

**PROPUESTA PEDAGÓGICA**

# ¿Cómo escapar del laberinto?

**iHola!**

Te damos la bienvenida a esta propuesta pedagógica. En ella encontrarás una síntesis del recorrido y el desarrollo de cada una de sus etapas, las dimensiones que trabaja, así como las soluciones y material anexo.

Esperamos que te resulte un recurso clave para la implementación de ALTEC en el aula.

# Guía de lectura

Durante el recorrido, te encontrarás con diversos bloques que indicarán diferentes momentos de la metodología:

**Momento de un desafío:** son los bloques punteados. Estos desafíos se proponen directamente al grupo de estudiantes.

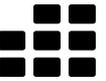
**Reflexionar con el grupo en clase:** son los bloques de color gris claro. En estos momentos, se espera que estas preguntas sirvan de guía para trabajar con todo el grupo.

**Seleccionar un viaje en RoboGarden:** son los bloques verdes.

**Preguntas clave de la etapa:** son los recuadros amarillos. Estas preguntas pretenden asentar los conocimientos trabajados en clase, así como ser disparadoras de nuevas preguntas del grupo.

**Elementos clave de la metacognición:** son los bloques de color gris oscuro. En estos recuadros se establecen las habilidades de pensamiento computacional desarrolladas o evaluadas en la etapa, así como los conocimientos de programación.

**Trabajo interdisciplinario entre distintas asignaturas:** son los bloques de color gris oscuro. Con estos bloques se presentan ideas para trabajar junto con otras áreas de conocimiento.



La clase tendrá cuatro dinámicas:

**ALT+P**

es una dinámica dedicada a comunicar, compartir, reflexionar en grupo.

**ALT+T**

es una dinámica orientada al uso de la tecnología, a la acción.

**ALT+C**

es la dinámica de cierre, aprendizajes y conclusiones.

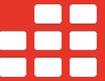
**ALT+PC**

es un recurso docente de estrategias para profundizar el trabajo colaborativo y la vinculación entre la etapa y el pensamiento computacional.

# Índice

## **¿Cómo escapar del laberinto?**

Descripción del proyecto	6
Dimensiones de Ciencias de la Computación	7
Síntesis del recorrido	9
Etapa 1	11
Etapa 2	22
Etapa 3	34
Etapa 4	44
Etapa 5	52
Etapa 6	62
Etapa 7	72
<b>Soluciones a los desafíos</b>	<b>76</b>
<b>Anexos</b>	<b>82</b>



# ¿Cómo escapar del laberinto?

## Descripción del proyecto

En esta propuesta, el grupo de estudiantes diseñan y programan un juego en RoboGarden, donde el robot tiene que escapar del laberinto en la menor cantidad posible de pasos. La construcción del juego les permite comprender y utilizar conceptos computacionales tales como las secuencias de comandos, el trabajo con eventos, la estructura de repetición y la eficiencia de un algoritmo. En el recorrido de la propuesta, interpretan y diseñan diagramas de flujo para representar algoritmos y simulan el recorrido de laberintos al aplicar distintos algoritmos conocidos.

**Duración aproximada: 720 - minutos ~ 7 semanas**

**Entorno de programación: RoboGarden**

# Dimensiones de las ciencias de la computación

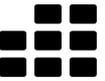
Espacio: Técnico  
Tecnológico

Unidad curricular: Ciencias  
de la Computación

Tramo 5 | Grado 7.º - Énfasis en alfabetización digital

## **Criterios de logro:**

- Conoce la forma en que las computadoras intercambian información.
- Propone e implementa soluciones de mejora, de forma individual y colaborativa, poniendo en juego su imaginación y creatividad.
- Asume sus responsabilidades y distintos roles en trabajos colectivos.
- Desarrolla programas sencillos, integrando adecuadamente tipos de datos, instrucciones secuenciales, estructuras de selección y de repetición.
- Emplea en sus proyectos estrategias y técnicas (planificación, análisis, abstracción, división en subtareas, depuración, reutilización de soluciones, presentación y documentación de la solución, roles y trabajo colaborativo) características del desarrollo de software.



### **Competencias generales:**

Comunicación · Pensamiento computacional · Metacognitiva

### **Competencias específicas:**

- Comprende y analiza el funcionamiento de las computadoras, reconoce los distintos componentes y sus funcionalidades e identifica cómo se procesan datos y cómo se intercambia información en las redes informáticas.
- Expresa y comunica ideas en diferentes formatos digitales, de forma individual o compartiendo responsabilidades en construcciones colectivas, explorando su potencial creativo mediante el uso pertinente de una o múltiples aplicaciones.
- Identifica y resuelve en grupo, de forma mediada, con ideas creativas, problemas sencillos del mundo real que pueden ser solucionados mediante la lógica computacional, usando tecnología, herramientas digitales, automatismos y lenguajes informáticos, partiendo de sus experiencias previas similares.

### **Contenidos: Algoritmia, programación, robótica y problemas computacionales**

- Algoritmia: Concepto de algoritmo, funcionamiento e implicancia. Ejemplos de algoritmos de ordenamiento vinculados al pensamiento computacional.
- Estrategias de búsqueda simples.
- Lenguajes de programación por bloques, tipos de datos, variables, estructuras selectivas e iterativas. Expresiones lógicas y usos en programación. Desarrollo de programas sencillos.

## Síntesis de recorrido

1

¿Cómo nos comunicamos con las computadoras?

- Instrucciones y comandos primitivos.
- Introducción a algoritmos.
- Presentación de RoboGarden.

**80 min.**

2

¿Qué tienen en común los telares y las computadoras?

- Historia de Ada Lovelace.
- Representación binaria.
- Bit.
- Bloque de repetición.

**80 min.**

3

¿Cuál es la "trama" de un algoritmo?

- Diagramas de flujo.
- Eficiencia de un algoritmo.

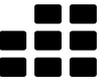
**120 min.**

4

¿Cómo desentramar un laberinto?

- Diseño de los niveles del juego.

**80 min.**



5

Un laberinto,  
¿muchos  
caminos?

- Algoritmos para resolver laberintos.
- Camino más corto.

**120 min.**

6

Creación de un  
juego de  
laberintos.

- Eventos.
- Programación de las reglas del juego.

