciencias Económicas.

Universidad Nacional de la Patagonia San Juan Bosco, Facultad de Ciencias Económicas.

Universidad Nacional de La Plata, Facultad de Ciencias Económicas.

Universidad Nacional de Mar del Plata, Facultad de Ingeniería.

Universidad Nacional de Río Cuarto, Facultad de Ciencias Exactas, Físico - Químicas y Naturales.

Universidad Nacional de Rosario, Facultad de Ciencias Económicas y Estadística.

Universidad Nacional de Rosario, Facultad de Ciencias Exactas Ingeniería y Agrimensura.

Universidad Nacional de San Juan, Facultad de Ingeniería.
Universidad Nacional de Santiago del Estero, Facultad de Ciencias Exactas y Tecnologías.
Universidad Nacional del Litoral, Facultad de Agronomía y Veterinaria
Universidad Nacional del Litoral, Facultad de Ciencias Económicas.
Universidad Tecnológica Nacional, Facultad Regional Avellaneda.
Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay.

Cámaras, Asociaciones y Entidades Empresarias

Asociación Argentina de Marketing	AAM
Asociación Argentina de Televisión por Cable	ATVC
Asociación de Bancos de la República Argentina	ABRA
Asociación de Bancos Públicos y Privados de la República Argentina	ABAPPRA
Asociación de Fábricas Argentinas Terminales de Electrónica	AFARTE
Asociación de Industriales Metalúrgicos de la República Argentina	ADIMRA
Asociación Marketing Bancario Argentino	AMBA
Cámara Argentina de Industrias Electromecánicas	CADIEM
Cámara Argentina de Industrias Electrónicas	CADIE
Cámara Argentina de máquinas de oficina, comerciales y afines	CAMOCA
Cámara de Comercio Argentino - Brasileña	
Cámara de Comercio Argentino - Chilena	CACH
Cámara de Comercio Argentino - Uruguaya	CCAI
Consejo Profesional en Ciencias Informáticas	CPCI
Fundación para la Investigación el Desarrollo y la Aplicación de las Nuevas Tecnologías de la Informática	FIDANTI
Instituto de Desarrollo Empresario Bonaerense	IDEB
Instituto de Ingenieros en Electricidad y Electrónica	IEEE
Instituto para el Desarrollo Empresarial de la Argentina	IDEA
Sociedad Argentina de Informática e Investigación Operativa	SADIO
Unión Argentina de Entidades de Servicios	UDES
Unión Industrial Argentina	UIA

AUTORES

APELLIDO, NOMBRE	ENTIDAD	PAG.
ALLES, A.	Top Management	104
ANSALDI, J. C.	Universidad Tecnológica Nacional	16
ARNONE, L.	Universidad Nacional de Mar del Plata	26
BALBI, R.	Universidad Tecnológica Nacional	257
BALLARIN, V.	Universidad Nacional de Mar del Plata	143
BELFORTI, G.	Universidad Nacional de Mar del Plata	26
BENITEZ, S.	Universidad de Buenos Aires	272
BERTONE, E.	Universidad Nacional de La Pampa	117
BEVILACQUA, R.	Universidad de Buenos Aires	272
CALIGARIS, M.	Universidad Tecnológica Nacional	72; 57
CALIGARIS, R.	Universidad Tecnológica Nacional	72; 57
CARTAGENOVA, S.	Universidad F.A.S.T.A.	297
CASAL, R.	Universidad Nacional del Sur	242
CAVALIE, P.	Universidad Nacional de San Luis	199
CORRAL, R.	Universidad Nacional del Sur	2 4 2
DASSO, A.	Universidad Nacional de San Luis	283
DELRIEUX, C.	Universidad Nacional del Sur	128; 164
DI PEGO, C.	Universidad Nacional de La Pampa	37
FORNARI, J.	INGAR, Instituto de Diseño y Desarrollo	175
FUNES, A.	Universidad Nacional de San Luis	228
GARCIA MARTINEZ, R.	Buenos Aires Institute of Technology // University of Buenos Aires	97; 184; 191
GAYOSO, C.	Universidad Nacional de Mar del Plata	26; 154
GIULIETTI, A.	Universidad Nacional del Sur	128
GONZALEZ, M.	Universidad Nacional de Mar del Plata	143
GONZALEZ, C.	Universidad Nacional de Mar del Plata	26
GONZALEZ, II.	Universidad Tecnológica Nacional	257
GRAMAJO, E.	Buenos Aires Institute of Technology	97
HAVLIK, J. M.	Universidad Nacional de Buenos Aires	87
ISERN, G.	Universidad Central de Venezuela	1
KAVKA, c.	Universidad Nacional de San Luis	228

LAFUENTE, G.	Universidad Nacional de La Pampa	117
LISNIZER, M.	Pharma Argentina	297
LOPEZ, E.	Instituto Superior de Electrónica de Entre Ríos	16
LOPEZ, N.	Universidad Nacional del Sur	242
LOVOTTI, L.	Universidad Nacional del Sur	242
LUCERO, M.	Universidad Nacional de San Luis	283
MANSILLA, G.	Universidad Tecnológica Nacional	72
MARTINEZ, E.	INGAR, Instituto de Diseño y Desarrollo	175
MOLER, E.	Universidad Nacional de Mar del Plata	143
MONTEJANO, G.	Universidad Nacional de San Luis	283
NAGEL, S.	University of Illinois	4 8
NECCO, ,C.	Universidad Nacional de San Luis	213
NICOLAU, S.	Universidad Nacional de La Pampa	117
OLMO, D.	Universidad Nacional de Mar del Plata	143
OLSINA, L. A.	Universidad Nacional de La Pampa	117
PEREZ, G.	INGAR, Instituto de Diseño y Desarrollo	175
PESSANA, F.	Universidad Nacional de Mar del Plata	143
PICA, A.	Universidad Tecnológica Nacional	257
PRUZZO, V.	Universidad Nacional de la Pampa	37
QUIJANO, A.	Universidad Nacional de Mar del Plata	154
QUINTAS, L.	Universidad Nacional de San Luis	199; 213; 228
RODRIGUEZ, G.	Universidad Tecnológica Nacional	57
ROSSI, B.	Buenos Aires Institute of Technology // Universidad Tecnológica Nacional	97; 257
SANCHEZ, A.	Universidad Nacional de San Luis	283
SEOANE, J.	Universidad de Buenos Aires	272
STELLI, I.	Universidad Nacional de Mar del Plata	154
UZAL, R.	Universidad Nacional de San Luis	283
VILLAPOL, M. E.	Universidad Central de Venezuela	1
WAINER, G.	Universidad de Buenos Aires	272
WELCH, D. A	Universidad Nacional de San Luis	199
ZITO, E.	Universidad Nacional del Sur	242



Si no tiene la solución, forma parte del problema.

Por eso en TTI Professional Services trabajamos junto a nuestros más de 400 profesionales para crear las respuestas específicas a las necesidades técnicas de cada uno de nuestros clientes, desarrollando soluciones de Outsourcing, Network & Systems Management, Consultoría Tecnológica, Desarrollos a medida, Imágenes & Data Center y Data Center Inteligente de Datos y TTI 2000, la respuesta para resolver los problemas informáticos frente al cambio de milenio.

Y brindamos resultados confiables, para que cuando obtengan la solución ya no formen parte de ningún problema.



Soluciones confiables

Av. Pueyrredón 1770 (1119) Bs. As. - Argentina

Tel.: (54-1) 821-6090/99 - Fax: (54-1) 962-4502

E-mail: info@tti.com.ar