

Colección **Actualizaciones Académicas**

# Actualización Académica en enseñanza de Educación Tecnológica con foco en programación

Módulo 5: **Estrategias para la enseñanza de la  
programación III**



## Índice

<b>Clase 1: Presentación. Introducción a variables .....</b>	<b>3</b>
<b>Clase 2: Contadores .....</b>	<b>15</b>
<b>Clase 3: Proyecto integrador de programación (Parte 1) .....</b>	<b>30</b>
<b>Clase 4: Proyecto integrador de programación (Parte 2) .....</b>	<b>49</b>

## Módulo 5: Estrategias para la enseñanza de la programación III

### Clase 1: Presentación. Introducción a variables

#### Bienvenidos y bienvenidas

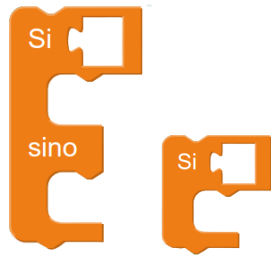

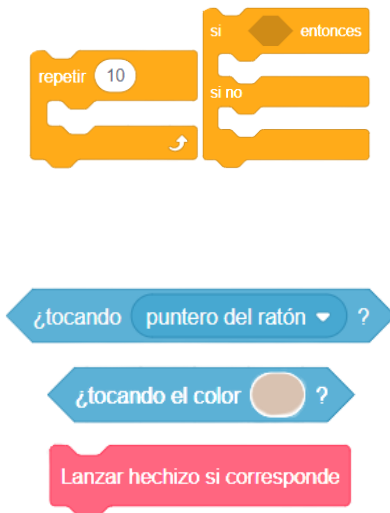
Sean bienvenidos y bienvenidas al tercer módulo de Estrategias para la enseñanza de la programación. En este módulo continuaremos desarrollando proyectos de programación en Scratch con el objetivo de incorporarles nuevas funcionalidades como la posibilidad de contar los puntos en un videojuego. Para alcanzar este objetivo será necesario conocer cómo trabajar con datos en un programa y conservarlos durante toda su ejecución.

En esta primera clase motivaremos e introduciremos las **variables** como herramienta de programación para almacenar datos durante la ejecución de un programa.

#### Repaso de conceptos

En el módulo anterior comenzamos a trabajar con el entorno Scratch donde introdujimos nuevos conceptos de programación, como los eventos, y reforzamos conceptos previos, como la repetición, los procedimientos y la alternativa condicional. Hagamos un breve recorrido por los conceptos de programación que abordamos en los módulos anteriores:

PROYECTO/ACTIVIDAD	CONCEPTOS PRINCIPALES TRABAJADOS	BLOQUES
Desafíos en Pilas Bloques	<ul style="list-style-type: none"> <li>Comandos básicos</li> <li>Repetición</li> <li>Procedimientos</li> <li>Alternativa condicional</li> </ul>	

		
Scratch con tarjetas (tanda 1)	<ul style="list-style-type: none"> <li>Comandos básicos</li> <li>Eventos</li> </ul>	
Scratch con tarjetas (tanda 2)	<ul style="list-style-type: none"> <li>Repetición simple</li> <li>Alternativa condicional</li> <li>Sensores</li> <li>Procedimientos</li> </ul>	

## Actividad 1

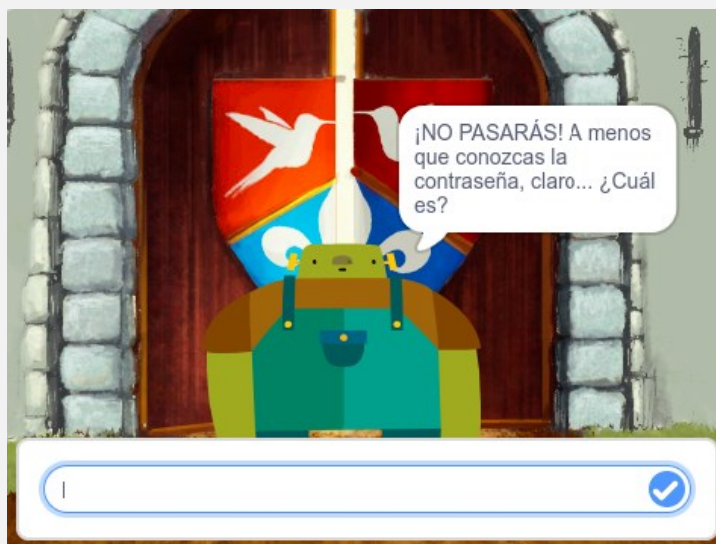
Vamos a comenzar este módulo con un proyecto en Scratch en el que deberán programar diferentes momentos de una conversación con el guardia de un castillo para poder ingresar. En este primer desafío deberán descubrir la contraseña para ingresar al castillo, poniendo el foco en las herramientas que nos brinda el entorno para implementar esta parte de la conversación. En los siguientes desafíos deberán incorporar nuevas partes de la conversación.

## Parte 1



### El castillo exclusivo, parte 1

Ingresen a [este proyecto](#) de Scratch, presionen sobre la bandera verde y prueben algunas frases para que el guardia les habilite el ingreso al castillo.



Luego de unos intentos, revisen el código del programa presionando sobre el botón “Ver dentro” (o “Reinventar” si iniciaron sesión con su cuenta para que además se les guarde una copia del proyecto) para averiguar la frase correcta. Una vez que la descubran, ejecuten nuevamente el proyecto y prueben la frase para entrar al castillo.



Es importante que **tengan una copia** de este proyecto en su repositorio seleccionando el botón “Reinventar” o guarden una copia en su computadora una vez que hayan terminado (seleccionando en el menú “Archivo” la opción “Guardar en tu computadora”), ya que **lo necesitarán en los siguientes desafíos**.



Registren en sus bitácoras las respuestas a las siguientes preguntas:

- ¿Les resultó fácil descubrir la frase secreta? ¿Qué parte del programa tuvieron que revisar con mayor detalle? ¿Qué cosas les facilitaron la búsqueda?

- ¿Cómo funciona el procedimiento “Controlar ingreso al castillo”?
- ¿Cuáles son los nuevos bloques que aparecen y cuál es su propósito? ¿Cómo funcionan?
- ¿Qué relación pueden establecer entre la forma ovalada del bloque “respuesta” y su propósito?

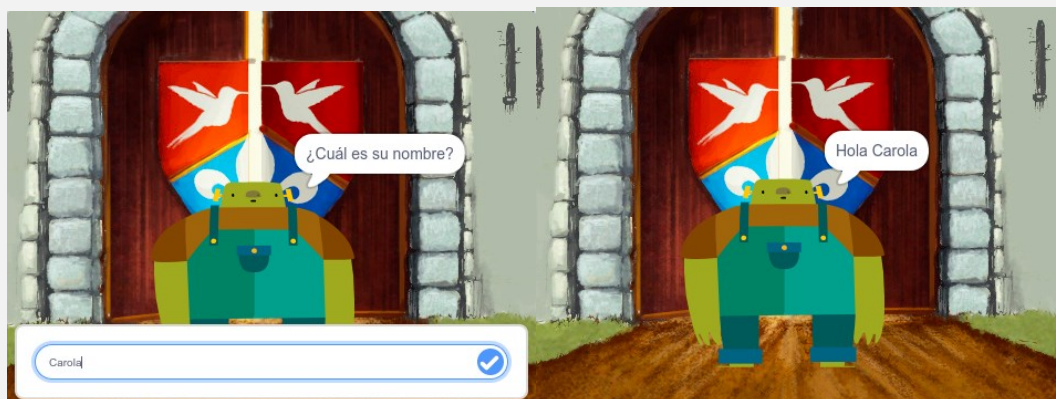
## Parte 2

Vamos a continuar con el proyecto anterior, aplicando los nuevos bloques para resolver un nuevo desafío.



### El castillo exclusivo, parte 2

Abran la primera parte del proyecto y modifiquen el programa para que el guardia les pida su nombre y lo utilice para saludarlos antes de pedirles la contraseña.



**¿Cómo podemos construir un saludo personalizado?** Explore la categoría *Operadores* y busquen algún bloque que les permita combinar palabras para construir un saludo como “Hola, Juanita”. Tengan en cuenta que el saludo tiene fija la palabra “Hola” y lo que varía es el nombre que ingresamos como respuesta. Una vez que logren construir el saludo deberían utilizarlo dentro del bloque “Decir”.



Registren en sus bitácoras las respuestas a las siguientes preguntas:

- ¿Cuál es el objetivo de la actividad?
- ¿Pudieron construir el saludo comenzando con la palabra “Hola” junto con el nombre? ¿Qué bloque/s podemos utilizar? ¿Son necesarios para resolver el desafío?
- ¿Con qué obstáculos se encontraron?
- ¿Utilizaron procedimientos? ¿Cómo modificaron la estrategia?

## Parte 3



### El castillo exclusivo, parte 3

Abran la segunda parte del proyecto y modifiquen el programa para que el guardia además mencione su nombre cuando se les niega o concede el ingreso al castillo. ¡No vale preguntar el nombre más de una vez!

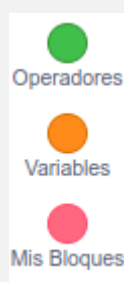




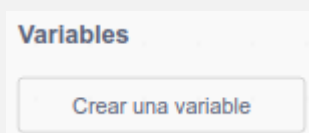
Las **variables** son herramientas de programación que sirven para *almacenar datos* durante la ejecución de un programa. Además, los datos recordados podrán ser *utilizados* y *modificados* en cualquier momento del programa.