**160 min.**

7

Socialización de  
los juegos.

- Socialización de los juegos.
- Evaluación.

**80 min.**

# Etapa 1. ¿Cómo nos comunicamos con las computadoras?

Tiempo estimado: 80 minutos

## Resumen de la etapa

En esta etapa se trabaja el concepto de *instrucción* y de *comandos primitivos* y se introduce la noción de *algoritmo*. Se comparan distintos algoritmos en relación con la cantidad de bloques utilizados, para comenzar a comprender la idea de algoritmos eficientes. Se presenta la plataforma RoboGarden, que será el entorno de programación que acompañará la propuesta pedagógica.



ALT+P

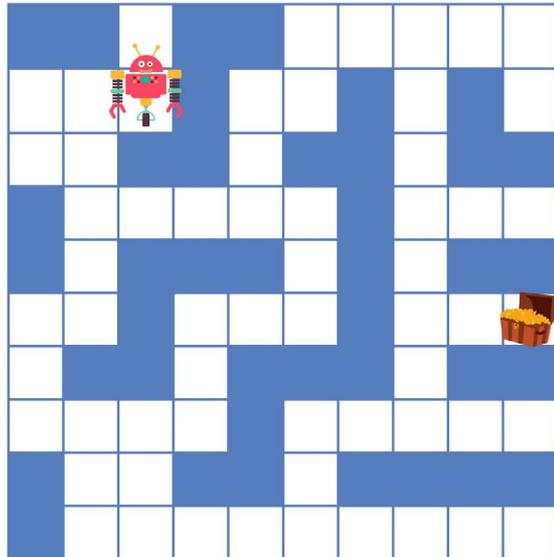
## Alta Personalización

Duración aproximada: 30 minutos

### > Aproximación a las instrucciones ( ) {

Para comenzar con el abordaje del concepto de instrucción se trabaja en una actividad en duplas de estudiantes. Cada equipo tiene una copia del laberinto o pueden crear sus propios laberintos. Cada estudiante tendrá un rol: será el ordenador (robot) o quien dictará las órdenes.

Ayudemos a Robo a llegar al tesoro



Escribe las instrucciones para que el robot llegue hasta el tesoro del laberinto.

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Desafío 1.1

Antes de proponer el segundo desafío, se puede hacer una puesta en común con el grupo:

¿Todos los grupos pudieron llegar al tesoro? ¿Pudieron establecer una forma de comunicarse efectiva para orientar al robot hacia el tesoro? ¿Existieron dificultades al respecto?

El rol del robot y de quien da las órdenes se puede asociar a la computadora y a quien programa, respectivamente.

\*Para reflexionar con el grupo de clase

Se propone el siguiente desafío para complejizar el problema:

## ¡Ojo con las instrucciones que se le dan a Robo!

En esta instancia, el robot sólo puede seguir las instrucciones dados.

Girar 90° en sentido horario

Avanza 2

Retrocede 1

Girar 90° en sentido antihorario

Escribe las instrucciones para que el robot llegue hasta el tesoro del laberinto.

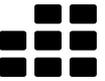
---

---

---

---

---



Una vez que el grupo de estudiantes tengan escritos los programas de ambos desafíos, se propone trabajar con ellos. Puede ser intercambiando desafíos entre equipos o pegando todas las soluciones en el pizarrón. Se invita a reflexionar al respecto.

¿Todos los programas son iguales? ¿Todos los programas llevan al robot al tesoro? ¿Hay una única solución para los desafíos? ¿Las instrucciones que se le dan al robot pueden estar en cualquier orden? ¿Se obtiene el mismo resultado si se altera el orden de las instrucciones? ¿Para todos los laberintos se puede utilizar la misma solución?

\*Para reflexionar con el grupo de clase

Se presenta el concepto de algoritmo, a partir de los ejemplos, como un *conjunto de instrucciones definidas y ordenadas para resolver un problema. En función del estado inicial, o entrada dada, se ejecutan las instrucciones para obtener una salida.*

Se puede utilizar el siguiente [recurso](#) en el aula.

}

ALT+T

## Alta Tecnología

Duración aproximada: 40 minutos

### > Comenzando el viaje () {

Se presenta RoboGarden, una herramienta para que los estudiantes aprendan a programar y adquieran habilidades de pensamiento computacional.

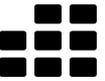
Se asigna al grupo de clase el viaje "Explorando el laberinto".

Los estudiantes trabajarán en las aventuras 1, 2, 3 y 4 del viaje. En estas misiones se utilizan los bloques de movimiento del robot.

Finalmente, se puede reflexionar sobre los comandos usados en los primeros desafíos y en RoboGarden.

¿Qué instrucciones se pueden dar en RoboGarden? Si se desea programar a Robo para que avance 5 celdas, ¿qué instrucciones se le pueden dar?

}



ALT+C

## Cierre y reflexión de la etapa

Duración aproximada: 10 minutos

Para cerrar las etapas vamos a trabajar con el siguiente cuadro, que el grupo de estudiantes deberán completar con lo aprendido.

Completa las frases de acuerdo a lo aprendido en cada etapa:

De lo aprendido, le puedo explicar a alguien de la clase...

Three horizontal lines for writing inside a rounded rectangular box with a green border.

De lo aprendido, puedo explicarle a alguien que no estuvo en clase...

Three horizontal lines for writing inside a rounded rectangular box with a blue border.

Aún me queda por entender...

Three horizontal lines for writing inside a rounded rectangular box with a red border.

Se encuentra en formato para imprimir en el [Anexo 1](#).

Una forma de completar el cuadro anterior podría ser la siguiente:

De lo aprendido, le puedo explicar a alguien de la clase...

*que puedo programar en RoboGarden usando los bloques "dar un paso adelante", "dar un paso atrás", "girar a la derecha" y "girar a la izquierda".*

De lo aprendido, puedo explicarle a alguien que no estuvo en clase...

*que el orden de las instrucciones importa. En el caso de RoboGarden, que es una plataforma para aprender a programar, algunos de los comandos son "dar un paso adelante" y "dar un paso atrás".*

Aún me queda por entender...

*si existen otros comandos. ¿No necesito decirle los grados para girar?*

Pedirle a alguien del grupo de estudiantes que comparta lo que escribió para poder cerrar la etapa.



