

Colección Actualizaciones Académicas

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

Módulo 4: Estrategias para la enseñanza de la  
programación II





## Índice

<b>Clase 1: Presentación. Programas y escenarios cambiantes .....</b>	<b>3</b>
<b>Clase 2: Escenarios aún más cambiantes .....</b>	<b>11</b>
<b>Clase 3: Programas interactivos .....</b>	<b>17</b>
<b>Clase 4: Creamos un videojuego .....</b>	<b>32</b>



## Módulo 4: Estrategias para la enseñanza de la programación II

# Clase 1: Presentación. Programas y escenarios cambiantes

## Bienvenidas y bienvenidos

Sean bienvenidas y bienvenidos al segundo módulo de Estrategias para la enseñanza de la programación. En este módulo abordaremos un modo de enseñar una nueva herramienta conceptual de programación a nuestros estudiantes y cómo acompañarlos en la creación de programas interactivos en Scratch, un entorno de programación orientado a proyectos.

En esta primera clase introduciremos la alternativa condicional, una herramienta conceptual que nos permitirá resolver varios escenarios de un mismo problema con un único programa. Será importante que recuperen y repasen división en subproblemas, procedimientos, repetición simple y legibilidad pues las actividades de este módulo se irán integrando con las herramientas y conceptos trabajados en el módulo anterior.

## Actividad 1

Para comenzar este nuevo módulo, les proponemos resolver tres desafíos en el entorno de Pilas Bloques que introducirán la **alternativa condicional**, una herramienta conceptual para trabajar con problemas con escenarios cambiantes.



En este módulo, trabajaremos con una bitácora individual. Para eso, su tutor/a les compartirá un documento de Google Drive para que vuelquen sus registros. El documento ya tiene un esqueleto con todas las actividades del módulo y algunas preguntas orientadoras.



Resuelvan los desafíos de Pilas Bloques presentados a continuación y registren en sus bitácoras las respuestas a las preguntas de cada desafío. Es recomendable, además guardar una captura de pantalla de las soluciones elaboradas.

#### Desafío en Pilas Bloques: El mono y las bananas

<https://pilasbloques.program.ar/online/#/desafio/13>



- ¿Cómo se motiva la necesidad de introducir la alternativa condicional?
- ¿Qué forma adoptan las condiciones en Pilas Bloques?
- ¿Qué dificultades creen que encontrarán las y los estudiantes en esta actividad? ¿Qué orientaciones darían? ¿Qué ejemplos de la vida cotidiana se les ocurre para favorecer la transferencia de este concepto?

#### Desafío en Pilas Bloques: La elección del mono

<https://pilasbloques.program.ar/online/#/desafio/14>



- ¿Qué otras combinaciones de alternativas condicionales permiten resolver este desafío?
- ¿Qué pasaría con la solución que construyeron si el problema incluyera un escenario en que la manzana y la banana aparecieran ambas al mismo tiempo? ¿Y si el casillero estuviera vacío?

#### Desafío en Pilas Bloques: Laberinto corto

<https://pilasbloques.program.ar/online/#/desafio/15>



- ¿Cuántas formas de resolver el problema consideraron? ¿Existe alguna forma más adecuada que otra? ¿Por qué?

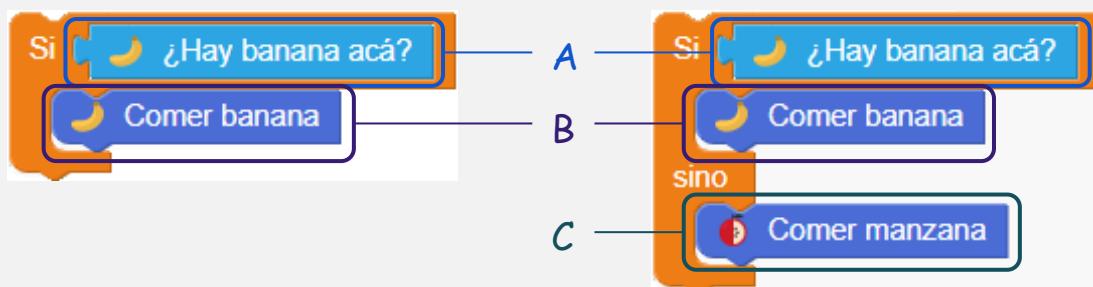


La **alternativa condicional** es un comando que permite que las computadoras ejecuten ciertos comandos sólo en determinadas situaciones. Estas situaciones se pueden identificar mediante una **condición**, es decir, una pregunta con respuesta afirmativa o negativa que permite determinar las diferencias entre las situaciones planteadas en el problema.

Para esto, Pilas Bloques agrega la opción *Sensores*, con todas las condiciones disponibles en cada desafío, y dos bloques especiales.

En ambos bloques, en la parte superior (A) se coloca la condición que hará que los comandos o instrucciones (B) se ejecuten si la respuesta a la condición es SI.

Si la respuesta a la condición es NO, en el caso de la alternativa condicional simple (izquierda) las instrucciones simplemente no se ejecutarán y la ejecución continuará en los comandos que haya debajo (o terminará si no los hay). En el caso de la alternativa condicional completa (derecha), se ejecutarán los comandos o instrucciones que se incluyan en la sección del *sino* (C) y luego continuará la ejecución de los siguientes bloques.



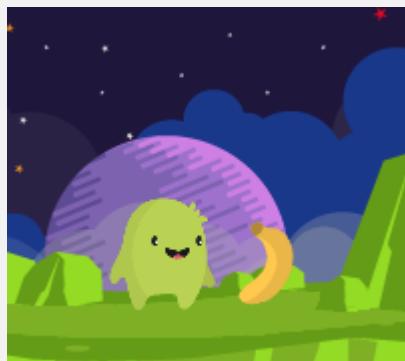
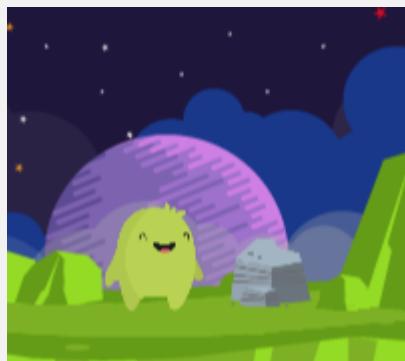


## Actividad 2

Con los nuevos comandos de alternativa condicional incorporados en nuestro repertorio de herramientas de programación, practiquemos un poco más con ellos. En los siguientes desafíos deberán elaborar estrategias de resolución que den cuenta de los subproblemas involucrados en cada desafío y decidir qué comando de alternativa condicional, o combinación de ambos, es la mejor para lograr una resolución efectiva de los desafíos.



Tomando como base las reflexiones y conclusiones de la actividad anterior, resuelvan los siguientes desafíos en el entorno Pilas Bloques. Recuerden ejecutar varias veces para identificar los diferentes escenarios y elaborar estrategias de resolución que contemplen esas variaciones.