### ¿Cómo creamos una variable?

1. Debemos ir a la categoría de bloques “Variables”:



2. Seleccionamos el botón “Crear una variable”:



3. Debemos ingresar el nombre que le daremos a la variable. Como siempre que elegimos nombres, es importante que refleje claramente cuál es el dato que queremos almacenar. En este caso, el nombre del usuario y presionar sobre el botón “De acuerdo”:



Nueva variable ×

Nuevo nombre de variable:

nombre del usuario

☒ Para todos los objetos
 ☐ Sólo para este objeto

Cancelar

De acuerdo

4. Veremos en el panel izquierdo que aparece la variable que acabamos de crear:

Crear una variable

☐

nombre del usuario

### ¿Cómo almacenamos datos en una variable?

El bloque que nos permite almacenar un valor en una variable es el bloque **“Fijar”** (o **“Dar”**, según el idioma elegido):

fijar nombre del usuario a 0

dar a nombre del usuario el valor 0

A la acción de almacenar un dato en una variable también se le conoce como **“asignación”**. Observen que este bloque, como representa una acción, es un **comando** y, por lo tanto, tiene la forma que lo permite encastrarlo con otros comandos.

### ¿Cómo utilizamos una variable?

Para acceder o usar el dato almacenado de una variable basta con usar el bloque ovalado que lleva su nombre:

nombre del usuario

Observen que la forma del bloque es la misma que la del bloque **“Respuesta”**, porque representa un dato, en este caso, el nombre del usuario/a que quiere entrar al castillo.



Luego del encuentro sincrónico, registren en sus bitácoras:

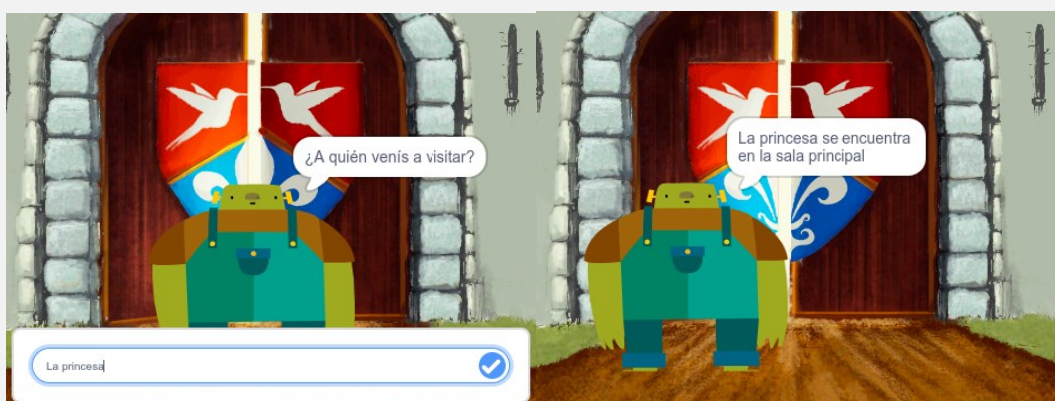
- **¿Cuál es la importancia de resolver los dos primeros desafíos antes de presentar el concepto de variable?**
- **¿Por qué surge la necesidad de introducir las variables? ¿Con qué idea se introducen?**

## Parte 4



### **El castillo exclusivo, parte 4**

Agreguen al proyecto la posibilidad de que el guardia les pregunte con quien desea hablar luego de pedirles su nombre. Si se les permite el ingreso, el guardia les debe indicar que la persona con la que desea hablar se encuentra en la sala principal.



## Actividad 2

En los módulos anteriores presentamos las actividades sin computadoras como un recurso complementario para trabajar ideas o contenidos específicos de programación. A continuación, les presentamos dos propuestas posibles, sin computadoras, para introducir el concepto de “variable”.



- **Actividad 1:** [Actividad sin computadoras - Variables: Recuerden la contraseña](#)
- **Actividad 2:** [Actividad sin computadoras - Variables: Próximo cumple](#)



Registren en sus bitácoras las respuestas a la siguiente situación: si tuvieran que incluir estas dos actividades en la secuencia propuesta en esta clase, ¿qué modificaciones le harían a la secuencia y/o a las actividades, ya sean de las nuevas o las que ya estaban incluidas en la secuencia? ¿Por qué realizarían esas modificaciones?



**Encuentro sincrónico**

**Fecha del encuentro:** DD/MM/AAAA

En este encuentro sincrónico abordaremos entre todos hasta la **parte 3** de la Actividad 1, centrándonos en cómo motivar e introducir el concepto de variable. Además, conversaremos sobre las actividades sin computadoras y cómo articularlas con la secuencia de la clase.

## Actividad 3 (entrega obligatoria)



### Actividad individual con entrega obligatoria

Diseñen e implementen un proyecto en Scratch para introducir el concepto de “Variable”. El proyecto debe estar pensado para ser construido incrementalmente por partes con el acompañamiento de un/a docente. Se puede partir desde un proyecto nuevo o la base de uno ya creado.

El formato de entrega de esta actividad será una copia de la [plantilla de diseño de proyecto](#) más un enlace a la implementación en Scratch. En la plantilla encontrarán una serie de secciones y preguntas que las y los guiarán para que tengan en cuenta las consideraciones que se vienen trabajando en el trayecto.

## Reflexión de cierre



El recorrido de esta clase debiera permitirles:

- A. Reconocer que los programas tienen la capacidad de **conservar o almacenar** datos.
- B. Abordar el concepto de **variable** como herramienta para almacenar o conservar datos durante la ejecución de un programa.
- C. Identificar que, antes de incorporar una variable al programa, hay que crearla (o definirla) y en esta instancia es fundamental elegir un nombre que represente aquello que guardaremos o conservaremos en la variable.
- D. Diferenciar entre la asignación y uso de una variable: para utilizar una variable tenemos que crearla (definirla), indicarle (asignarle) el dato que queremos almacenar

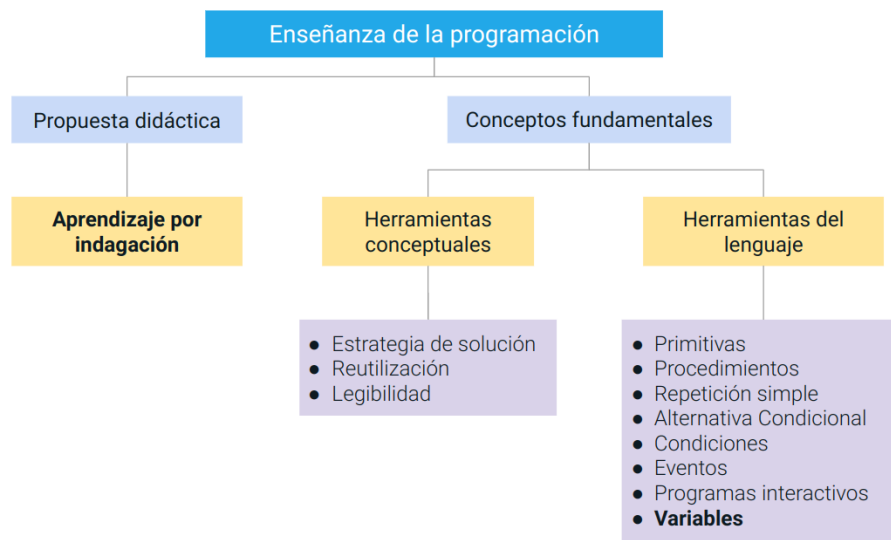
y luego utilizarla en algún bloque que consulte (lea) el dato resguardado, por ejemplo, el bloque “Decir”.

- a. Reconocer que almacenar un dato en una variable es una **acción** que se llama “asignación” (decimos “asignar un dato a una variable”) y que se realiza con un **comando**, por ejemplo, fijar/dar en Scratch.
  - b. Comprender que para usar o recuperar el dato almacenado en una variable debemos utilizar el bloque que lleva su nombre.
- E. Comprender que una variable puede conservar un dato a la vez y que cada vez que se le asigna un dato nuevo se reemplaza el dato que tenía.
- F. Reconocer que los bloques que representan datos, como el bloque *respuesta* o las variables, tienen una forma diferente a los bloques de comandos ya que sus propósitos son diferentes: los primeros son datos que por sí solos no tienen ningún efecto, sino que deben ser utilizados dentro de otros bloques mientras que los bloques de comandos representan acciones que se pueden combinar unas debajo de otras.



### Marco conceptual

Incorporamos las **variables** como una nueva herramienta de los lenguajes de programación para almacenar datos y utilizarlos durante la ejecución de un programa:



## Bibliografía de referencia

Balch, C.; Chung, M. y Brennan K. (2014). [Computación creativa](#). Escuela de Graduados en Educación de la Universidad de Harvard.

Fundación Sadosky (2018). [Ciencias de la computación para el aula: 2do ciclo de primaria](#), Capítulo 4: Procedimientos. Buenos Aires: Fundación Sadosky.

## Créditos

Autores: Fundación Sadosky

Fundación Sadosky (2022). Clase 1: Presentación. Introducción a variables. Estrategias para la enseñanza de la programación III. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.



Esta obra está bajo una licencia Creative Commons  
Atribución-NoComercial-CompartirIgual 3.0

## Módulo 5: Estrategias para la enseñanza de la programación III

# Clase 2: Contadores

## Introducción

En esta clase vamos a programar un pequeño videojuego desde cero en Scratch en el que dos personajes deberán atrapar objetos que aparecen en diferentes lugares del escenario para ganar puntos. Para esto, iremos resolviendo una secuencia de pequeños desafíos que nos permitan construir incrementalmente el proyecto. El objetivo es recuperar el concepto de **variable** y utilizarlo para almacenar los puntos de cada jugador.

## Actividad 1



A continuación, les presentamos algunas capturas de videojuegos. En cada una identifiquen qué datos se están almacenando durante la ejecución.



Super Mario Bros



**Clash Royale - Subway Surfers**



### Actividad en foro

Compartan en el foro los datos que identificaron en cada una de las capturas de los videojuegos.

## Actividad 2

En esta actividad comenzarán a programar el videojuego desde cero. El proyecto consta de diez pasos separados en siete bloques que llamaremos “Desafíos”.