**> Preguntas claves de la etapa () {**

¿La computadora acepta todo tipo de instrucciones?  
¿Cuáles usa en RoboGarden?  
¿Importa el orden de las instrucciones?  
¿Puedo escribir dos secuencias de instrucciones que  
obtengan el mismo resultado?

}

ALT+PC

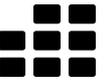
## > Metacognición: elementos claves

Pensamiento computacional y programación:

- Pensamiento algorítmico. Se introduce el concepto de algoritmo. En RoboGarden se utiliza el pensamiento algorítmico para resolver los problemas.
- Generalización. Para los desafíos se generalizan las soluciones
- Comando con entrada constante. Permite comprender la noción de entrada a un método.
- Entorno RoboGarden:
  - ◆ "Dar un paso adelante"
  - ◆ "Dar un paso atrás"
  - ◆ "Girar a la derecha"
  - ◆ "Girar a la izquierda"

\*Los conceptos y habilidades de PC que se introducen.

\*Los bloques nuevos de programación de la etapa.



### > **Profundización**

Estudiantes que quieran profundizar en los conocimientos trabajados en esta etapa pueden completar las aventuras 5 y 6 del viaje "Explorando el laberinto".

\* Para estudiantes que quieran profundizar

### > **Trabajo interdisciplinario**

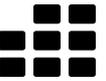
El concepto de algoritmo es compartido con la asignatura Matemática; se puede realizar una comparación con los algoritmos conocidos que usamos, tales como la multiplicación, la adición, la división.

\*Con esta nota se busca el trabajo interdisciplinario entre distintas asignaturas.

## > Nota sobre RoboGarden

En RoboGarden se puede asignar viajes a cada grupo y evaluar el avance de cada estudiante. Conceptos importantes a la hora de trabajar en RoboGarden:

- **Misión:** Es la unidad de aprendizaje. En cada misión se pide al grupo de estudiantes que realicen una tarea mediante la programación. Después de crear el programa, RoboGarden evalúa el código y proporciona una retroalimentación inmediata.
- **Aventura:** Es la lección en RoboGarden. Cada aventura tiene un conjunto de resultados de aprendizaje. Se compone de cinco misiones. Las herramientas utilizadas en estas misiones están detalladas en un tutorial. Cada estudiante debe completar un cuestionario que evalúa los conceptos trabajados en la aventura.
- **Viaje:** Es el curso en RoboGarden, consta de ocho aventuras.
- **Previaje:** Es el entorno del viaje en RoboGarden. Cada mundo está compuesto de una línea de relatos y objetos visuales. Esto involucra a cada estudiante y permite introducir conocimientos complementarios de objetos en diferentes entornos.



## **Etapa 2. ¿Qué tienen en común los telares y las computadoras?**

Tiempo estimado: 80 minutos

### **Resumen de la etapa**

En esta etapa se trabaja el concepto de orden en la secuencia de instrucciones de un algoritmo a través de la representación binaria. Se espera que el grupo de estudiantes encuentren patrones para introducir la estructura de control “repetición”.

ALT+P

Duración aproximada: 30 minutos

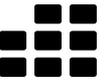
### > **Representación ( ) {**

Comenzar la etapa con las preguntas disparadoras:

¿Sabían que los telares sirvieron de inspiración para los primeros programas de computadora? ¿Qué les parece que tienen en común?

\*Para reflexionar con el grupo de clase

Las computadoras y los telares tienen en común la representación que utilizan para comprender las instrucciones. Para entenderlo mejor y profundizar en ello, trabajaremos a través de una actividad de a pares o grupos.



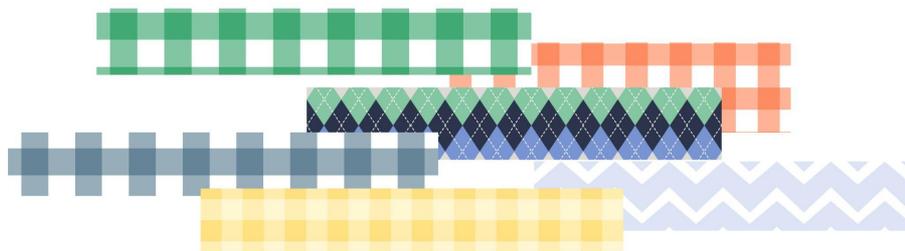
### **DESAFÍO 2.1. Dibujo secreto**

Te proponemos crear tu propio diseño en el telar en Scratch. Con el siguiente [programa de Scratch](#), podrás pintar una celda con un estampado haciendo clic sobre ella. Para quitar el estampado, vuelve a hacer clic sobre ella.

Desafía a alguien de tu clase a que descubra a qué figura corresponde tu diseño.

Si quieres ver la representación que utiliza la computadora para trabajar con el estampado creado, presiona el botón "Representación Computadora".

Dibujo secreto



Desafío 2.1 

\*Desafío para proponer al grupo de estudiantes

Se motiva a la clase a utilizar el botón “Representación Computadora” para reflexionar al respecto.

¿Qué representación usa la computadora cuando hay un estampado en la celda? ¿Y cuando no hay un estampado? ¿Conocen la palabra *bit*? ¿Qué es un bit? Esto permite el trabajo con la representación binaria.

\*Para reflexionar con el grupo de clase

Para profundizar al respecto, se trabajará con la historia de Ada Lovelace. Ada vinculó la máquina analítica de Charles Babbage con el telar de Jacquard. El siguiente [video](#) brinda una explicación visual sobre esto. Otro recurso sobre la vida de Ada es el libro *Ada Lovelace descifra el código* (Favilli, E., & Cavallo, F. (2020). *Ada Lovelace descifra el código*. Planeta México), de donde extraemos la siguiente frase que puede resultar de interés para el grupo de estudiantes: “No solo es una máquina calculadora... La Máquina Analítica teje patrones algebraicos, tal como el telar Jacquard teje flores y hojas en la tela”.

Se sugiere a la clase una actividad que retoma la relación entre las computadoras y los telares, en la cual se les pide que diseñen laberintos. El grupo se divide en parejas y cada quien solicita a su par que reproduzca el laberinto diseñado en la computadora. Para lograr esto, deben emplear estrategias que les permitan comunicar el laberinto de manera precisa.