[Las rocas de Nano](#)

[Los caminos de Nano](#)



[Una fiesta arruinada](#)

[Redecorando fiestas](#)



Una vez resueltos los cuatro desafíos anteriores respondan en sus bitácoras las siguientes preguntas:

- ¿Qué dificultades agregadas tienen estos desafíos con respecto a los de la actividad anterior (“El mono y las bananas”, “La elección del mono” y “Laberinto corto”) en cuanto a su resolución?
- ¿Se podría haber resuelto cada desafío utilizando un comando de alternativa condicional u otro indistintamente? ¿Por qué?
- ¿Qué indicaciones o ayudas les darían a sus estudiantes que, frente a las dificultades de estos desafíos, no puedan resolverlos?



Cada uno de los comandos de alternativa condicional funciona mejor en ciertas situaciones.

- Utilizamos una alternativa condicional simple cuando nuestras estrategias de resolución deben realizar pasos adicionales bajo cierta condición y retomar su curso normal, como si se tratara de un **desvío**.
- Utilizamos una alternativa condicional completa cuando nuestra estrategia necesita seguir dos cursos de acción diferentes dependiendo de si se cumple o no una condición dada, como si se tratara de una **bifurcación**.

En cualquier caso, resulta indispensable analizar el problema planteado, identificar las variaciones en el escenario y elaborar una buena estrategia de resolución a través de la división en subproblemas.



#### Actividad individual con entrega obligatoria

Entreguen en un documento de texto (Word, Google, etc.) capturas de pantalla o fotos de las soluciones que elaboraron para los desafíos de esta actividad (“Las rocas de Nano”, “Los caminos de Nano”, “Una fiesta arruinada” y “Redecorando fiestas”).

Su tutor/a tendrá en cuenta los siguientes factores en las correcciones:

- Que el programa elaborado solucione todos los escenarios de cada desafío.



- Que se hayan utilizado las herramientas de programación pertinentes de forma correcta.
- Que el programa elaborado sea legible.

## Actividad 3

Según los objetivos de aprendizaje que definamos, a veces será conveniente recurrir a actividades desenchufadas y/o actividades de programación en computadora. En esta línea, les proponemos revisar las siguientes actividades y luego responder la pregunta debajo.



- **Actividad ¿Qué harías si...?** extraída del manual *Ciencias de la Computación para el aula, Primer Ciclo Primaria*.  
<https://drive.google.com/file/d/1uT9fW38K2Avc-eCUF19zcYiEDaJm4iZO/view?usp=sharing>
- **Actividad La carrera de los palos españoles** extraída del manual *Ciencias de la Computación para el aula, Segundo Ciclo Primaria*.  
[https://drive.google.com/file/d/1lj9kPlUtTwfCmXZK6XF3v\\_H9IKBOeDGL/view?usp=sharing](https://drive.google.com/file/d/1lj9kPlUtTwfCmXZK6XF3v_H9IKBOeDGL/view?usp=sharing)



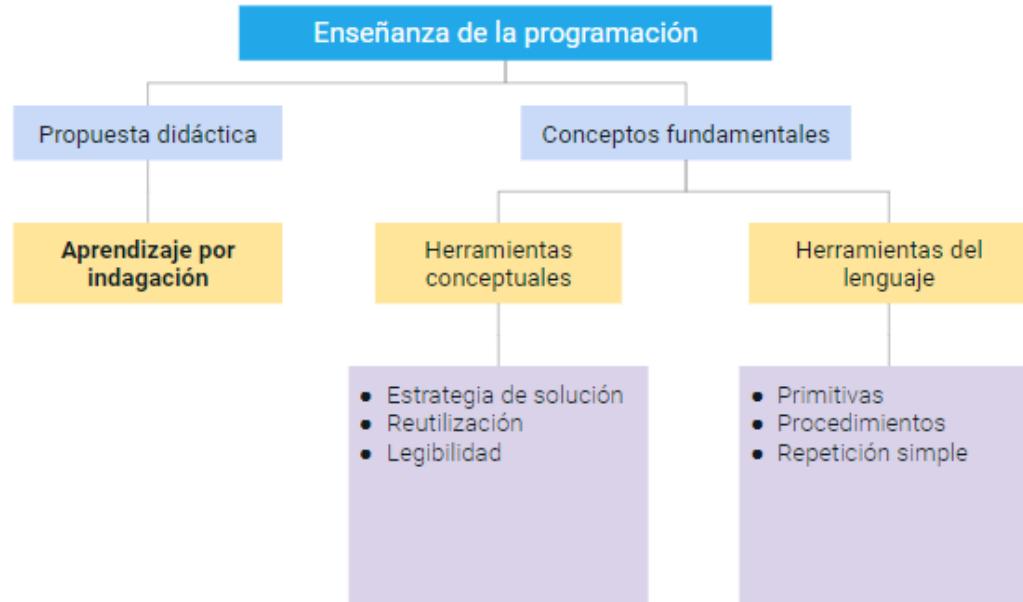
Registren en sus bitácoras las respuestas a la siguiente situación:

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?

Para finalizar, en la bitácora encontrarán el **mapa conceptual** que habían comenzado en el módulo anterior. Les proponemos incorporar las herramientas de programación trabajadas en esta clase.



Deberán agregar a los conceptos fundamentales, las herramientas que abordamos en esta clase: **Alternativa condicional - Condiciones.**



## Clase de consulta



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

Antes de participar en la clase destinada a consultas es importante haber intentado resolver las actividades de la clase. En este encuentro, su tutor/a presentará los desafíos que hayan traído más problemas o que sean más interesantes de revisar para que puedan visualizar un ejemplo de cómo trabajar estas actividades en el aula con el enfoque didáctico propuesto y recordar nuestro rol como docentes, así como los diferentes momentos que les propondremos a los y las estudiantes durante su desarrollo.

Este encuentro sincrónico **es opcional, pero altamente recomendado**. Quienes necesiten resolver algunas dificultades que hayan surgido durante la resolución de los desafíos podrán consultar con su tutor/a y compañeras y compañeros; quienes hayan podido resolverlos, podrán comparar sus soluciones con otras y conocer nuevas ideas.



## Reflexión de cierre



El recorrido de esta clase debiera permitirles:

A. Identificar y comprender los principales conceptos trabajados:

- **Alternativa condicional:** herramienta para condicionar la ejecución de otros comandos disponibles. Las alternativas condicionales, combinadas con las condiciones, son muy importantes para lidiar con problemas sobre escenarios cambiantes.
- **Condiciones:** herramientas para realizar preguntas sobre el estado del problema a resolver que siempre se responden por Sí o por No.

B. Reforzar la importancia de explicitar estrategias de solución por división en subtareas y su rol fundamental en la tarea de resolver problemas mediante programas.

C. Modificar la secuencia de pasos recomendada para solucionar un problema mediante un programa, relacionando los escenarios cambiantes con las alternativas condicionales como una solución a problemas de este tipo.

## Bibliografía de referencia

Factorovich, P. Sawady O'Connor, F. (2017). Actividades para aprender a Program.AR, Capítulo *Alternativas condicionales*. Secuencia didáctica 7, Secuencia didáctica 8 y Ejercitaciones. Buenos Aires: Fundación Sadosky.

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto: Fundación Sadosky (2022). Clase Nro. 1: presentación. Programas y escenarios cambiantes. Módulo: Introducción a la programación II. 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-CompartirlIgual 3.0



## Módulo 4: Estrategias para la enseñanza de la programación II

# Clase 2: Escenarios aún más cambiantes

## Introducción

En esta clase reforzaremos los nuevos comandos de alternativa condicional que presentamos la clase anterior mediante problemas que requieren elaborar estrategias de resolución que integran los comandos de alternativa condicional con todas las herramientas de programación trabajadas hasta el momento.