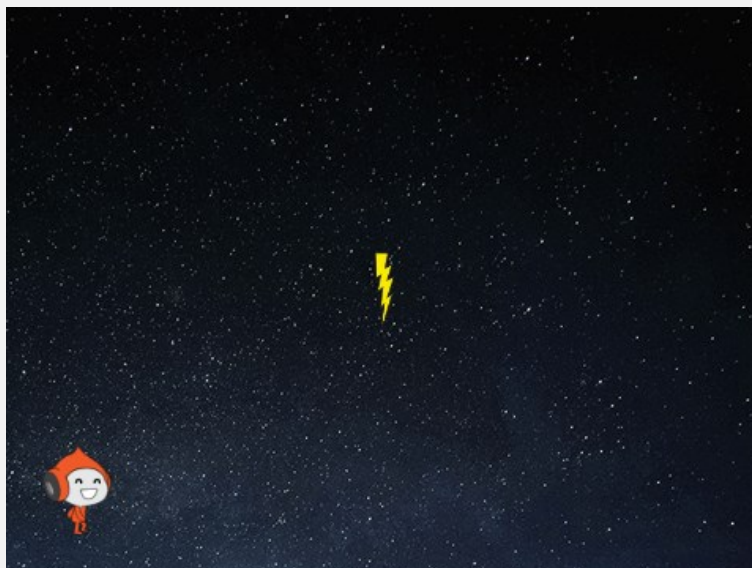
## Desafío 1



**Creen un nuevo proyecto** en Scratch y agréguenle:

- El personaje que más les guste. Pueden eliminar el personaje existente y agregar uno nuevo o modificar el disfraz del personaje existente.
- El objeto para capturar (el objetivo).
- Un fondo al escenario.

Por ejemplo, un extraterrestre (el personaje) y un rayo (el objeto a capturar) con un fondo estrellado:



**Programen el personaje** para que se mueva con el teclado en cuatro direcciones: arriba, abajo, izquierda y derecha.



En los primeros desafíos nos vamos a preocupar por programar a un solo personaje. Luego agregaremos un segundo personaje.

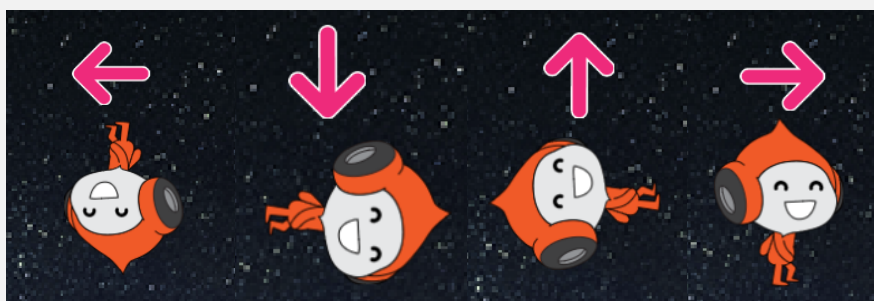


Para facilitar la jugabilidad se pueden modificar el tamaño de los objetos, por ejemplo, en la figura el extraterrestre tiene tamaño 40 y el rayo 30.



Les sugerimos utilizar el bloque “fijar estilo de rotación a izquierda-derecha” al momento de presionar la bandera verde para que el personaje no rote cuando cambia su dirección.

fijar estilo de rotación a en todas las direcciones ▼



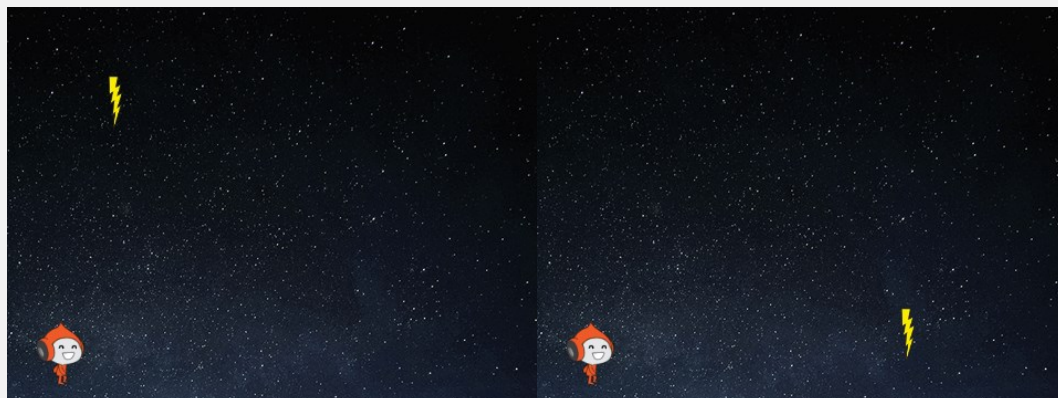
Registren en su bitácora:

- ¿Qué conceptos de programación se recuperan?
- ¿Qué características del entorno tendrían en cuenta al trabajar con esta actividad?



### PASO 1

Programen el **objetivo** para que se ubiquen en posiciones aleatorias del escenario al presionar la bandera verde.



### PASO 2

Programen el **objetivo** para que aparezca en *20 lugares aleatorios* del escenario cada medio segundo al presionar la bandera verde.

## Desafío 2



Programen el **objetivo** para que aparezca en lugares aleatorios del escenario *durante toda la ejecución* del juego.



¿Cuántas veces deberíamos repetir esto? Les proponemos que exploren Scratch en busca de un bloque que les permita repetir esto indefinidamente.



Registren en su bitácora:

- ¿Qué bloques permiten resolver el desafío? ¿Cómo se motiva su necesidad? ¿Con qué bloques conocidos lo relacionarían?
- Identifiquen qué problemas intermedios se propusieron para llegar a esta nueva forma de repetición: ¿qué recuperaba cada uno?, ¿qué modificaba o introducía cada uno?

## Desafío 3



Modifiquen la programación del **objetivo** para que cambie su posición aleatoriamente durante todo el juego **cuando sea atrapado por el personaje** en vez de cambiar cada medio segundo.



¿Cuándo se tiene que reubicar el objetivo ahora? ¿Cuándo se reubicaba antes? ¿Qué parte del programa tendrán que modificar? ¿Qué bloques nos provee el entorno para saber si estamos en la situación en la que se debe reubicar el personaje? ¿Qué significa que el objetivo haya sido atrapado?



Para probar si funciona correctamente presionen la bandera verde y muevan al personaje con el teclado. Al atraparlo debería cambiar la ubicación del objetivo.



Registren en su bitácora:

- ¿Qué bloques utilizaron para programar la colisión entre objetos?
- Identifiquen qué problemas intermedios se propusieron para llegar a resolver el problema de la colisión: ¿qué recuperaba cada uno?, ¿qué modificaba o introducía cada uno?

## Desafío 4



Agreguen al videojuego la posibilidad de contar los puntos del personaje, teniendo en cuenta que por cada objetivo atrapado se acumula un punto.



¿Qué herramienta de programación podemos usar para almacenar los puntos acumulados?

Teniendo en cuenta que al capturar el objetivo sumamos un punto, ¿Qué bloques podemos utilizar para modificar el puntaje? Exploren los bloques asociados.



¿Qué sucede con los puntos cuando volvemos a ejecutar el programa? ¿Qué debería pasar realmente?



Registren en su bitácora:

- ¿Qué bloques usaron para sumar puntos y para volver a cero el puntaje al empezar el juego? ¿Cómo se comporta cada uno?
- ¿Programaron el personaje o el objetivo? ¿Qué deberían tener en cuenta para programar cada uno?



## Contadores

Aprovechando que las variables pueden almacenar datos, son útiles cuando necesitamos llevar la **cuenta** de algo, en este caso, los puntos ganados (que no son más que la *cantidad* de objetivos capturados). En estos casos, decimos que una variable es un **contador**.

### ¿Cómo funciona un contador?

El objetivo es actualizar el valor de una variable, como si estuviéramos llevando la cuenta con los dedos de las manos: cuando una nueva ocurrencia sucede (por ejemplo, cuando capturamos un objetivo en el juego) “levantamos un dedo más”, es decir, actualizamos el valor de la variable sumándole uno. Para realizar esta acción de contar un punto, tenemos disponible el bloque **cambiar/sumar**:



Cada vez que se ejecuta este comando, la variable se actualiza según el valor indicado en el bloque.

### ¿Qué sucede con los puntos cuando volvemos a ejecutar el programa?

Cada vez que ejecutamos el programa, el valor almacenado en la variable no se restablece automáticamente y, por lo tanto, es necesario realizar esto en algún momento del programa. En este caso, debemos restablecer la cantidad de puntos a cero cada vez que presionamos la bandera verde, o dicho de otra manera, debemos **inicializar la variable** en cero. Para lograr esto debemos utilizar el bloque **dar/fijar** que vimos en la clase anterior.



### Para pensar...

¿Qué valor está almacenado en la variable *puntos de Nano* después de la ejecución en cada caso?



Lo primero que podemos notar es que ambos programas comienzan *inicializando* la variable puntos en cero. Luego, ambos repiten tres veces un bloque diferente: **dar** y **sumar**. En el primer caso, recordemos que el bloque dar/fijar reemplaza el valor almacenado por el valor indicado en el bloque, entonces en cada repetición se almacena el número 10. Por lo tanto, luego de ejecutar el programa la variable va a almacenar el valor 10 (diez).

En el segundo caso, recordemos que el efecto del bloque sumar/cambiar es actualizar el valor de una variable en función del valor que tiene almacenado. Por lo tanto, en cada repetición se actualiza el valor de la variable puntos sumándole 10, dando como resultado que al finalizar la ejecución la variable tenga el valor 30: primero se le suman 10 (y como está inicializada en cero va a almacenar 10), en la segunda repetición almacena 20 y en la tercer repetición almacena 30.