### **DESAFÍO 2.2 := Diseñando mi propio laberinto**

Utilizando el siguiente [programa de Scratch](#), diseña tu propio laberinto.

Luego, pide a alguien de tu clase que lo reproduzca en su computadora a partir de tus indicaciones. Utiliza la representación binaria para indicar cuando un casillero está pintado o no.

Una vez finalizado, comparen los diseños.

Sugerencias para las indicaciones: ¿Por dónde comienzas a dar las instrucciones? ¿Recorres las filas o las columnas? ¿Qué estrategia puedes utilizar cuando se repite algún patrón en el diseño?

Diseña tu propio laberinto



\*Desafío para proponer al grupo de estudiantes

Finalmente, se puede profundizar al respecto lo siguiente:

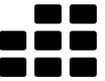
¿Los diseños quedaron idénticos? ¿Qué tan simple les resultó dar las indicaciones? ¿Qué estrategia utilizaron cuando se repetía algún patrón?

En la búsqueda de una respuesta a la pregunta inicial, podemos trabajar con el grupo que las indicaciones que cada estudiante debía dar a sus compañeros en representación binaria se pueden vincular con la lectura de tarjetas perforadas que utilizaban las precursoras de las computadoras tal cual las conocemos hoy.

Los telares y las primeras computadoras tienen en común el uso de tarjetas perforadas para procesar información. Los telares antiguos utilizaban tarjetas perforadas para controlar el patrón de tejido, mientras que las primeras computadoras, como la Máquina Analítica de Charles Babbage, utilizaban tarjetas perforadas para introducir instrucciones y datos.

En ambos casos, las tarjetas perforadas eran una forma de programación que permitía automatizar la ejecución de tareas repetitivas y precisas. Además, en ambos casos, el uso de las tarjetas perforadas abrió el camino para la creación de sistemas más complejos y avanzados de procesamiento de información.

}



ALT+T

Duración aproximada: 30 minutos

> **Más de una vez ( ) {**

Se asigna el previaje "Bucle simple", donde el grupo de estudiantes incorporarán el bloque de repetición.

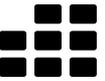
Misiones:

1. Mensaje alienígena. Se utiliza el bloque "escribir en la celda", que permite escribir un valor en una celda. El bloque se encuentra en la categoría Acciones de Robo y tiene una entrada que es un valor numérico (el bloque lo encontramos en la categoría Valores).
2. ¡Un rompecabezas más!
3. Tiempo de sumar.
4. ¿Cuál es el número faltante?

Si bien las misiones se pueden resolver repitiendo los bloques de forma consecutiva, tal como se hizo antes, se espera que el grupo de estudiantes utilicen el bloque "repetir" para programar de forma más eficiente.



}



ALT+C

Duración aproximada: 10 minutos

Para cerrar las etapas vamos a trabajar con el siguiente cuadro, donde cada estudiante deberá completar con lo aprendido.

Completa las frases de acuerdo a lo aprendido en cada etapa:

De lo aprendido, le puedo explicar a alguien de la clase...

Three horizontal lines for writing, enclosed in a green rounded rectangular border.

De lo aprendido, puedo explicarle a alguien que no estuvo en clase...

Three horizontal lines for writing, enclosed in a blue rounded rectangular border.

Aún me queda por entender...

Three horizontal lines for writing, enclosed in a red rounded rectangular border.

Se encuentra en formato para imprimir en el [Anexo 1](#). Se puede pedir a alguien del grupo que comparta lo que escribió para poder cerrar la etapa.

### > Preguntas clave de la etapa () {

¿Qué tienen en común los telares y las computadoras? ¿Cuál es la representación que utilizan?

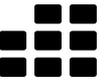
¿Qué bloque se utiliza para no escribir muchas veces la misma instrucción?

Para escribir en la celda en la plataforma RoboGarden, ¿qué bloque se usa? ¿Qué tipo de datos se puede escribir en una celda?

¿Podemos hacer que el robot haga un giro completo y vuelva a la posición inicial? ¿Cómo?

Se puede pedir al grupo de estudiantes que piensen una actividad en la vida diaria donde usen el bloque "repetir" (por ejemplo, en Educación Física, cuando nos indican hacer 20 abdominales, utilizamos el repetir 20 veces).

}



ALT+PC

## > Metacognición: elementos clave

Pensamiento computacional y programación:

- Abstracción. El trabajo con la representación binaria.
- Generalización. Para generar patrones en la guarda.
- Descomposición. Al trabajar con la repetición, se puede descomponer el problema y buscar una solución más simple.
- RoboGarden:
  - ◆ "Repetir"
  - ◆ "Escribir en la celda"

\*Los conceptos y habilidades de PC que se introducen.

\*Los bloques nuevos de programación de la etapa.

## > Profundización

Se puede utilizar el [video](#) de Paenza para profundizar sobre la representación binaria. También se puede utilizar el siguiente [recurso](#) para trabajar la representación de datos.

Estudiantes que quieran profundizar en los conocimientos trabajados en esta etapa pueden completar el previaje "Movimiento de Robo". La complejidad mayor la encontrarán en el previaje "Waffle y bizcocho".

En esta misión aparece la noción de menor cantidad de bloques y se puede usar el bloque "mover distancia" con números negativos.

\*Para estudiantes que quieran profundizar

## > Trabajo interdisciplinario

Esta etapa se puede trabajar de forma conjunta con la asignatura Lengua, utilizando el poema de Galeano sobre la vida de Ada Lovelace, *Las edades de Ada*.

“A los dieciocho años, se fuga en brazos de su preceptor.

A los veinte, se casa, o la casan, a pesar de su notoria incompetencia para los asuntos domésticos.

A los veintiuno, se pone a estudiar, por su cuenta, lógica matemática. No son esas las labores más adecuadas para una dama, pero la familia le acepta el capricho, porque quizás así pueda entrar en razón y salvarse de la locura a la que está destinada por herencia paterna.

A los veinticinco, inventa un sistema infalible, basado en la teoría de las probabilidades, para ganar dinero en las carreras de caballos. Apuesta las joyas de la familia. Pierde todo.

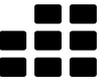
A los veintisiete, publica un trabajo revolucionario. No firma con su nombre. ¿Una obra científica firmada por una mujer?