## Actividad 1

Les proponemos realizar una primera aproximación a algunos desafíos en Pilas Bloques que trabajan con recorridos sobre escenarios cambiantes. Además de intentar responder las preguntas a continuación de los desafíos, utilicen sus bitácoras para registrar cualquier inquietud e inconveniente que les surja durante la resolución de los desafíos para trabajarlas durante la próxima clase de consulta.



Resuelvan los siguientes desafíos en el entorno Pilas Bloques y registren en sus bitácoras las respuestas a las preguntas debajo.



[Tres naranjas](#)



[Tito recargado](#)



[Laberinto largo](#)



- ¿Qué estrategias de resolución plantearon? ¿Qué conclusiones podemos pensar con respecto a las similitudes y diferencias entre las estrategias de resolución de los tres desafíos propuestos?
- ¿Cómo hubieran resultado si cada desafío tuviera solamente un casillero? ¿Se podría generalizar a cualquier cantidad de casilleros?
- ¿Qué dificultades les surgieron al combinar alternativa condicional y repetición? ¿Les parece que a sus estudiantes les surgirán las mismas? ¿Qué orientaciones darían?



Vimos que, cuando detectamos patrones en problemas de programación, contamos con dos herramientas que aportan mucha legibilidad y simpleza a las soluciones elaboradas: procedimientos y repeticiones.

En ambos casos, puede haber pasos en los patrones que pueden suceder a veces sí a veces no, es decir, la ocurrencia o no ocurrencia de un suceso forma parte del patrón. Es ahí cuando la alternativa condicional cumple el rol fundamental de permitir elaborar estrategias que contemplan problemas con escenarios cambiantes.

Al combinar alternativa condicional con repetición se vuelve imprescindible determinar qué pasos de los patrones suceden siempre y qué pasos se ejecutan en determinadas condiciones. Esto se debe a que se dificulta mucho realizar el seguimiento de un programa que no realiza siempre las mismas acciones.

## Actividad 2

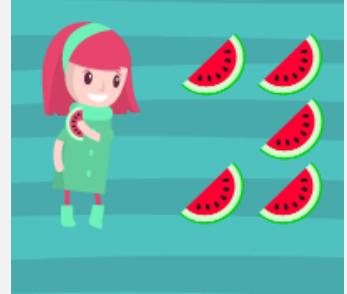
Con una idea general de cómo resolver problemas de recorridos sobre escenarios cambiantes, les proponemos resolver otros dos desafíos de Pilas Bloques que permitirán reforzar y profundizar esta estrategia de resolución. Recuerden registrar dudas e inconvenientes en sus bitácoras para su posterior consulta.



Resuelvan los siguientes desafíos en el entorno Pilas Bloques. A continuación, respondan en sus bitácoras las preguntas debajo.



[El desierto multifrutal](#)



[El pasillo curvo de sandías](#)

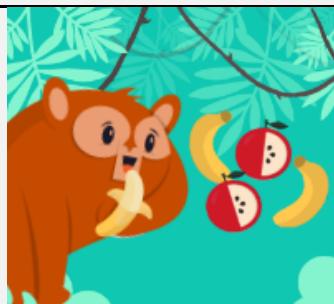
- ¿Qué dificultad agregan estos desafíos con respecto a los anteriores (“Tres naranjas”, “Tito recargado” y “Laberinto largo”)? ¿Qué problemas les surgieron al resolverlos, o consideran que podrían encontrar sus estudiantes? ¿Qué orientaciones les darían?
- ¿Qué reflexiones pueden elaborar en cuanto a la secuenciación de estos desafíos (desde “Tres naranjas” hasta “El pasillo curvo de sandías”)? ¿Cuál es el objetivo de la secuencia?

## Actividad 3

En la siguiente actividad les proponemos resolver un par de desafíos en el entorno Pilas Bloques que terminarán de integrar todas las herramientas conceptuales de programación trabajadas desde el módulo anterior, así como las estrategias de resolución para cada escenario diferente.



Resuelvan los últimos desafíos de Alternativa condicional en el entorno Pilas Bloques.  
Luego respondan en sus bitácoras las preguntas debajo.



[El festín frutal](#)



[Recolector de galaxias](#)

- ¿Cuál es la dificultad agregada de estos desafíos? ¿Qué problemas tuvieron o consideran que podrían encontrar sus estudiantes? ¿Qué orientaciones les darían?
- ¿Cuál es el aporte de estos desafíos a la secuencia completa de actividades de alternativa condicional en el entorno Pilas Bloques? ¿Qué cambios o mejoras consideran que habría que realizar?

#### Actividad individual con entrega obligatoria



Entreguen en un documento de texto (Word, Google, etc.) capturas de pantalla o fotos de las soluciones que elaboraron para los desafíos “El desierto multirfrutal”, “El pasillo curvo de sandías”, “Un festín frutal” y “Recolector de galaxias”.

Su tutor/a tendrá en cuenta los siguientes factores en las correcciones:

- Que el programa elaborado solucione todos los escenarios de cada desafío.
- Que se hayan utilizado las herramientas de programación pertinentes de forma correcta.
- Que el programa elaborado sea legible.



## Clase de consulta



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

Antes de participar en la clase destinada a consultas es importante haber intentado resolver las actividades de la clase. En este encuentro su tutor/a presentará los desafíos que hayan traído más problemas o que sean más interesantes de revisar para que puedan visualizar un ejemplo de cómo trabajar estas actividades en el aula con el enfoque didáctico propuesto y recordar nuestro rol como docentes, así como los diferentes momentos que les propondremos a los y las estudiantes durante su desarrollo.

Este encuentro sincrónico **es opcional, pero altamente recomendado**. Quienes necesiten resolver algunas dificultades que hayan surgido durante la resolución de los desafíos podrán consultar con su tutor/a y compañeras y compañeros; quienes hayan podido resolverlos, podrán comparar sus soluciones con otras y conocer nuevas ideas.

## Reflexión de cierre



El recorrido de esta clase debiera permitirles:

- A. Reforzar los conceptos de alternativa condicional y condición como herramientas conceptuales de programación que permiten que los programas ejecuten ciertos comandos dependiendo de si se cumple o no la condición y, de esta manera, solucionar problemas sobre escenarios cambiantes.
- B. Mejorar la determinación de qué comando de alternativa condicional encaja mejor con cada situación planteada en los problemas presentados: alternativa condicional simple para situaciones donde se deben realizar pasos adicionales, y alternativa condicional completa para situaciones donde se debe bifurcar la ejecución de comandos en base al cumplimiento o no cumplimiento de una condición.



**C.** Fortalecer la identificación de patrones, en particular aquellos que contienen la ocurrencia o no ocurrencia de ciertas situaciones, y la elaboración de estrategias de resolución que se basan en la división en subproblemas.

## Bibliografía de referencia

Factorovich, P. y Sawady O'Connor, F. (2017). Actividades para aprender a Program.AR, Capítulo *Alternativas condicionales*. Ejercitaciones. Buenos Aires: Fundación Sadosky.