## Clase de consultas



### Consultas sincrónicas

Fecha del encuentro: **DD/MM/AAAA**

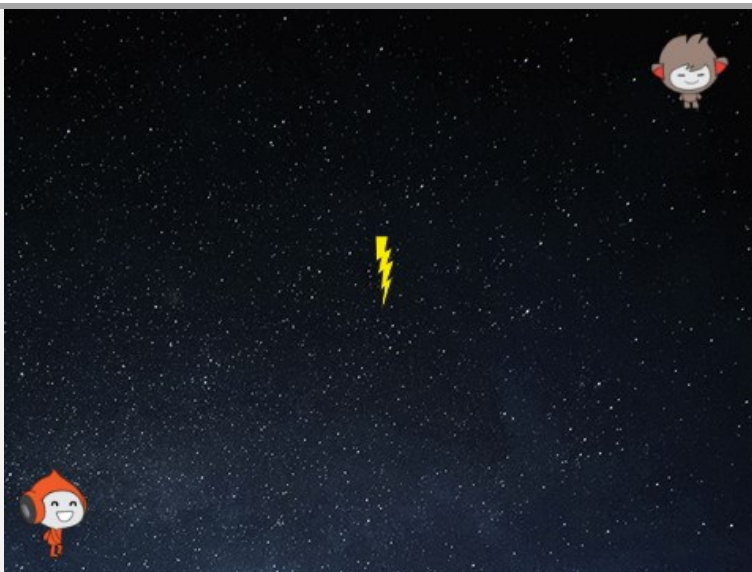
Durante la clase de consultas nos centraremos en cómo presentar y trabajar las variables como contadores. Además, repasaremos los objetivos principales de cada paso y conversaremos sobre cómo trabajarlas en el aula, diferentes formas de resolverlas y las dificultades con las que se hayan encontrado.

## Desafío 5



Llegó el momento de agregar el segundo personaje al juego. Para esto, deberán hacer crecer el programa e introducir las modificaciones necesarias para:

- Incorporar un nuevo personaje al juego que se mueva con el teclado, utilizando teclas diferentes a las del personaje existente (por ejemplo, A-S-D-W).
- Hacer que el objetivo se reubique cada vez que sea alcanzado por cualquiera de los dos personajes.
- Contar los puntos del nuevo personaje.



Registren en su bitácora:

- ¿Cómo utilizaron la alternativa condicional? ¿Qué otras combinaciones podrían realizar?
- ¿Cómo utilizaron las variables para contar los puntos del nuevo personaje?
- ¿Identificaron problemas parecidos a otros que ya habían resuelto durante este proyecto? ¿Se ayudaron con estos para resolver los problemas nuevos?
- Valoren (positiva y negativamente) el hecho de haber comenzado la programación del juego con un solo personaje con desafíos muy guiados y, recién en este punto, agregar el segundo con más libertad.

## Desafío 6



Agreguen al videojuego dos nuevos objetivos que cambien aleatoriamente su ubicación al ser capturado por alguno de los personajes teniendo en cuenta que uno debe **sumar 3 puntos** y el otro **restar 1 punto**.



*¿Qué significa restar un punto si solo tenemos el bloque “sumar”?*

Podemos utilizar el mismo bloque pero indicando un número negativo. En este caso, como queremos “restar un punto” podemos indicar “-1” como valor en el bloque.



Registren en su bitácora:

- ¿Qué **estrategias** pudieron reutilizar de los desafíos anteriores?
- ¿Qué bloques usaron para cambiar el puntaje con los nuevos objetivos? ¿Cómo utilizaron los bloques para cambiar el puntaje con los nuevos objetivos?

## Desafío 7



Implementen la finalización del juego a partir de una propuesta propia o tomando como sugerencia alguna de las siguientes. Por ejemplo, el juego puede terminar:

- Cuando alguno de los personajes captura una cantidad definida de objetivos.
- Cuando alguno de los personajes alcanza una cantidad definida de puntos.
- Cuando se captura un total definido de objetivos entre ambos personajes.
- Cuando se alcanza una cantidad de tiempo.

Además, registren en la bitácora:

- ¿Qué variables utilizaron? ¿Cuál es su propósito? ¿Cómo las utilizaron?



### Actividad individual con entrega obligatoria

Deberán entregar un enlace a su bitácora que incluya un enlace al proyecto completo en la página de Scratch (o el archivo del proyecto para descargar) para que el/la tutor/a pueda evaluar la implementación en función de algunos criterios.

## Reflexión de cierre



El recorrido de esta clase debiera permitirles:

- A. Recuperar una estrategia para detectar colisiones entre objetos.
- B. Comprender que las variables se pueden utilizar para contar o acumular valores, en este caso, un puntaje o la cantidad de objetivos capturados.
- C. Conocer y utilizar el bloque “Cambiar/Sumar” para actualizar el valor almacenado en una variable.
- D. Recuperar y reforzar el desarrollo incremental de un proyecto de programación por objetivos.

## Bibliografía de referencia

Balch, C.; Chung, M. y Brennan, K. (2014). [Computación creativa](#). Escuela de Graduados en Educación de la Universidad de Harvard.

Fundación Sadosky (2018). [Ciencias de la computación para el aula: 2do ciclo de primaria](#), Capítulo 4: Procedimientos. Buenos Aires: Fundación Sadosky.

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto: Fundación Sadosky (2023). Clase 2: Contadores. Estrategias para la enseñanza de la programación III. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.



Esta obra está bajo una licencia Creative Commons

Atribución-NoComercial-CompartirIgual 3.0

## Módulo 5: Estrategias para la enseñanza de la programación III

### Clase 3: Proyecto integrador de programación (Parte 1)

#### Introducción

El objetivo de esta clase es comenzar a diseñar un proyecto de programación que integre una selección de los conceptos trabajados hasta el momento, que se convertirá en el trabajo final del módulo. En esta clase vamos a presentar instrumentos para organizar los contenidos y saberes involucrados en el proyecto, definir los objetivos de aprendizaje y explicitar aquellas evidencias (o indicios) que relevaremos de las actividades. El objetivo de las actividades de esta clase es que puedan definir los lineamientos principales de su trabajo final con el acompañamiento de su tutor/a. Finalmente comenzaremos a diseñar las actividades en función de los puntos anteriores.



Para el diseño del proyecto vamos a retomar el enfoque propuesto en el módulo *Estrategias para la Enseñanza de la Programación I* para diseñar una actividad de programación donde primero nos concentramos en definir los objetivos de aprendizaje, es decir lo que queremos que nuestros estudiantes aprendan al realizar el proyecto, luego para cada uno de los objetivos establecimos evidencias que den cuenta de los aprendizajes logrados y finalmente diseñamos una secuencia de desafíos que tengan como propósito que nuestros/as estudiantes puedan alcanzar esos objetivos.



“Pensarnos como profesionales, ejecutantes virtuosos de la partitura del currículum, es un primer paso para tomar decisiones conscientes que nos ayuden a poner en primer plano los aprendizajes que queremos generar y la experiencia que tienen que recorrer los alumnos para llegar ahí”.

Furman, Melina. Enseñar distinto.

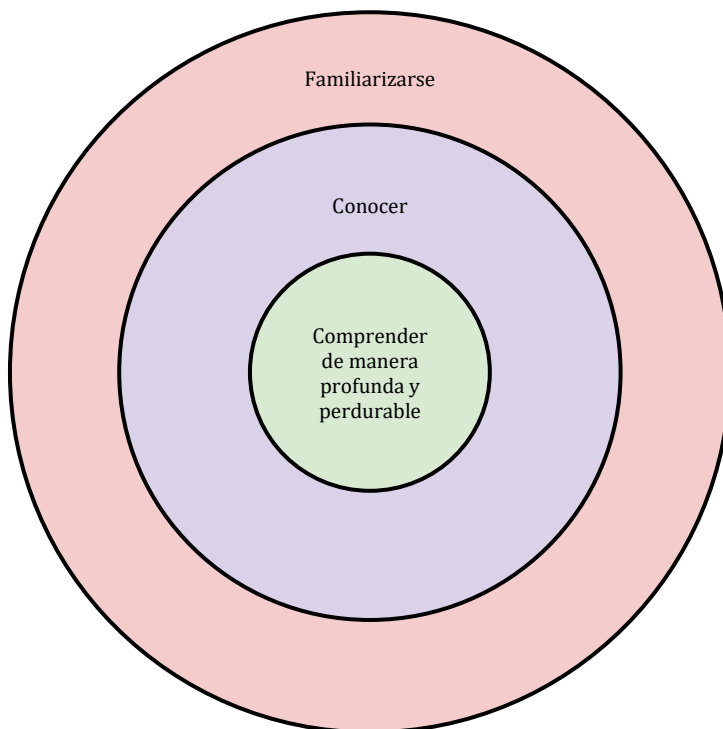
## Actividad 1

El proyecto que comenzaremos a diseñar en esta clase tiene como propósito integrar algunos de los contenidos trabajados hasta el momento. En este sentido, el primer paso será identificar los contenidos que queremos trabajar en el proyecto y organizarlos según el nivel de profundidad con que los vamos a abordar, diferenciando los contenidos que queremos que los/as estudiantes trabajen en profundidad de aquellos que también son relevantes, pero nos alcanza con que los conozcan o estén familiarizados.

Para alcanzar este objetivo, les presentamos un instrumento propuesto por los educadores Wiggins y McTighe para organizar los contenidos curriculares priorizando aquellos que creemos esenciales (e “irrenunciables”) para abordar en el proyecto.

### Los círculos de la comprensión

Los **círculos de la comprensión** son un instrumento propuesto por los educadores Wiggins y McTighe (2005) para organizar y priorizar contenido curricular. Los autores proponen tres círculos que organizan nuestras intenciones con los contenidos:



- En el círculo “Comprender de manera profunda y perdurable” identificaremos los contenidos (tanto conceptos como capacidades) **esenciales**, es decir, los que queremos que toda/os la/os estudiantes aprendan de manera profunda y perdurable. A ellos tendremos que dedicarles tiempo suficiente en el proyecto, por ejemplo, ofreciendo desafíos que permitan abordarlos en concreto y generando momentos que habiliten el análisis, la evaluación y la reflexión.
- En el círculo “Conocer” ubicaremos los contenidos que esperamos que la/os estudiantes **conozcan**, aunque no necesariamente con tanta profundidad, porque no son tan fundamentales como los primeros pero que están fuertemente relacionados con otros contenidos. Los contenidos de este círculo pueden ser aquellos que las/os estudiantes ya conozcan de antemano o que queremos que conozcan al realizar el proyecto.
- En el tercero, pondremos aquellos contenidos que son importantes para llevar a cabo el proyecto, pero con los cuales queremos que la/os estudiantes simplemente se **familiaricen** (o estén familiarizados). Aquí nos alcanza con que esos contenidos les suenen, o que sepan dónde ir a buscarlos si alguna vez necesitan profundizarlos. Serán contenidos a los que no vamos a dedicarles mucho tiempo, espacios específicos o que incluso podrán quedar para que la/os estudiantes consulten por fuera del tiempo planificado del proyecto.