Esa obra la convierte en la primera programadora de la historia: propone un nuevo sistema para dictar tareas a una máquina que ahorra las peores rutinas a los obreros textiles. A los treinta y cinco, cae enferma. Los médicos diagnostican histeria. Es cáncer.

En 1852, a los treinta y seis años, muere. A esa misma edad había muerto su padre, Lord Byron, poeta, a quien nunca vio. Un siglo y medio después, se llama Ada, en su homenaje, uno de los lenguajes de programación de computadoras.”

Eduardo Galeano. *Las edades de Ada*. Galeano, E. (2015).  
*Mujeres*

\*Con esta nota se busca el trabajo interdisciplinario entre distintas asignaturas.



## Etapa 3. ¿Cómo es la “trama de un laberinto”?

Tiempo estimado: 120 minutos

### Resumen de la etapa

En esta etapa se trabaja el concepto de ejecución de un algoritmo utilizando la representación de diagrama de flujo. Se evalúa la eficiencia de un algoritmo utilizando la menor cantidad de bloques como unidad de medida.

ALT+P

Duración aproximada: 80 minutos

### > Diagramas de flujo ( ) {

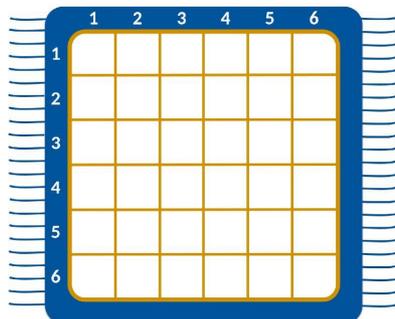
En la siguiente instancia, se retoma la definición de algoritmo de la etapa anterior y se propone una actividad como introducción a los diagramas de flujo. Para experimentar con este tipo de representación gráfica, se puede utilizar el [recurso](#) proporcionado.

## Desafío 3.1:= Tejiendo alfombras

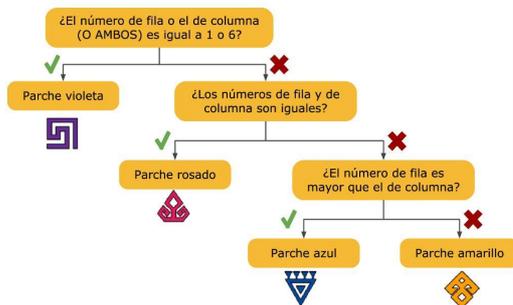


### Tejido de alfombras

Hale es un artista de tejido turco. Ella está haciendo una alfombra cuadrada con 6 filas y 6 columnas.



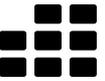
Hale pone un símbolo en cada cuadrado de la cuadrícula de la alfombra usando las siguientes preguntas:



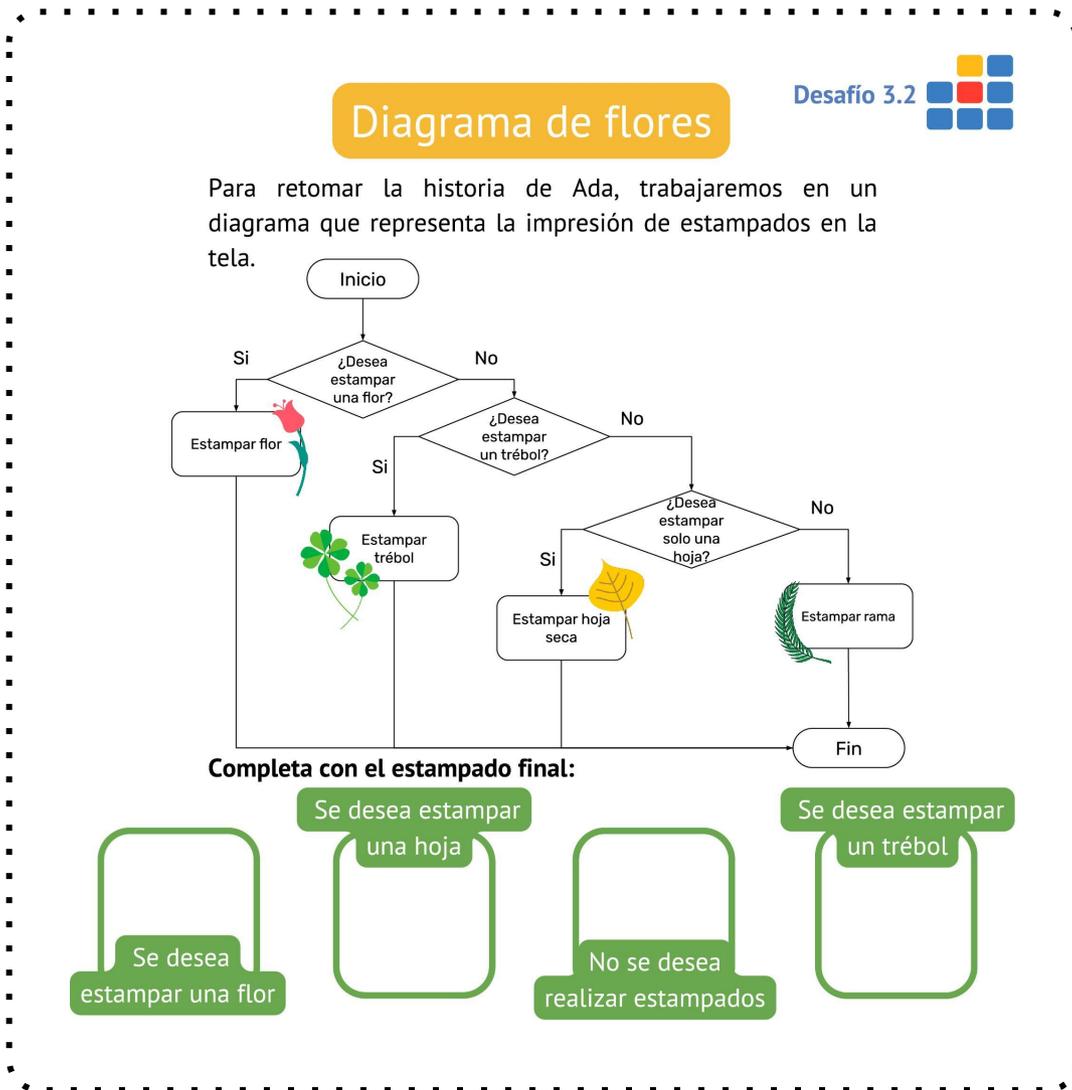
Usando este método,

¿Cómo se vería la alfombra resultante?