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto: Fundación Sadosky (2022). Clase Nro. 2: escenarios aún más cambiantes. Módulo: Introducción a la programación II. 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-CompartirlIgual 3.0



## Módulo 4: Estrategias para la enseñanza de la programación II

### Clase 3: Programas interactivos

#### Introducción

En esta clase comenzaremos a trabajar con Scratch, un entorno para enseñar a programar en donde podrán crear sus propios proyectos. El objetivo no es que las y los estudiantes aprendan Scratch, sino que aprendan programación *usando* Scratch. De esta manera, este entorno no reemplaza a Pilas Bloques, sino que lo complementa, permitiendo la introducción de nuevos conceptos de programación y la creación de programas interactivos.

En este recorrido vamos a resolver pequeños desafíos de programación como estrategia para familiarizarse con el entorno. Estos desafíos, además, tienen como propósito reforzar los conceptos de *comandos básicos, procedimientos, repetición simple y alternativa condicional*. También introducen las herramientas necesarias para la creación de programas interactivos, es decir, programas en los que las personas que los usan pueden intervenir durante su ejecución, por ejemplo, utilizando el teclado o el mouse para mover a un personaje en la pantalla.

Teniendo en cuenta las características y objetivos de Pilas Bloques y Scratch, compararemos ambos entornos e identificaremos límites y posibilidades que ofrecen, qué tipos de actividades podemos desarrollar para nuestro grupo de estudiantes y de qué manera podrían impactar en la dinámica de trabajo en el aula.

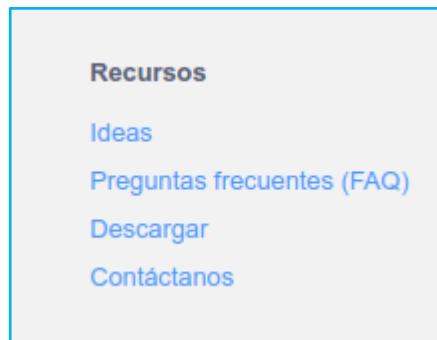
#### Actividad 1

### Primera parte: exploración de la comunidad de Scratch

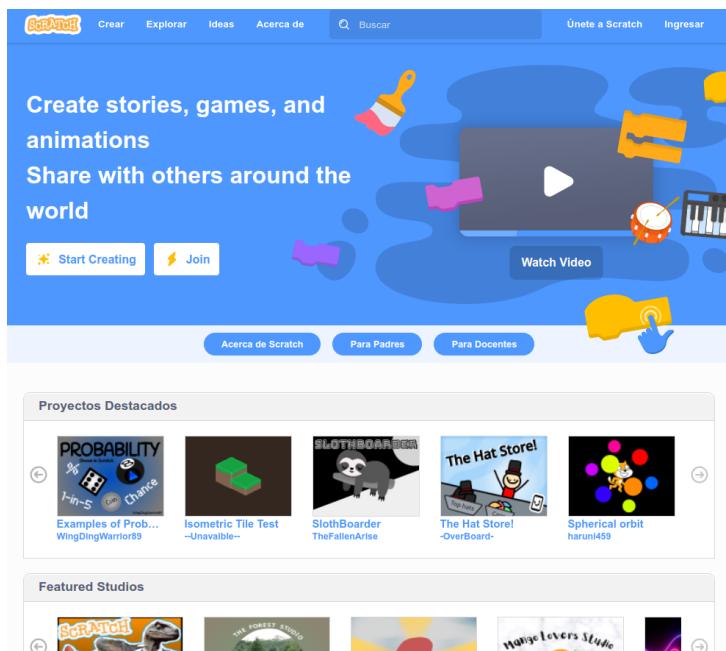
Explorar la galería de proyectos de Scratch con el propósito de conocer la herramienta y algunas actividades que se pueden trabajar en este entorno.



Para utilizar Scratch pueden acceder a la versión 3.0, disponible online en <https://scratch.mit.edu/>, o descargar e instalar la versión offline ingresando a <https://scratch.mit.edu/download> (o accediendo desde la parte inferior de la página principal, en la opción “Descargar”). Tengan en cuenta que la versión offline no está disponible para sistemas operativos Linux ni celulares con Android (solo tablets).



Para acceder a la galería de proyectos, pueden explorar los proyectos que se encuentran en la página principal como “Proyectos destacados”, o bien, ingresar a la opción “Explorar” de la parte superior. Una vez que seleccionan un proyecto pueden ejecutarlo presionando sobre la bandera verde.



The screenshot shows the Scratch website homepage. At the top, there's a blue banner with the text "Create stories, games, and animations" and "Share with others around the world". Below the banner are two buttons: "Start Creating" and "Join". The main navigation bar includes "Crear", "Explorar", "Ideas", "Acerca de", "Buscar", "Únete a Scratch", and "Ingresar". Below the banner, there's a section titled "Proyectos Destacados" (Featured Projects) displaying five thumbnail images of Scratch projects. Further down, there's a section titled "Featured Studios" showing several studio icons.



### Actividad en Foro:

Una vez que accedan a la galería, elijan un proyecto que les haya llamado la atención y comparten en el foro:

- ¿Cuál es el proyecto que eligieron? (pueden compartir una captura de pantalla o link).
- ¿Propone algún desafío? ¿Cuál?
- ¿Qué dinámica tiene? ¿Qué cosas se pueden hacer? ¿Requiere de la participación del usuario? ¿De qué forma puede interactuar?
- ¿Pudieron ver cómo está construido un proyecto por dentro? ¿Encontraron similitudes con Pilas Bloques?



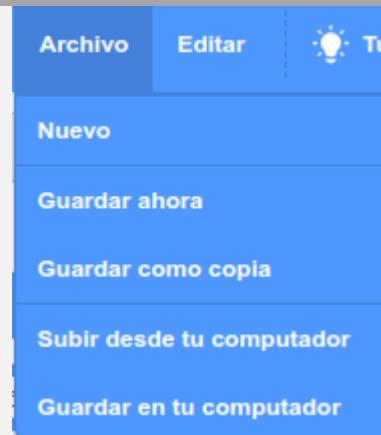
### Crear una cuenta en Scratch

Tener una cuenta les permitirá guardar los proyectos que realicen y poder compartirlos con otras personas. Primero deberán ingresar a la opción “Únete a Scratch” desde la barra superior de la página principal o bien ingresando a <https://scratch.mit.edu/join>.

[Únete a Scratch](#)    [Ingresar](#)

Una vez que hayan ingresado deberán completar los pasos para seleccionar un nombre de usuario, contraseña, país, fecha de nacimiento, género y correo electrónico. Luego deberán confirmar la creación de la cuenta desde su correo.