Lo desafiante de este instrumento es que no hay una única manera de completar los círculos ya que depende fuertemente de nuestras intenciones curriculares. Para facilitar esta tarea y determinar qué contenidos podemos ubicar en cada círculo les sugerimos dos criterios (Furman, 2021):

- **Importancia disciplinar:** en el círculo del centro ubicaremos “las grandes ideas”, es decir, bloques conceptuales fundamentales que nos permiten dar sentido a un gran rango de fenómenos y situaciones y seguir aprendiendo nuevos contenidos. [...] Se trata de aprendizajes que, si no se logran en profundidad, no nos permiten seguir avanzando con bases firmes.
- **Relevancia:** ubicaremos en el centro también aquellos contenidos que identificamos como más pertinentes para nuestras/os estudiantes y que tienen chances de importar en su vida presente y futura. Este criterio dependerá de nuestro grupo de estudiantes en particular, incluyendo su edad y contexto.

Definir estos círculos es una tarea que requiere de tiempo, análisis y reflexión. Por lo tanto, durante el diseño del proyecto volveremos a los círculos para hacer los ajustes necesarios. Es importante que



en esta etapa de diseño y en las siguientes puedan ir registrando todos los cambios y las decisiones por las cuáles han realizado los ajustes.



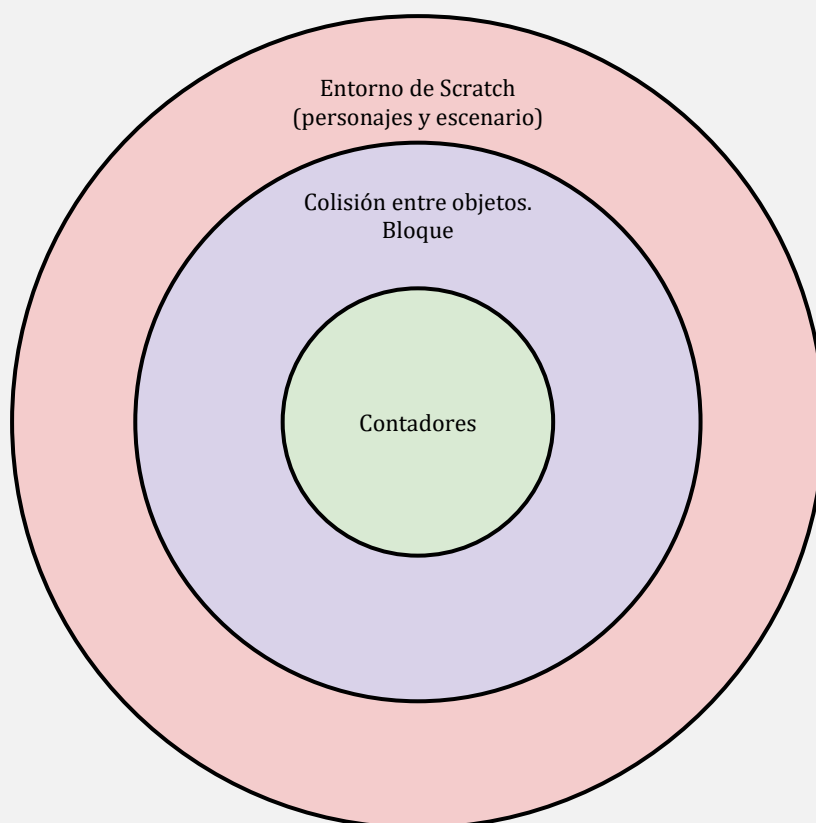
En los módulos anteriores organizamos los contenidos en un **mapa conceptual** que nos daba una visión general de los contenidos que se venían trabajando. En ese mapa identificamos la estrategia didáctica, las herramientas conceptuales y las herramientas del lenguaje. En este caso, los círculos de la comprensión sugieren otra organización donde el foco está puesto en la profundidad con la que vamos a abordar los contenidos.



#### En la práctica...

A continuación, les mostramos un ejemplo sobre el proyecto de la clase 2 donde programaron un videojuego en el que dos personajes debían capturar un objetivo para ganar puntos.

Como primer paso, seleccionamos los siguientes contenidos: *contadores, bloque cambiar/sumar, colisión entre objetos y entorno de Scratch (personajes y escenario)*. Luego, podemos organizarlos de la siguiente manera:



En este ejemplo ubicamos en el centro el uso de variables como **contadores** porque es el contenido principal sobre el cuál queremos que las/os estudiantes profundicen durante el proyecto.

En el segundo círculo, el de “Conocer”, ubicamos el **bloque Sumar/Cambiar** que tenemos a disposición en el entorno para modificar el valor de una variable y la estrategia de colisión entre objetos para poder resolver la dinámica principal del juego (“hacer algo cuando capturamos un objeto”). Una justificación por la cual decidimos ubicar estos contenidos en el segundo círculo es:

- El bloque Sumar es una manera que tenemos en Scratch de manipular una variable para utilizarla como contador. En otros lenguajes de programación que nos permitan usar variables, los bloques y/o comandos pueden llamarse diferente y utilizarse de otra manera. Sin embargo, el contenido principal son los contadores y nos parece importante que conozcan la manera que tenemos de implementarlos en Scratch.

- Si bien la colisión de objetos no es el contenido central del proyecto es importante para la dinámica del videojuego y una estrategia que seguramente habilite a las/os estudiantes a programar otros proyectos en Scratch (y otros entornos).

Finalmente, en el tercer círculo decidimos ubicar aquellos contenidos relacionados al entorno que son importantes para llevar a cabo el proyecto pero que resultan más “operativos” o específicos del entorno. En este sentido, en el proyecto de la clase no dedicamos demasiado tiempo a estos contenidos, pero sí los tratamos en diferentes momentos específicos a lo largo del módulo.

Como podemos observar en el ejemplo anterior, los contenidos esenciales son los cimientos, los más importantes que queremos que los y las estudiantes se lleven. Si no logramos que comprendan el concepto de contador difícilmente puedan conocer diferentes usos de las variables, identificar situaciones en las que deban resolver un problema donde deban llevar la cuenta de algún valor (por ejemplo, un puntaje) y comprender el propósito del bloque Sumar/Cambiar.

Parece evidente cuando lo pensamos de este modo, pero en la práctica a menudo dedicamos mucho tiempo y esfuerzo a que los alumnos recuerden datos y términos que se ubican en el tercer círculo de prioridades. Como consecuencia, pasamos rápido los conceptos esenciales que requieren que ofrezcamos múltiples actividades y puertas de entrada para su real comprensión.



Cuando trabajamos con un entorno de programación abierto, muchas veces invertimos demasiado tiempo y esfuerzo en presentar sus características y funcionalidades, poniendo en segundo plano los conceptos principales que queremos abordar. Por ejemplo, cuando presentamos Scratch nos preocupamos por minimizar o aislar estos momentos, como crear un nuevo proyecto, compartirlo, agregar nuevos personajes o modificar el escenario, para que se acople lo menos posible con el contenido principal.

### Actividad TF1



Teniendo en cuenta los contenidos trabajados en los tres módulos, seleccionen los contenidos que les gustaría trabajar en su proyecto y utilicen los círculos de la comprensión para organizarlos. Para esto les sugerimos que piensen, ¿qué contenidos ubicarían en cada círculo?, ¿por qué?, ¿en qué orden completarían cada círculo?

Les recordamos algunos de los contenidos de programación trabajados en los módulos que pueden considerar para su proyecto: *comandos básicos, procedimientos, repetición simple, alternativa condicional, eventos y variables*. También pueden considerar otros contenidos que hayan trabajado y no estén en la lista.

Para completar esta actividad hagan una copia de la [plantilla de diseño de proyecto](#) donde van a encontrar cada una de las consignas de trabajo. Este documento lo utilizaremos además para resolver las actividades del módulo y será parte fundamental de la entrega del trabajo final.

### Actividad con intervención en el foro



Compartan en el foro los círculos de la comprensión que armaron para que el/la tutor/a pueda darles una devolución y además observen el trabajo de otros/as compañeros/as. Para que puedan ver las respuestas de sus compañeros primero tienen que intervenir en el foro.

## Actividad 2

En la actividad anterior organizaron y priorizaron los contenidos que van a trabajar a lo largo del proyecto. En esta actividad nos ocuparemos de establecer los **objetivos de aprendizaje** del proyecto, es decir, **¿qué queremos que los y las estudiantes aprendan al realizar el proyecto?** Para conseguir esto, vamos a presentarles un instrumento que los ayudará a definir y organizar los objetivos en función de algunas habilidades generales que pueden poner en práctica los/as estudiantes.