\* Desafío para proponer al grupo de estudiantes



Se recomienda el trabajo en duplas o equipos para resolver el problema anterior. La solución se encuentra en la sección [soluciones](#).

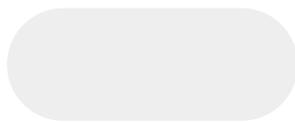


\*Desafío para proponer al grupo de estudiantes

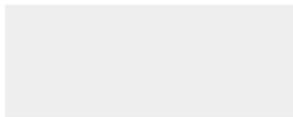
¿Cómo son las salidas del diagrama de flujo? ¿Cómo se puede modificar el diagrama para que la salida sean dos flores? ¿Por qué es necesario diferenciar los elementos de los diagramas?

\*Para reflexionar con el grupo en clase

Utilizaremos una adaptación de los diagramas de flujo para representar los algoritmos. Algunos de los elementos que los componen son los siguientes:



Inicio o fin



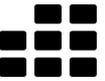
Proceso o acción



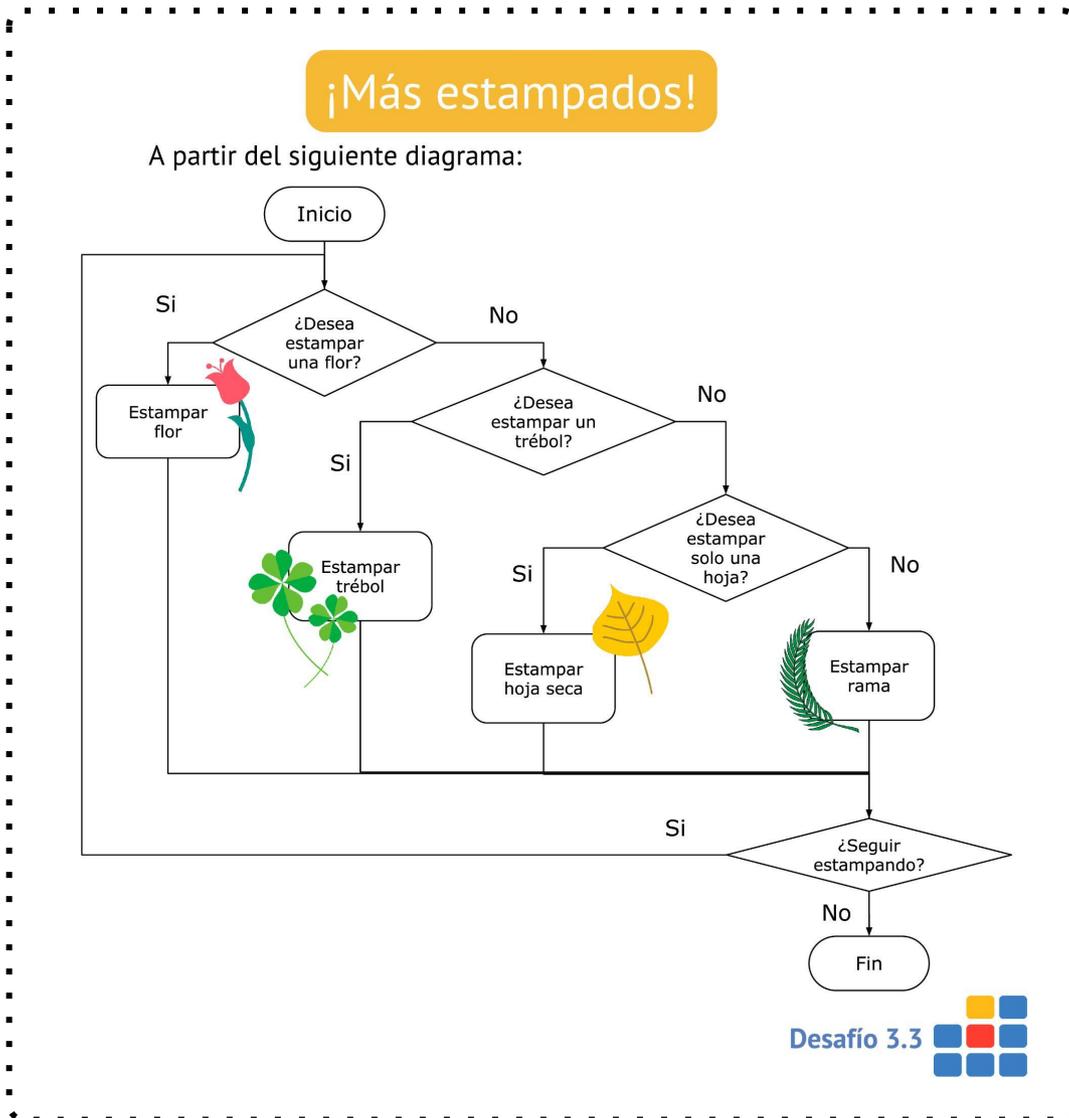
Decisión



Conectores



Finalmente, se incorpora el concepto de *bucle* dentro del diagrama de flujo.



**1** Completa con el estampado final:

Se desea estampar una flor, seguir estampando, estampar un trébol, y no seguir estampando.

Se desea estampar una hoja. Luego, estampar una rama. Por último, estampar una flor. No seguir estampando

**2** Escribe la entrada para que el patrón estampado sea el siguiente:



---

---

---

**3** Crea propio patrón para el telar utilizando el diagrama anterior.

**4** Diseña tu propio diagrama para poder tejer lo que desees.

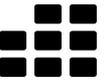


Desafío 3.3 

\*Desafío para proponer al grupo de estudiantes

La solución a la actividad se puede ver en la sección [soluciones](#).

}



ALT+T

Duración aproximada: 30 minutos

**> Menos bloques por favor ( ) {**

Se le asigna al grupo el viaje "Siente la diversión en la fiesta de cumpleaños".

El grupo de estudiantes deberán completar las aventuras 1 y 2 del viaje.

Se pide al grupo que busquen una solución con una menor cantidad de bloques.