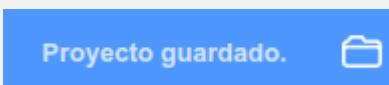
Todos los proyectos que realicen los pueden guardar en su cuenta ingresando a la opción “Guardar ahora”:



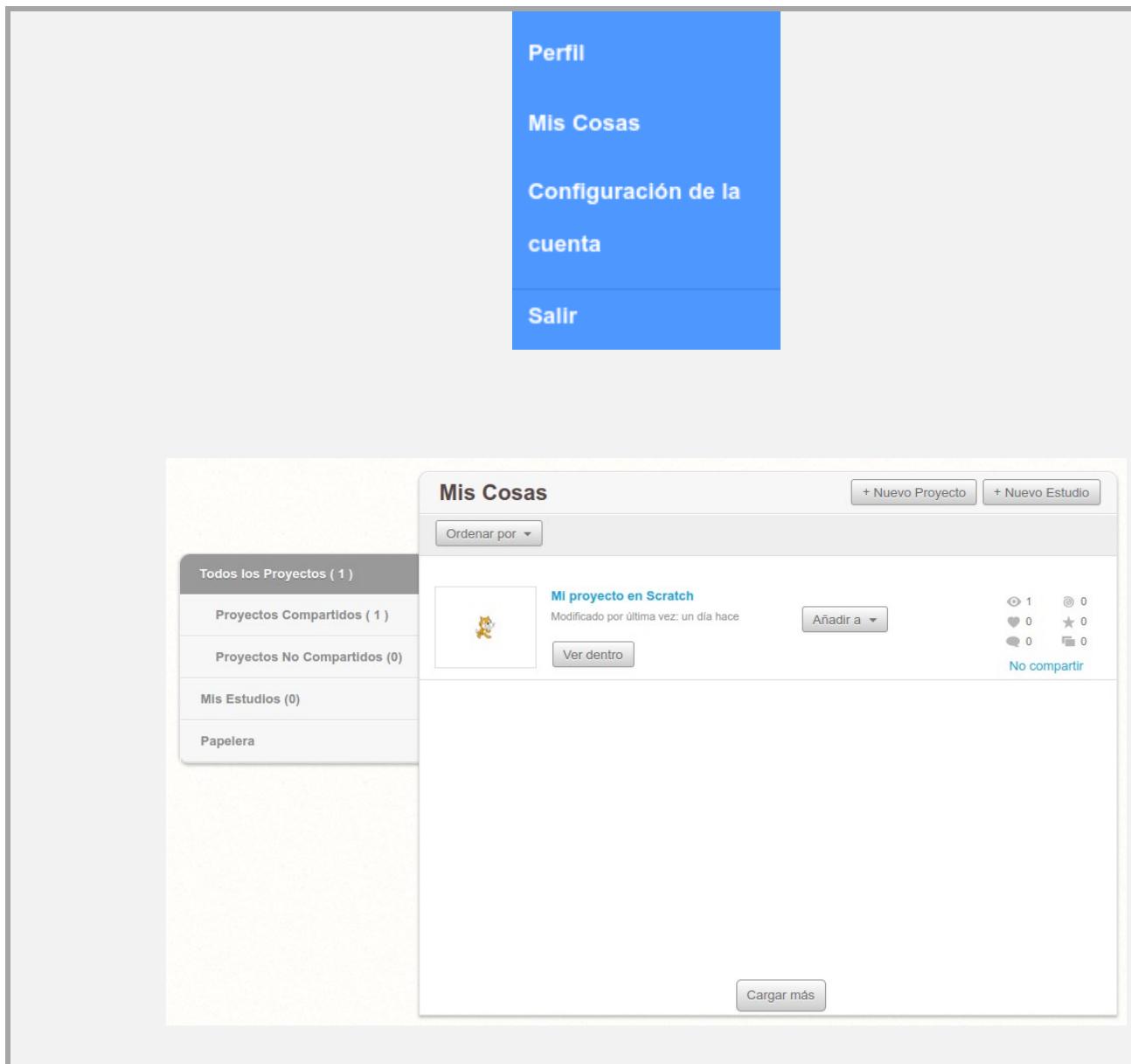
O desde la parte superior derecha en



Cuando el proyecto se haya guardado se mostrará:



Para acceder a su repositorio de proyectos deberán seleccionar su nombre de usuario que se encuentra en la barra superior y seleccionar “*Mis cosas*”:

The screenshot shows the INFoD platform interface. On the left, there is a sidebar with the following options:

- Perfil
- Mis Cosas** (highlighted in blue)
- Configuración de la cuenta
- Salir

The main content area is titled "Mis Cosas". It includes the following elements:

- A header with buttons for "+ Nuevo Proyecto" and "+ Nuevo Estudio".
- An "Ordenar por" dropdown menu.
- A sidebar on the left under "Todos los Proyectos (1)" containing:
  - Proyectos Compartidos (1)
  - Proyectos No Compartidos (0)
  - Mis Estudios (0)
  - Papelera
- A list item for "Mi proyecto en Scratch" with a thumbnail image of a cat-like sprite, a description "Modificado por última vez: un día hace", a "Ver dentro" button, and sharing statistics: 1 like, 0 comments, 0 stars, and a "No compartir" link.
- A "Cargar más" button at the bottom right.

## Segunda parte: primeros pasos programando en Scratch

Les proponemos una secuencia de desafíos de programación en Scratch. El objetivo es que comiencen a familiarizarse con el entorno, por ejemplo, creando y ejecutando un programa o modificando el escenario principal y los personajes. Para esto, durante el encuentro sincrónico, su tutor/a le asignará a cada participante unas tarjetas con un desafío y pistas para lograrlo. Pueden consultar las tarjetas en [este enlace](#).



### Cambiar tamaño

**Desafío 1:** Hacer que el tamaño del personaje aumente cuando se presione la tecla G.




**Indicaciones:**

1. Elegí un escenario.
2. Agregá un personaje.
3. Hacé que su tamaño aumente cada vez que presiones la tecla G.

**Probá con estos bloques:**

**Desafío 2:** Hacer que el personaje vuelva a su tamaño original al ejecutar el programa desde el inicio (haciendo clic en ).

**Probá con estos bloques:**

**Desafío 3:** Hacer que el personaje reduzca su tamaño cuando se presione la tecla P.

**Pista:** Probá cambiando el valor del bloque **cambiar tamaño**. ¿Admite valores negativos?

### Repetición simple y procedimientos | 01

**Desafío 1:** Hacer que el pterodáctilo haga un berrinche y luego se calme.




**Indicaciones:**

1. Elegí un escenario.
2. Agregá un pterodáctilo (*Dinosaur3*).
3. Hacé que se enoje (disfras *dinosaur3-d*) y vuela a 10 posiciones aleatorias al hacer clic sobre .
4. Hacé que luego se calme (disfras *dinosaur3-e*) en el centro del escenario.

**Probá con estos bloques:**

**Desafío 2:** Crear y utilizar los bloques:

**Hacer berrinche**    **Calmarse**

Además, estos desafíos los introducirán en la dinámica de trabajo con programas interactivos, en los que deberán lograr que los personajes realicen diferentes acciones al presionar una tecla o mover el mouse. Luego, trabajaremos sobre otro conjunto de tarjetas con actividades en Scratch en la que deberán resolver nuevamente desafíos de programación con el objetivo de recuperar conceptos trabajados en Pilas Bloques: *procedimientos, repetición simple y alternativa condicional*. Pueden consultar estas tarjetas [aquí](#).



**Encuentro sincrónico: Primeros desafíos en Scratch**

**Fecha del encuentro:**



En este encuentro sincrónico abordaremos entre todos el acercamiento al entorno Scratch. Cada uno/a de ustedes resolverán algunos de los desafíos de las tarjetas y ensayaremos entre todos esta dinámica de introducción a la herramienta para que puedan reproducirla con sus potenciales estudiantes.