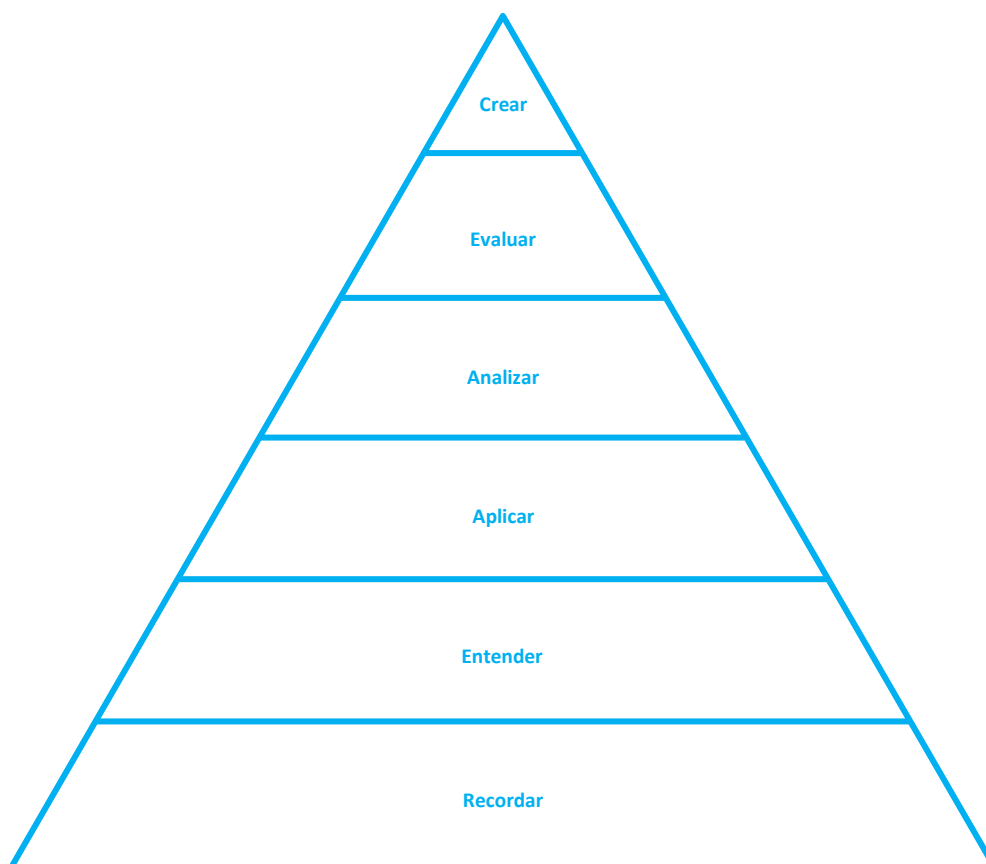


En los módulos anteriores diseñaron una actividad y un proyecto de programación en los que tuvieron que pensar los objetivos de aprendizaje. Para esta actividad pueden recuperar estas definiciones y tomarlas como referencia para analizar las propuestas que les presentamos durante la clase.

## Taxonomía de Bloom

La **Taxonomía de Bloom** ofrece un marco para pensar y clasificar los objetivos de aprendizaje. Fue propuesta en 1956 por el psicólogo educacional Benjamin Bloom junto con un grupo de colegas y establece una clasificación jerárquica de objetivos basada en diferentes habilidades de orden inferior y superior.

Esta taxonomía sigue vigente hoy día y ha sido revisada en varias ocasiones durante los últimos años, siendo una de las más aceptadas la propuesta de Anderson y Krathwohl (2001):



Cada categoría de la pirámide tiene como propósito:

- **Recordar:** recuperar y reproducir hechos y conceptos básicos, definir, mencionar.
- **Entender:** explicar ideas y conceptos, escribirlos con sus palabras.
- **Aplicar:** usar el conocimiento en situaciones nuevas, resolver problemas o casos.
- **Analizar:** interpretar datos u observaciones, sacar conclusiones, establecer relaciones, representar con analogías o metáforas, organizar la información, comparar.
- **Evaluar:** evaluar un asunto usando uno o más criterios o desde distintos puntos de vista, establecer parámetros, elaborar argumentos.
- **Crear:** producir algo original, diseñar, planificar, elaborar.

Es importante notar que el aprendizaje en los niveles superiores depende de haber alcanzado los conocimientos y habilidades necesarias en los niveles inferiores. Sin embargo, la pirámide es una sugerencia y no es necesario que abarquemos todas las categorías. En ese caso, es un buen momento para analizar si podemos precisar algún objetivo que ponga en juego ciertas habilidades.

A continuación, les mostramos una lista de verbos que sirven como guía para cada una de las categorías:

RECORDAR	ENTENDER	APLICAR	ANALIZAR	EVALUAR	CREAR
Definir, describir, identificar, listar, recuperar, reconocer.	Clasificar, conversar, comparar, describir, diferenciar, explicar,  resumir.	Adaptar, cambiar, clasificar, construir, ejercitar, resolver.	Caracterizar, clasificar, comparar, contrastar, explicar, identificar.	Concluir, criticar, justificar, recomendar, resumir, verificar.	Crear, construir, diseñar, mejorar, generar, modificar, preparar, programar, reorganizar, reescribir.



### En la práctica...

Retomando el proyecto de la clase 2 y los círculos de la comprensión que definimos en el ejemplo de la actividad anterior, podemos definir los siguientes objetivos:

- (Recordar) Identificar datos que se almacenan durante la ejecución de un programa.
- (Entender) Diferenciar datos que tienen como propósito llevar la cuenta de algún tipo de información (contadores).
- (Entender) Inferir que los contadores de un programa son variables porque almacenan información y van incrementando/decrementando su valor.
- (Aplicar) Implementar un contador utilizando variables.
- (Evaluar) Concluir que un contador se *inicializa* e *incrementa/decrementa* su valor durante la ejecución del programa.
- (Evaluar) Comparar el propósito de las operaciones de variables *sumar* y *dar*.
- (Recordar) Identificar el problema de colisión entre objetos.
- (Aplicar) Programar una estrategia para detectar colisión entre objetos utilizando repetición y alternativa condicional.

Para mayor claridad y organización listamos los objetivos jerárquicamente y por cada uno mencionamos su categoría en la pirámide. De esta manera podemos tener dimensión del tipo de objetivos que estamos considerando. Como ya mencionamos, la taxonomía es solo un marco sugerido para pensar nuestros objetivos y no es necesario cubrir todas las categorías.



A continuación, les listamos algunos ejemplos de objetivos del proyecto de la clase 2 que pueden ser ambiguos o poco claros:

- *Aprender contadores*
  - ¿Qué queremos aprender de este concepto? ¿Cómo podemos saber que lo aprendieron? ¿Es observable? ¿Y medible?
- *Ejercitar el uso de variables*
  - ¿Qué significa “ejercitar”? ¿Cómo lo vamos a lograr? ¿Es observable este objetivo?
- *Aprender el bloque “por siempre”*
  - ¿Qué queremos que aprendan? ¿Para qué lo van a utilizar? ¿Cómo hacemos que sea observable?



En los objetivos no estamos explicitando todos los contenidos del círculo, en particular, los del círculo de familiarización. Es importante que los contenidos a “comprender en profundidad” estén considerados en los objetivos ya que son contenidos principales y a los que tendremos que dedicarles más tiempo durante el proyecto.



### Actividad TF2

En función de los contenidos que hayan definido en la actividad anterior escriban en su plantilla de diseño los **objetivos de aprendizaje** de su proyecto. Les sugerimos que:

- Los objetivos sean claros, precisos y alcanzables.
- El cumplimiento de cada objetivo sea medible y observable.
- Identifiquen el círculo de la comprensión que está involucrado.



- Identifiquen la categoría de la taxonomía.

Una vez que los hayan definido, les proponemos que compartan un enlace a su diseño de proyecto en un documento compartido que pondrá a disposición su tutor/a con el propósito de formar parejas y retroalimentar sus objetivos. Para esto cada uno/a de ustedes deberá dar una retroalimentación a su par docente teniendo en cuenta que:

- **Valoren** aspectos positivos de los objetivos, es decir, resalten aspectos que consideran que están bien logrados o resultan interesantes.
- **Sugieran** ideas lo más precisas posibles de cómo podría haberse hecho distinto.

Una vez que hayan recibido la retroalimentación, deberán volver al diseño de su proyecto y ajustar sus objetivos teniendo en cuenta las sugerencias que recibieron. En la plantilla de diseño verán una sección de “retroalimentación” donde deberán registrar una breve justificación de los aciertos y cambios que realizaron en los círculos.

## Actividad 3

Una vez establecidos los objetivos de aprendizaje les proponemos definir **evidencias de aprendizaje** que den cuenta de si las/os estudiantes alcanzaron los objetivos que buscamos con este proyecto, es decir, buscamos recoger información relevante sobre el estado de cumplimiento de los objetivos: ¿Qué debería poder demostrar un/a estudiante que alcanzó los aprendizajes que buscábamos? ¿Qué demostraría una/o que aún no los alcanzó, o que los alcanzó parcialmente? ¿Vamos por el buen camino? ¿Cómo ajustamos el rumbo? ¿Qué devolución les hacemos a las/os estudiantes?

Cómo mencionan Anijovich y Cappelletti (2017), podemos **definir las evidencias** como *“aspectos observables que remiten a los procesos o elementos más complejos que son objeto de evaluación”*, en este caso, remiten a aspectos observables que podemos recolectar durante las actividades que vamos a llevar a cabo durante el proyecto y nos permiten medir el cumplimiento de los objetivos establecidos. Las evidencias pueden ser tanto el resultado que las/os estudiantes demuestran tras el desarrollo de una propuesta didáctica (en nuestro caso puede ser una actividad, una secuencia de actividades o el proyecto completo), como los testimonios y registros que el/la docente tenga la posibilidad de obtener de los desempeños o actuaciones durante distintos momentos del proceso de desarrollo del proyecto.



Las evidencias de aprendizajes debieran permitir, entre otras cosas:

- Indagar **qué han aprendido** las/os estudiantes y **qué han enseñado** las/os docentes en una determinada experiencia y también **qué dicen ellos/as** que han aprendido/enseñado.
- Visualizar **cómo el/la docente planificó, desarrolló y evaluó** los procesos de enseñanza y de aprendizaje.