}



Duración aproximada: 10 minutos

Para cerrar las etapas vamos a trabajar con el siguiente cuadro, donde el grupo de estudiantes deberán completar con lo aprendido.

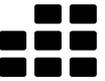
Completa las frases de acuerdo a lo aprendido en cada etapa:

De lo aprendido, le puedo explicar a alguien de la clase...

De lo aprendido, puedo explicarle a alguien que no estuvo en clase...

Aún me queda por entender...

Se encuentra en formato para imprimir en el [Anexo 1](#).



Se encuentra en formato para imprimir en el [Anexo 1](#). Se puede pedir a alguien de la clase que comparta lo que escribió para poder cerrar la etapa.

**> Preguntas claves de la etapa () {**

¿Para qué sirven los diagramas de flujo?

¿Qué actividad de la vida cotidiana podrían representar como un diagrama de flujo?

¿Podrían obtener el algoritmo de esa actividad a través del diagrama de flujo? ¿Cómo?

¿Dada una entrada, cómo verificamos la salida esperada utilizando un diagrama de flujo?

}

ALT+PC

### > **Metacognición: elementos clave**

Pensamiento computacional y programación:

- Evaluación. Para resolver los problemas en RoboGarden se pide que trabajen con la solución que utilice la menor cantidad de bloques.
- Generalización. En el desafío se generaliza el diagrama para cada celda.
- Pensamiento algorítmico. En el trabajo con los diagramas de flujo.
- RoboGarden:
  - ◆ “ver en la pantalla”. Imprime en pantalla.

\*Los conceptos y habilidades de PC que se introducen.

\*Los bloques nuevos de programación de la etapa.

### > **Profundización**

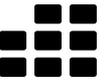
Estudiantes que quieran profundizar pueden continuar trabajando en los viajes de RoboGarden.

\*Para estudiantes que quieran profundizar

### > **Trabajo interdisciplinario**

Se puede trabajar conjuntamente con la asignatura Ciencias del ambiente (Biología) para la elección de las hojas a estampar.

\*Con esta nota se busca el trabajo interdisciplinario entre distintas asignaturas



# Etapa 4. ¿Cómo desentramar un laberinto?

Tiempo estimado: 80 minutos

## Resumen de la etapa

En esta etapa se retoma el trabajo con diagramas de flujo. Además, se comienzan a diseñar los niveles del juego. Para esto se introduce la sección "Proyectos" en RoboGarden.

ALT+P

Duración aproximada: 30 minutos

### > Luna en el laberinto () {

Se continúa con el trabajo con diagramas de flujo y laberintos a partir de la siguiente actividad. Se pide al grupo de estudiantes que resuelvan el desafío; pueden trabajar en equipos o de forma individual.

#### Desafío 4.1:= Luna en el laberinto



### Laberinto

La pequeña bruja Luna descubre una cueva con un tesoro escondido. Para llegar al tesoro, debe atravesar un laberinto. Luna descubre cinco tesoros: monedas, un rubí, un libro antiguo, un cofre y una poción mágica.

Luna no sabe qué tesoro elegir, así que sigue las siguientes reglas:

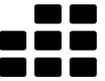
- Bajar (↓) como prioridad;
- Cuando no puede ir hacia abajo, va hacia la derecha (→);
- Cuando no puede ir hacia abajo o hacia la derecha, va hacia la izquierda (←);
- No da la vuelta y no va hacia arriba (↑) hasta que obtiene un tesoro.

¿A qué tesoro llegará Luna?



The maze is a square grid with a witch on a broom at the top entrance. The path leads to five treasure icons at the bottom, numbered 1 to 5: 1. Gold coins, 2. Red gemstone, 3. Old book, 4. Treasure chest, 5. Magic potion.

\* Desafío para proponer al grupo de estudiantes



**DESAFÍO 4.2 := Diagrama de Luna**

Diseña el diagrama de flujo que sigue Luna para recorrer el laberinto en el desafío anterior

\* Desafío para proponer al grupo de estudiantes

La solución se encuentra en la sección [soluciones](#).

}

ALT+T

Duración aproximada: 45 minutos

### > **Diseñando niveles ( ) {**

El juego final se diseñará en la sección de "Proyectos" en RoboGarden.

El grupo de estudiantes crearán un proyecto nuevo del tipo "Juego".  
En la sección de edición de niveles crearán al menos 4 niveles. Cada nivel tiene que contar con el objetivo al que debe alcanzar el robot y con obstáculos que no pueda atravesar

Para la edición de niveles es importante la grilla de la pantalla. Por defecto, el tamaño de la grilla es de 4 x 7, pero se puede modificar. El programa permite agregar y eliminar celdas. El robot, por defecto, comienza en la celda que se ubica en la esquina inferior izquierda.

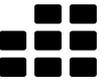
El grupo de estudiantes utilizarán los elementos de la categoría **Control** para marcar la celda objetivo a la que debe llegar el robot. Si en el nivel se desea agregar elementos que debe recoger el robot en el camino, pueden utilizarse los componentes dentro de la categoría **Coleccionable**.

Sugerimos que las paredes del laberinto se construyan con los elementos de la categoría **Daño**. Si se utilizaran los elementos de la categoría **Constructor**, se podría programar al robot para que se posicione encima de las paredes del laberinto, y la solución para evitar este problema requiere de una complejidad que excede los objetivos de esta propuesta.

Al finalizar esta etapa, el grupo de estudiantes habrán diseñado los niveles del juego final. En el [Anexo 2](#) se muestran ejemplos de niveles.

Se propone a la clase que compartan las imágenes de los productos elaborados.

}



ALT+C

Duración aproximada: 5 minutos

Para cerrar la etapa se propone otra estrategia de metacognición. En esta instancia deberán escribir tres ideas principales que aprendieron, dos preguntas que les quedan pendientes y una imagen que represente lo trabajado. En [Anexo 1](#) se brinda el recurso para imprimir, si se desea trabajar de esta forma.

3 ideas



2 preguntas



1 imagen



Una posible forma de completar el cuadro anterior es la siguiente:

### 3 ideas

1. En RoboGarden puedo crear juegos.
2. Las instrucciones que sigue Luna para resolver el laberinto son parecidas a las de Robo.
3. Vamos a crear un juego que tenga algunos niveles, y esos niveles son laberintos.