### Actividad con entrega en Debate obligatorio

En función de las tarjetas que les asigne su tutor/a de cada una de las secuencias, publiquen en el foro una captura de la solución que elaboraron y comenten por cada una: *¿Qué tipos de bloques utilizaron? ¿Son similares a los que utilizaron en Pilas Bloques? ¿Qué diferencias tienen?*

## Actividad 2

Luego del encuentro sincrónico y a partir de lo trabajado con su tutor/a, les proponemos las siguientes actividades. Quienes no hayan podido asistir, pueden revisar el documento con todas las tarjetas y la grabación del encuentro.



1. Registren en su bitácora las reflexiones centrales de la actividad de introducción a Scratch que realizaron con la dinámica de las tarjetas. A modo de guía, les proponemos una serie de preguntas:
  - ¿Les resultó directo el cambio de entorno? ¿Con qué dificultades se encontraron? ¿Cómo las resolvieron? ¿Cuáles creen que se pueden repetir en el aula con sus estudiantes? ¿Cómo las preverían?



- ¿Con qué objetivo se introduce el concepto de **evento**? ¿En qué se diferencian los programas que utilizan eventos de los que realizaron en Pilas Bloques?
- ¿Fue necesario abordarlo de manera específica? ¿Por qué? ¿Creen que son intuitivas las herramientas que nos da el entorno para trabajarla? ¿En qué momento de la secuencia se conceptualiza la idea?

**a. PARTE 1: Introducción a Scratch y eventos**

- ¿Cómo están organizadas las tarjetas? ¿Qué propósito tiene cada parte?
- ¿Con qué dificultades se encontraron en los desafíos? ¿Cómo las resolvieron? ¿Cómo las preverían? ¿Qué creen que deberían tener en cuenta para replicar esta dinámica en el aula con sus estudiantes?
- ¿Cómo se recupera el concepto de “comando básico”? ¿Qué cosas explicitarían para reforzar la transferencia de este concepto a Scratch?
- ¿Se recuperan características de la didáctica por indagación? ¿Cuáles? ¿En qué momentos?

**b. PARTE 2: Procedimientos, repetición simple, alternativa condicional en Scratch**

- ¿Cómo están organizadas las tarjetas? ¿Qué objetivo tiene cada parte?
- ¿Cómo se introducen los conceptos de procedimiento, repetición simple y alternativa condicional? ¿Qué características del problema motivaron el uso de estos conceptos? ¿Qué dificultades encontraron? ¿Cómo las resolvieron? ¿Qué acciones deberían tener en cuenta para prever estas dificultades en el aula con sus estudiantes?



- ¿Les parece que las tarjetas son suficientes para transferir los conceptos previos a Scratch? ¿Por qué? ¿Qué acciones realizarían para reforzar esa transferencia?

2. Ubiquen en el mapa conceptual de su bitácora los conceptos principales de la clase:

a. **Evento**

b. **Programa interactivo**



Los **eventos** permiten ejecutar una secuencia de comandos cuando se dan ciertos sucesos, por ejemplo: mover el mouse, hacer clic, presionar una tecla, etc.

al presionar tecla      espacio ▾

al hacer clic en este objeto

Podemos observar que cada evento se refiere a una acción del usuario y debe contener una secuencia de comandos que se ejecutará cuando esa acción suceda, por ejemplo, que *cuando* se presione la tecla espacio el personaje aumente su tamaño y se mueva 10 pasos:



De la misma manera podemos agregar otros eventos a un programa y lograr que reaccione a diferentes sucesos. A este tipo de programas los llamamos **interactivos**.

### Actividad 3

Con base en las tarjetas de Scratch que utilizaron en la segunda parte de la Actividad 1, elaboren una nueva tarjeta que agregarán como parte del primer conjunto.



#### Actividad con entrega

Hagan una copia de [este documento compartido](#), donde encontrarán el diseño de una tarjeta en blanco, y complétenla con el desafío que propusieron.

Además, con el propósito de argumentar sus decisiones, respondan en el mismo documento:

- ¿Qué conceptos recupera? ¿Cuáles introduce?
- ¿De qué partes del entorno depende?
- ¿Qué dificultades creen que podrían tener sus estudiantes al resolver el desafío?  
¿Cómo preverían esto?



Finalmente, entreguen el documento para obtener una devolución de su tutor/a.

## Comparación de entornos de programación

Pilas Bloques y Scratch son entornos creados para la enseñanza de la programación que comparten características, como la programación por bloques. Sin embargo, el tipo de actividades y la dinámica que podemos proponer en cada entorno es diferente. En este sentido, resulta pertinente conocer cuáles son las decisiones sobre las que se funda cada entorno y cuáles son los límites y posibilidades que debemos tener en cuenta al momento de diseñar una actividad de programación en cada uno de ellos.

### Actividad 4

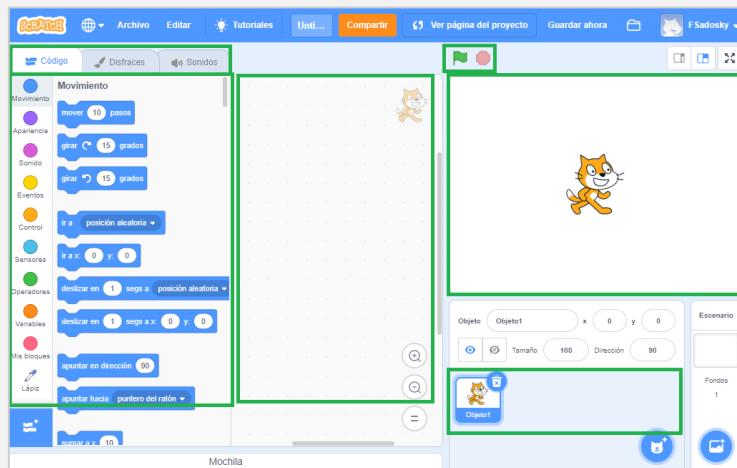


Registren en la bitácora las siguientes reflexiones sobre los entornos de programación.

1. Con el propósito de comprender y contrastar algunas de las decisiones sobre las que se apoyan Pilas Bloques y Scratch, les proponemos responder las siguientes preguntas en base a su experiencia con estas herramientas y analizando las dos lecturas del material de referencia: "[Reviviendo el sueño de Papert](#)" y "[Pilas Bloques: una plataforma de aprendizaje infantil basada en escenarios](#)".
  - ¿Cuál es la propuesta pedagógica sobre la que se apoya cada herramienta? ¿Qué necesidades motivaron su desarrollo? ¿Qué capacidades se buscan potenciar en cada entorno?
  - ¿Cómo aparecen en **Scratch** cada una de las partes marcadas en color rojo? ¿Cuáles no aparecen? ¿Cómo relacionarían estas decisiones con la propuesta pedagógica de Pilas Bloques?



- ¿Cómo aparecen en **Pilas Bloques** cada una de las partes marcadas en color verde? ¿Cuáles no aparecen? ¿Cómo relacionarían estas decisiones con la propuesta pedagógica de Scratch?



- ¿Tienen características para determinar si la solución que plantearon es correcta? ¿Cuáles? ¿Piensan que tiene alguna relación con la propuesta pedagógica? ¿Creen que estas características reemplazan una posible intervención del docente?