Una evidencia de aprendizaje es efectiva si nos permite **indagar, conocer y comprender acerca de los aprendizajes que los y las estudiantes están construyendo.**



### En la práctica...

Continuemos con el ejemplo sobre el proyecto de la Clase 2 y definamos algunas evidencias de aprendizaje que podemos recoger para indagar sobre el nivel de cumplimiento de nuestros objetivos. Además, observen si estas evidencias cumplen con las sugerencias que les propusimos previamente:

Objetivo	Instrumentos de recolección	Evidencia/as
(Recordar) Identificar datos que se almacenan	Lista con información/datos almacenados durante el	Los datos recolectados refieren a información que es almacenada y utilizada por un programa, por

durante la ejecución de un programa.	uso de una aplicación/videojuego.  Lista con otros ejemplos de datos almacenados por un programa.	ejemplo, un puntaje, la cantidad de vidas disponibles, el porcentaje de batería o la hora.
(Entender) Diferenciar datos que tienen como propósito llevar la cuenta de algún tipo de información (contadores).	Clasificación de los datos por algún criterio en un pizarrón/hoja/afiche.	La clasificación da cuenta de que hay valores que pueden aumentar o disminuir durante la ejecución de un programa (por ejemplo, los puntos, los segundos transcurridos o el porcentaje de batería) y otros que no (el nombre del usuario, el día de hoy)
(Entender) Inferir que los contadores de un programa son variables porque almacenan información y van incrementando/decrementando su valor.	Explicación verbal/escrita del comportamiento de esos datos.	La explicación pone en evidencia que hay datos almacenados por un programa que cambian en función de su valor actual.
(Aplicar) Implementar un contador utilizando variables.	Programa en Scratch.	En el programa se observa el uso de variables y el bloque Sumar para contar los puntos.

( <i>Evaluar</i> ) Concluir que un contador se <i>inicializa</i> e <i>incrementa/decrementa</i> su valor durante la ejecución del programa.	Explicación verbal/escrita de las características que tiene un contador.	La explicación muestra la importancia de la inicialización como acción necesaria para poder incrementar o disminuir el valor de una variable.
( <i>Evaluar</i> ) Comparar el propósito de las operaciones de variables <i>Sumar</i> y <i>Dar</i> .	Cuadro comparativo entre las dos operaciones.	En el cuadro se muestra que el bloque <i>Sumar</i> actualiza el valor de una variable a partir de su valor actual y el bloque <i>Dar</i> reemplaza el valor actual.
( <i>Recordar</i> ) Identificar el problema de colisión entre objetos.	Estrategia de solución del problema escrita en papel/pizarrón.	La estrategia da cuenta de una alternativa condicional para detectar que dos objetos se están chocando y que es necesario realizar este comportamiento durante todo el juego.
( <i>Aplicar</i> ) Programar una estrategia para detectar colisión entre objetos utilizando repetición y alternativa condicional.	Programa en Scratch.	En el programa se observa que utiliza un bloque de alternativa condicional para detectar si toca un objeto dentro de un bloque “por siempre”.



### Actividad TF3

Seleccionen tres objetivos de aprendizaje de su proyecto y para cada uno definan **evidencias de aprendizaje** y los instrumentos de recolección. Como guía les proponemos que piensen: Las evidencias, *¿nos ofrecen la posibilidad de indagar, conocer y comprender...*

- **qué** aprendieron?
- **cómo** lo hicieron?
- las **producciones** que lograron (sus características, los aspectos positivos y lo que debe ser revisado y reformulado/reelaborado)?
- la **utilización** que hayan podido hacer de lo aprendido en otras situaciones (la transferencia)?
- los modos en que **enfrentan y resuelven** situaciones?
- lo que **sienten y testimonian** acerca del nivel de satisfacción/insatisfacción con la experiencia de aprendizaje, con los logros alcanzados, con las dificultades?
- los **cambios (mejoras)** que ponen de manifiesto en relación con la apropiación de aprendizajes, tanto aquellos que esperábamos como otros que no estaban previstos?

Luego de que definan las evidencias, les proponemos una dinámica de retroalimentación similar a la de la actividad anterior: compartan un enlace a su diseño de proyecto en un documento compartido que pondrá a disposición su tutor/a para formar parejas y realizar una retroalimentación de las evidencias definidas. Para esto les sugerimos que al dar su retroalimentación:

- **Valoren** aspectos positivos de las evidencias, es decir, resaltar aspectos que consideramos que están bien logrados o resultan interesantes.
- **Sugieran** ideas lo más precisas posibles de cómo podría haberse hecho distinto.

Una vez que hayan recibido la retroalimentación, regresen al diseño de su proyecto y ajusten sus evidencias teniendo en cuenta las sugerencias que recibieron. En la sección de “retroalimentación” del diseño registren las evidencias que tuvieron que cambiar y por qué.



### Actividad individual con entrega obligatoria

Deberán entregar una versión del diseño del proyecto que incluya la organización de los contenidos, los objetivos de aprendizaje y las evidencias.

**IMPORTANTE:** para poder realizar la entrega de esta actividad **no deben trabajar en línea sobre este documento**. Les pedimos que **descarguen** [esta plantilla](#) en formato **Word** o similar (“*Archivo*”/“*descargar*”) para luego poder **adjuntarla en el buzón de entrega** (“Trabajo final”). Tengan en cuenta renombrar el documento para poder identificarlos. Ejemplo: “Juan Preciado - Aula xx”

## Clase de consultas



### Consultas sincrónicas

**Fecha del encuentro:** **DD/MM/AAAA**

Durante la clase de consultas repasaremos cada una de las actividades y conversaremos sobre aciertos y dificultades que se hayan encontrado. A partir de esto, aunaremos criterios para la organización de los contenidos, la definición de los objetivos y la selección de evidencias.

## Reflexión de cierre



El recorrido de esta clase debiera permitirles:

- A. Organizar el contenido a trabajar en un proyecto en función de la profundidad con la que esperan abordar cada uno.
- B. Pensar y escribir los objetivos de aprendizaje en función de la importancia en la comprensión de los contenidos.
- C. Clarificar las evidencias de aprendizaje que vamos a recoger y observar para medir el nivel de alcance de los objetivos.

## Bibliografía de referencia

Balch, C.; Chung, M. y Brennan, K. (2014). [Computación creativa](#). Escuela de Graduados en Educación de la Universidad de Harvard.

Furman, M. (2021). Enseñar Distinto. Guía para innovar sin perderse por el camino. Buenos Aires, Argentina: Siglo XXI Editores.

Anijovich, R. y Cappelletti, G. (2018). La evaluación como oportunidad. Buenos Aires, Argentina: Paidós.

Anijovich, R. y González, C. (2011). Evaluar para aprender: conceptos e instrumentos. Buenos Aires, Argentina: Aique.

Gobierno de Córdoba. Ministerio de Educación. Secretaría de Educación. Subsecretaría de Promoción de Igualdad y Calidad Educativa (2018). [Procesos de evaluación en la formación situada; Selección, análisis e interpretación de evidencias de aprendizajes](#). Córdoba, Argentina: Gobierno de Córdoba. Ministerio de Educación. Secretaría de Educación. Subsecretaría de Promoción de Igualdad y Calidad Educativa.

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto:

Fundación Sadosky (2023). Clase 3: Proyecto integrador de programación (Parte 1). Módulo: Estrategias para la enseñanza de la programación III. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.



Esta obra está bajo una licencia Creative Commons  
Atribución-NoComercial-CompartirIgual 3.0



## Módulo 5: Estrategias para la enseñanza de la programación III

# Clase 4: Proyecto integrador de programación (Parte 2)

## Introducción

En la clase anterior comenzamos a definir los primeros elementos del proyecto: contenidos y objetivos de aprendizaje. Tener previamente definidos los objetivos nos dan un horizonte claro hacia dónde queremos llegar con el proyecto. El paso siguiente, que abordaremos en esta clase, es poder obtener pistas claras sobre el nivel de alcance de los objetivos. Para esto diseñaremos listas de cotejo que nos brinden información relevante acerca del nivel de logro de los objetivos. A partir de estas definiciones, diseñaremos una secuencia de actividades que entre en diálogo con las listas de cotejo y permita alcanzar los objetivos propuestos. Como experimentaron en los módulos anteriores, el proyecto deberá estar dividido en pequeñas actividades que nos aproximen cada vez más a una resolución completa.

## Actividad 1

Antes de definir una secuencia de actividades para nuestro proyecto y tomando como referencia el diseño en retrospectiva (o “hacia atrás”), les proponemos pensar y definir los criterios de valoración asociados a los objetivos de aprendizaje. Para organizar los criterios les proponemos utilizar las **listas de cotejo** para identificar la presencia o ausencia de un objetivo de aprendizaje y compartir con los estudiantes qué vamos a valorar en sus producciones y transparentar nuestros criterios de evaluación del proyecto.

Esta actividad consta de dos partes: en la primera les presentamos ejemplos de listas de cotejo para el proyecto de la clase 2 que luego analizaremos juntos/as en el encuentro sincrónico y deberán aplicarlas al proyecto de un/a colega, mientras que en la segunda actividad deberán crear sus propias listas de cotejo sobre los objetivos de aprendizaje de su proyecto final.



Una **lista de cotejo** consiste en una tabla de doble entrada donde se registran los criterios o indicadores que vamos a considerar para evaluar el proyecto y su nivel de *presencia o ausencia* en una actividad. Es fundamental que podamos brindar esta lista de antemano para que los/as estudiantes tengan una referencia clara a la hora de resolver las actividades que les propongamos y que puedan autoevaluarse antes de entregarlos sobre una base concreta.

Además, resultan muy útiles para que los/as estudiantes puedan autoevaluar sus trabajos o evaluar los de sus compañeros/as. Podemos proponerles que usen este instrumento para valorar sus producciones antes de entregarlas, y en caso de que vean que en alguna dimensión no llegaron al nivel de logro más alto, traten de mejorarla. También podemos usarlas para que evalúen el trabajo de sus compañeros/as y les ofrezcan sugerencias de cómo mejorarlos.



Una sugerencia para definir los criterios que vamos a considerar es pensar en **evidencias de aprendizaje** que den cuenta si las/os estudiantes alcanzaron los objetivos que buscamos con este proyecto. Para poder decidir el nivel de alcance de los criterios es importante que esta elección se pueda deducir de las evidencias que hayan definido. Utilizar las evidencias como justificación del criterio les permite saber qué tan lejos o qué tan cerca están de alcanzar un objetivo.

## Parte 1

### En la práctica...



Continuemos con el ejemplo de la clase 2 y armemos una lista de cotejo para evaluar variables y contadores. Recordemos que uno de los objetivos del proyecto es “Implementar un contador utilizando variables”. Para ver si el objetivo fue alcanzado, vamos a observar si en la implementación del proyecto en Scratch está presente el uso de variables y el bloque Sumar para contar los puntos.

A partir de estas evidencias, podemos definir los siguientes criterios: “utiliza variables para contar la cantidad de objetos capturados”, “diferencia el efecto que producen los bloques Sumar y Dar en una variable” e “inicializa una variable que es utilizada como contador”. Los niveles los podemos definir de diferentes maneras en función de lo que estamos buscando evaluar, pero recordando que buscamos medir el nivel de presencia o ausencia de los criterios establecidos. En este ejemplo, como el objetivo es específico, nos alcanza como docentes saber si el criterio está presente o no, o, dicho de otra forma, si fue *alcanzado* o aún está *pendiente*. Además, podemos agregar una columna de *observaciones* donde se agreguen precisiones de lo que fuimos observando para pensar posteriormente en acciones para alcanzar o mejorar los desempeños. Teniendo en cuenta todo esto, una lista de cotejo posible sería:

OBJETIVO			
Implementar un contador utilizando variables			
CRITERIO	JUSTIFICACIÓN (Evidencias)	ALCANZADO	PENDIENTE

Utiliza variables para representar contadores y almacenar la cantidad de objetos capturados por jugador.			
El contador de un jugador se incrementa en uno cada vez que captura al objetivo.			
Los contadores se inicializan en cero cada vez que se ejecuta el videojuego.			

Si queremos evaluar el objetivo “Programar una estrategia para detectar colisión entre objetos”, donde queremos observar si en la implementación del proyecto utiliza un bloque de alternativa condicional para detectar si toca un objeto dentro de un bloque de repetición, podemos definir la siguiente lista de cotejo:

OBJETIVO			
Programar una estrategia para detectar colisión entre objetos			
CRITERIO	JUSTIFICACIÓN (Evidencias)	ALCANZADO	PENDIENTE
El objetivo puede detectar si algún personaje lo capturó.			
La detección de si está tocando a otro objeto se chequea durante toda la ejecución del programa.			
El objetivo cambia de posición cada vez que es capturado por algún personaje durante la ejecución.			

## Parte 2



### ACTIVIDAD TF4

Seleccionen al menos tres objetivos de aprendizaje de su proyecto y agreguen a su diseño de proyecto una **lista de cotejo** para evaluarlos. Una vez que la hayan construido vuelvan sobre la planilla donde vienen realizando las retroalimentaciones para dejar sus comentarios sobre la lista de cotejo de otro/a colega. Para esto les sugerimos que:

- **Valoren** aspectos positivos de la lista, es decir, resalten aspectos que consideran que están bien logrados o les resulten interesantes.
- **Sugieran** ideas lo más precisas posibles para mejorar aquellos criterios que crean que no evalúan los objetivos o no brindan información relevante.

Una vez que hayan recibido la retroalimentación, deberán volver al diseño de su proyecto y ajustar la lista de cotejo teniendo en cuenta las sugerencias que recibieron, registrando en la sección de “retroalimentación” una breve justificación de los aciertos y cambios que realizaron en los círculos.

Tengan en cuenta que la lista deberá ser fácil de interpretar y aplicar por potenciales estudiantes para que, además de transparentar los criterios de evaluación de ustedes como docentes, ellos/as mismos/as puedan autoevaluar su proyecto.

**Aclaración:** esta actividad no se entrega por medio de la plataforma. La lista de cotejo se agrega en el documento en línea donde cada uno/a está elaborando su proyecto final. De esa manera sus colegas pueden verlas para realizar la retroalimentación y su tutor/a puede dejar sugerencias/comentarios.



Tengan en cuenta que los criterios de la lista se aplican a los objetivos de aprendizaje y no al proyecto en sí. Esto quiere decir que es importante descartar de la lista aquellos aspectos que tienen que ver con la dinámica o la estética del proyecto que no están considerados en los objetivos y no tienen que ver con los contenidos que estamos evaluando.

## Encuentro sincrónico



### Encuentro sincrónico

Fecha del encuentro: **DD/MM/AAAA**

En este encuentro sincrónico presentaremos algunos de los proyectos que hayan realizado con el objetivo de recuperar experiencias del proceso de diseño, centrándonos en cómo se relacionan los desafíos con las definiciones de clase 3. Además, presentaremos y conversaremos sobre las listas de cotejo como un instrumento de evaluación de los aprendizajes involucrados en los proyectos.

## Actividad 2



### ACTIVIDAD TF5

Con base en los elementos definidos en la clase anterior, diseñen e implementen una primera versión de un proyecto de programación organizado en una secuencia de desafíos. Para cada uno de estos deberán:

- Especificar los objetivos que involucra.
- Escribir la consigna. Si es necesario, pueden agregar pistas.
- Describir, cuando corresponda, los momentos expositivos, de conceptualización, de indagación, de práctica individual/grupal, de reflexión o de evaluación/retroalimentación.
- Implementar el proyecto en Scratch.

Para esta actividad les sugerimos:

- Pensar en desafíos con complejidad incremental.
- Que cada desafío:
  - Tenga un objetivo específico.
  - Ponga en juego los criterios de valoración definidos en las listas de cotejo.
  - Involucre la menor cantidad de contenidos posibles. Es preferible que no haya más de dos conceptos del círculo central para evitar que se acoplen objetivos.
- En los módulos anteriores hablamos de la importancia de que un proyecto tenga un **piso bajo, paredes anchas y techo alto**. ¿Cómo justificarían brevemente la presencia de estas características en su proyecto?
- El recorrido debe tener **cohesión**. Para eso, tengan en cuenta cómo se pueden aprovechar los desafíos resueltos previamente y cómo aporta un nuevo desafío a los desafíos siguientes.

Además, les proponemos que revisen y ajusten sus objetivos y listas de cotejo a medida que van diseñando los desafíos. En caso de que hayan hecho modificaciones, registren en su plantilla qué objetivos/criterios tuvieron que ajustar y por qué.



El foco del proyecto debe estar puesto en los contenidos de programación. Si su proyecto involucra contenidos de otras áreas, no es necesario que los aborden específicamente en los desafíos, pero expliciten cuáles son esos contenidos y en qué actividades se recuperan.



### Repositorios de proyectos

Les ofrecemos algunos recursos donde podrán encontrar proyectos de ejemplo en Scratch que les puedan resultar de inspiración para pensar los desafíos de su proyecto:

- Computación Creativa. Disponible en español en <https://eduteka.icesi.edu.co/pdfdir/hgse-scratch-computacion-creativa.pdf>
- Repositorio oficial de Scratch:
  - <https://scratch.mit.edu/explore/projects/all>
  - <https://scratch.mit.edu/explore/studios/all/>



En caso de que decidan utilizar algún contenido que no se trabaje en los módulos registren en su plantilla de proyecto, ¿Cuáles utilizaron? ¿En qué círculo de la comprensión lo ubicarían? ¿Cuántos desafíos les llevaría introducirlo? ¿Tuvieron que agregar/modificar algún objetivo/evidencia?



### En la práctica...

Dentro de los ejemplos sobre el proyecto de la clase anterior mencionamos una lista de objetivos y criterios que definen las metas que deberían alcanzar quienes realicen el proyecto y que nos ayuda a nosotros/as docentes a tener claro a dónde tenemos que llegar y de qué manera ajustar el rumbo a partir de las evidencias que dan cuenta las listas de cotejo.

Les proponemos que vuelvan nuevamente a la clase 2 junto con las listas de cotejo que definimos en el ejemplo y relacionen los criterios con los desafíos propuestos en la clase. Además, ¿creen que los desafíos dan lugar a momentos donde se ponen en juego los criterios?



### Actividad individual con entrega obligatoria

Deberán entregar una versión del diseño del proyecto final (que incluya los contenidos, objetivos de aprendizaje, evidencias de aprendizaje, lista de cotejo, la secuencia de desafíos y el link al proyecto de Scratch).

Formato de entrega: deberán descargar el documento en línea de Drive donde trabajaron en formato DOCX u ODT (archivo/Descargar) y adjuntarlo en el buzón de entrega.

**Fecha de entrega:** 17 de mayo



## Reflexión de cierre



El recorrido de esta clase debiera permitirles:

- A. Pensar y diseñar listas de cotejo que nos brinden información relevante durante las actividades acerca del nivel de cumplimiento de los objetivos.
- B. Pensar y diseñar una secuencia de actividades para un proyecto de programación que se apoye en los criterios de valoración y nos dé insumos para ayudar a los/as estudiantes a alcanzar los objetivos de aprendizaje.
- C. Implementar un proyecto de programación en Scratch desde cero.



## Bibliografía de referencia

Balch, C.; Chung, M. y Brennan, K. (2014). [Computación creativa](#). Escuela de Graduados en Educación de la Universidad de Harvard.

Furman, M. (2021). Enseñar Distinto. Guía para innovar sin perderse por el camino. Buenos Aires, Argentina: Siglo XXI Editores.

Anijovich, R. y Cappelletti, G. (2018). La evaluación como oportunidad. Buenos Aires, Argentina: Paidós.

Anijovich, R. y González, C. (2011). *Evaluar para aprender: conceptos e instrumentos*. Buenos Aires, Argentina: Aique.

Gobierno de Córdoba. Ministerio de Educación. Secretaría de Educación. Subsecretaría de Promoción de Igualdad y Calidad Educativa (2018). [Procesos de evaluación en la formación situada; Selección, análisis e interpretación de evidencias de aprendizajes](#). Córdoba, Argentina: Gobierno de Córdoba. Ministerio de Educación. Secretaría de Educación. Subsecretaría de Promoción de Igualdad y Calidad Educativa.

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto: Fundación Sadosky (2023). Clase Nro. 4: Proyecto integrador de programación. Módulo: Estrategias para la enseñanza de la programación III. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.



Esta obra está bajo una licencia Creative Commons  
[Atribución-NoComercial-CompartirIgual 3.0](#)