2. Modifiquen el siguiente dibujo y asocien según su criterio las siguientes características/propiedades a PilasBloques y/o Scratch:

•



**Aprendizaje por indagación**  
**Resolución de desafíos**  
**Conceptos de programación**  
**Creación de proyectos**  
**Creatividad**  
**Bloques**  
**Personalización**  
**Programas interactivos**  
**Indicador de desafío logrado**  
**Entorno abierto**  
**Entorno cerrado**

*Observación:* Algunas de esas características podrían ser compartidas.

3. Piensen y agreguen una característica/propiedad a cada entorno que no esté en el listado. Además, respondan: ¿Esa característica se podría relacionar con alguna otra de la lista? ¿La asociaría a un aspecto de diseño del entorno o a una decisión pedagógica? ¿Por qué?



#### Detenerse a pensar...

Como pudieron analizar, Pilas Bloques y Scratch son herramientas que se apoyan en propuestas pedagógicas diferentes, pero que comparten un objetivo: son un medio sobre el cual podemos acercar conceptos básicos de programación.

Además, observaron que algunos de esos conceptos están disponibles en ambos entornos y es posible abordarlos de diferentes maneras. Incluso, algunos conceptos como el de evento no están disponibles por el momento en Pilas Bloques, lo que nos permite plantear la siguiente pregunta: **¿los conceptos de programación son específicos de un entorno?** La respuesta es que no. Si bien la posibilidad de trabajar un concepto en un entorno queda restringido a su disponibilidad (en caso de abordarlo a través de



actividades con computadoras) es importante destacar, como mencionamos al comienzo, que estas herramientas son un medio para acercar estos conceptos de programación.

Por lo tanto, teniendo en cuenta que los conceptos de programación no son específicos de un entorno, conocer límites y posibilidades de estas herramientas es una oportunidad para analizar cómo nos pueden ayudar a alcanzar los objetivos establecidos, cómo podemos abordar las propuestas de trabajo en el aula y de qué manera se interpela a los y las estudiantes en cada una de las actividades.

## Reflexión de cierre



Esta clase debiera permitirles:

**A.** Identificar y comprender los principales conceptos trabajados:

- **Evento:** bloque que contiene una secuencia de comandos y su ejecución depende de algún suceso (por ejemplo, presionar una tecla o mover el mouse).
- **Programa interactivo:** programa en el cual una persona puede influir directamente durante su ejecución.

**B.** Aproximarse al entorno de programación Scratch y a la creación de proyectos simples de programación.

**C.** Identificar diferencias y similitudes entre Pilas Bloques y Scratch y relacionarlas con los objetivos con los que fueron creados.

**D.** Conocer los límites y posibilidades que debemos tener en cuenta al momento de diseñar una actividad de programación en Pilas Bloques o Scratch.

**E.** Reconocer que los conceptos adquiridos (dentro del marco conceptual propuesto) no son específicos de un lenguaje o un entorno de trabajo particular, sino que son comunes a muchos entornos de programación usuales.



## Material de lectura

Resnick, M. (2012). [Reviviendo el sueño de Papert](#). *Educational Technology*, vol. 52, no. 4, pp. 42-46.

Sanzo, A.; Schapachnik, F.; Factorovich, P., & O'Connor, F. S. (2017). [Pilas Bloques: una plataforma de aprendizaje para niños y niñas basada en escenarios](#). 2017 Twelfth Latin American Conference on Learning Technologies (LACLO). doi:10.1109/laclo.2017.8120926

## Bibliografía de referencia

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

Rusk, N. (2016). [Tarjetas de programación de Scratch](#). No Starch Press; 1º. edición.

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto: Fundación Sadosky (2022). Clase Nro. 3: Programas interactivos. Módulo: Estrategias para la enseñanza de la programación II. 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-CompartirlIgual 3.0



## Módulo 4: Estrategias para la enseñanza de la programación II

# Clase 4: Creamos un videojuego

## Introducción

En esta clase desarrollaremos un pequeño videojuego en Scratch en el que un personaje debe alcanzar un objetivo superando obstáculos en el camino. Para esto, iremos construyendo el proyecto por partes, resolviendo pequeños desafíos que recuperan herramientas de programación vistas a lo largo del módulo. Este recorrido les permitirá vivenciar un modo de integrar aprendizajes de forma gradual y podrá utilizarse como base para desarrollar el Trabajo Final.

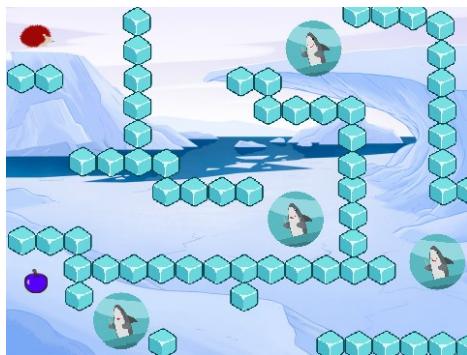
¡Comencemos a desarrollar el videojuego!

## Actividad

### Paso 1

El primer objetivo es que “Nacles”, el personaje jugador, se mueva con el teclado en las cuatro direcciones básicas (arriba, abajo, izquierda, y derecha). Para comenzar, creen una copia de [este proyecto](#) en su repositorio seleccionando el botón “Reinventar”. Tengan en cuenta que la opción solo estará visible si iniciaron sesión con su cuenta de Scratch.

Podrán observar que el proyecto ya viene con el laberinto construido, el personaje ubicado en su punto de partida y un programa ya creado (¿pueden identificar cuál es el propósito de este programa?). Con esto ya tienen todo listo para comenzar.



### Movemos al personaje

Hagan que Nacles se mueva hacia arriba, abajo, a la izquierda y a la derecha con las flechas del teclado. Por el momento, no importa si el personaje se mueve por encima de los cubos de hielo o se sale del escenario.



No se olviden de guardar los cambios que vayan realizando, seleccionando la opción “Guardar ahora” que se encuentra en la parte superior derecha de la pantalla (a la izquierda de su nombre de usuario).



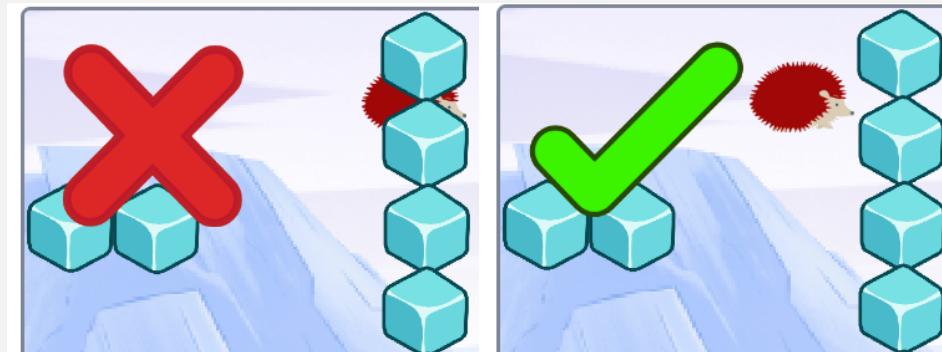
## Paso 2

Luego de que hayan conseguido que el/la jugador/a pueda mover al personaje por el laberinto, es momento de agregar algunas modificaciones para evitar que atraviese los cubos de hielo. Para lograr este objetivo deben recuperar y combinar algunos conceptos de programación vistos en clases anteriores.



### Chocando con el hielo

Modifiquen el programa anterior para que Nacles retroceda 10 pasos al tocar un cubo de hielo. Tengan en cuenta los conceptos de programación vistos para priorizar la legibilidad del programa y no haya fragmentos de bloques repetidos.



Si Nacles se mueve a la derecha y choca un cubo de hielo, *¿hacia dónde debería retroceder? ¿Qué debería pasar con las demás direcciones?*



*¿Cuántas formas de saber si el personaje está chocando con el hielo encontraron? ¿Cuál les resultó más intuitiva? ¿En qué casos creen que conviene utilizar cada una?*



Registren en su bitácora:

- ¿Qué herramientas de programación utilizaron para detectar el choque (colisión) entre objetos? ¿En qué partes del programa tuvieron que utilizarlos? ¿Tuvieron dificultades? ¿Cuáles? ¿Cómo las preverían en el aula con sus estudiantes?



- ¿Su programa quedó legible y cada subtarea es fácilmente identificable?  
¿Repetieron fragmentos de bloques en cada evento? ¿Cómo lo hicieron? ¿Les resultó intuitivo? ¿Qué cosas podrían recuperar de lo trabajado en Pilas Bloques? ¿Cómo trabajarían la estrategia de solución?

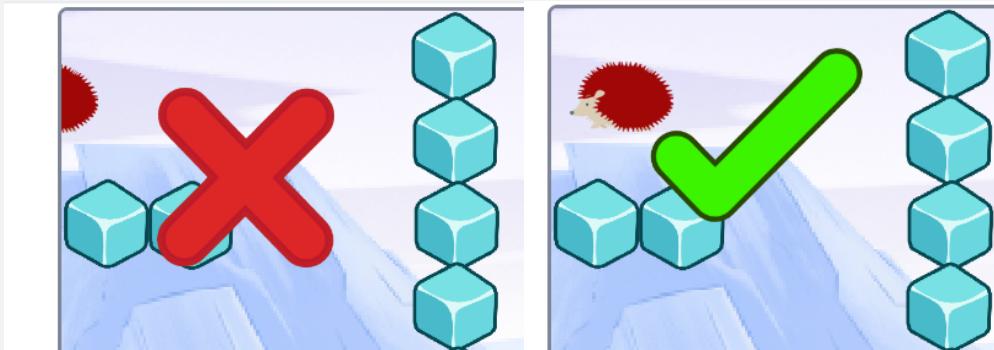
## Paso 3

Además de rebotar con los cubos de hielo para evitar atravesarlos cuando se mueve, deben lograr que el personaje no se salga del escenario.



### Chocando con el hielo y el borde

Modifiquen el programa para que Nacles también retroceda 10 pasos al tocar el borde del escenario.





## Clase de consultas



### Consultas sincrónicas

**Fecha del encuentro:** **COMPLETAR**

Durante la clase de consultas 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.

## Paso 4

En los pasos anteriores se preocuparon por programar al personaje para que puedan moverlo por el laberinto y rebotar cuando choque con el borde y las paredes del laberinto. Ahora, deben lograr que unos obstáculos aparezcan y desaparezcan continuamente.



### Tiburones en el escenario

Hagan que los tiburones hambrientos aparezcan, se queden en su lugar por 3 segundos y desaparezcan por otros 3 segundos. Los tiburones deben repetir este comportamiento durante todo el juego.

En este paso preocúpense únicamente por que aparezcan y desaparezcan. En el paso siguiente los convertirán en verdaderos enemigos con consecuencias para Nacles.



Exploren la categoría “**Apariencia**” y busquen bloques que hagan *aparecer* y *desaparecer* a un objeto.



Exploren la categoría “Control” y busquen algún bloque que les permita *esperar* una cantidad de tiempo indicada.

## Paso 5

Para agregarle dificultad al juego y evitar que Nacles alcance el objetivo fácilmente, si se choca con algún tiburón deben hacer que comience nuevamente el recorrido.



### Volver a empezar

Hagan que Nacles vuelva al comienzo del laberinto si choca con alguno de los tiburones.



## Paso 6

Si Nacles logra evitar a los tiburones y alcanzar el objetivo, deberá ponerlo de manifiesto.



### Cumpliendo el objetivo



Hagan que Nacles diga “*¡Conseguí la manzana!*”, reproduzca un sonido y termine la ejecución del programa cuando llegue a la manzana violeta.



Exploren la categoría “**Control**” y busquen algún bloque que *detenga* la ejecución del programa completo.



En función de las actividades del recorrido, les sugerimos registrar en su bitácora sus reflexiones a partir de las siguientes preguntas:

- ¿Qué conceptos principales se recuperan en cada actividad?
- ¿Les parece que las actividades presentan una dificultad gradual? ¿Por qué?
- ¿De qué manera se articula cada actividad con la anterior? ¿Cambiarían de orden algunas actividades? ¿Por qué?



#### Actividad individual con entrega obligatoria

Compartan con su tutor/a un enlace a su proyecto de Scratch. En el recuadro siguiente se muestran los pasos para obtener un enlace.



Su tutor/a tendrá en cuenta los siguientes factores en las correcciones:

- Que el programa elaborado solucione cada una de las seis actividades.
- Que se hayan utilizado las herramientas de programación pertinentes de forma correcta.
- Que el programa elaborado sea legible.



### Compartir un proyecto en Scratch

Para compartir un proyecto, primero deben abrirlo y seleccionar el botón “Compartir” que se encuentra en la parte superior de la pantalla:

Compartir

Luego, en la pantalla que se muestra podrán modificar el nombre del proyecto, agregar instrucciones de uso e información sobre el proyecto (notas y créditos). A continuación, pueden compartir el enlace del proyecto copiándolo desde la barra del navegador o seleccionando el botón “Copy Link” que se encuentra en la parte inferior izquierda.

¡Felicitaciones, compartiste tu proyecto! Otras personas podrán probarlo, dar comentarios y reinventarlo.

 Mi proyecto en Scratch Ver dentro

Flags: 0 | Likes: 0 | Stars: 0 | Comments: 1 | Unshared | Add to Studio | Copy Link

**Instrucciones**  
*Tell people how to use your project (such as which keys to press).*

**Notas y créditos**  
*How did you make this project? Did you use ideas, scripts or artwork from other people? Thank them here.*



## Reflexión de cierre



Esta clase debiera permitirles:

- A. Ejercitarse en la creación de programas interactivos en Scratch utilizando eventos.
- B. Concebir un proyecto de programación en Scratch como un recorrido gradual y coherente de actividades.
- C. Aplicar las herramientas de programación vistas para dinamizar el proyecto, por ejemplo, implementar la **colisión** entre objetos a través de alternativas condicionales y sensores, o programar objetos que se mueven de forma autónoma.

## Bibliografía de referencia

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

## Créditos

Autores: Fundación Sadosky

Cómo citar este texto: Fundación Sadosky (2022). Clase Nro. 4: Creamos un videojuego. Módulo: Estrategias para la enseñanza de la programación II. 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-CompartirlGual 3.0