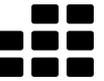
### 2 preguntas

1. ¿Cómo programo el juego?
2. ¿Qué relación hay entre los diagramas de flujo y el juego?



### 1 imagen





**> Preguntas claves de la etapa ( ) {**

¿Qué elementos de RoboGarden nos permiten indicar el tesoro del laberinto?  
¿Para qué son útiles los diagramas de flujo en la búsqueda de solución de laberintos?

}

ALT+PC

### > **Metacognición: elementos clave () {**

Pensamiento computacional y programación:

→Generalización. Al crear un nivel, los conocimientos se generalizan para los restantes.

→Proyectos en RoboGarden

- ◆ Constructor.
- ◆ Control.
- ◆ Coleccionable.
- ◆ Daño.

\*Los conceptos y habilidades de PC que se introducen.

\*Los bloques nuevos de programación de la etapa.

### > **Profundización () {**

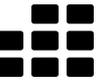
Estudiantes que deseen profundizar pueden complejizar el diseño de niveles, ya sea por la cantidad de niveles o por los objetos.

\*Para estudiantes que quieran profundizar.

### > **Trabajo interdisciplinario () {**

El diseño de los objetos, así como los laberintos, puede ser trabajado en coordinación con la asignatura Comunicación visual.

\*Con esta nota se busca el trabajo interdisciplinario entre distintas asignaturas.



# Etapa 5. Un laberinto, ¿muchos caminos?

Tiempo estimado: 120 minutos

## Resumen de la etapa

En esta etapa se trabajan algoritmos conocidos para encontrar la solución a distintos laberintos. Además, se presenta el concepto de *camino más corto*.

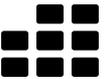
ALT+P

Duración aproximada: 30 minutos

> **Algoritmos con nombre () {**

¿Qué método podría seguir el robot para recorrer el laberinto y encontrar el tesoro, aún sin saber dónde está? ¿Podría seguir siempre las mismas instrucciones? ¿Cuáles? ¿Conocen algún algoritmo para resolver laberintos?

\*Para reflexionar con el grupo en clase



### **DESAFÍO 5.1 := ¿Algoritmos para recorrer laberintos?**

Observa los métodos que se describen a continuación y pruébalos en los laberintos creados en la etapa anterior.

#### **CHARLES PIERRE TRÉMAUX**

- Cuando siga un camino, marque la entrada una vez.
- Nunca siga un camino que tenga dos marcas.
- Si llega a un cruce que no tiene marcas (excepto el que está en el camino por el que entró), elija un camino sin marcas, sígalo y márkelo.
- En caso contrario:
  - Si el camino por el que entró tiene solo una marca, dé la vuelta y regrese por donde entró, marcándolo nuevamente. Este caso se da al llegar a un callejón sin salida.
  - De lo contrario, elija una de las rutas restantes que tenga la menor cantidad de marcas (cero si es posible, de lo contrario una), siga esa ruta y márkela.

#### **GASTON TARRY (1985)**

Para este método se supone que todos los pasillos comienzan y terminan en un cruce, y un cruce es donde se encuentran más de un pasillo. Según Tarry, al recorrer el laberinto:

- Cada vez que entramos a un pasillo dejamos dos marcas en su entrada.
- Cada vez que salimos de un pasillo, al llegar a un cruce dejamos una marca. De esta forma, si ya había recorrido el pasillo una vez, quedarán tres marcas al salir de él.
- Cuando estemos en un cruce, elegiremos los pasillos que están inexplorados o que solo tengan una marca.

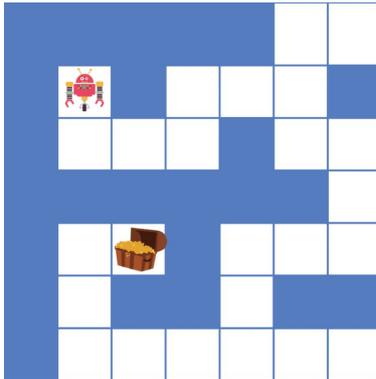
#### **TOMAR EL CAMINO DE LA DERECHA**

Finalmente, un método más sencillo es avanzar utilizando como guía la pared derecha del laberinto.

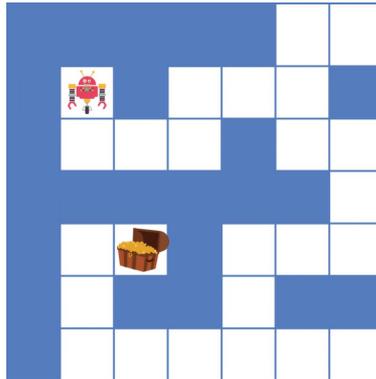
\*Desafío para proponer al grupo de estudiantes

Ejemplos de aplicación de los distintos métodos:

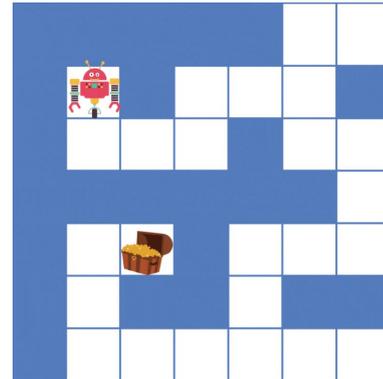
Trémaux ([VER GIF](#))



Tarry ([VER GIF](#))



Camino de la derecha ([VER GIF](#))



Se motiva a que el grupo de estudiantes comparta métodos distintos en caso de haber encontrado alguno. La exploración y profundización de estos métodos puede enriquecer el trabajo en el proyecto.

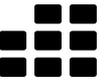
**DESAFÍO 5.2 := Diagrama de flujo y algoritmos**

Escribe el diagrama de flujo del último algoritmo explorado en el desafío anterior.

\*Desafío para proponer al grupo de estudiantes

Una posible solución al desafío se muestra en la sección [soluciones](#).

}



ALT+T

Duración aproximada: 80 minutos

### > **Caminos más cortos ( ) {**

Finalmente, se evalúa el camino más corto para encontrar un tesoro en un laberinto.

#### **DESAFÍO 5.3 := Encuentra TODAS las soluciones**

A partir de los niveles en RoboGarden que se diseñaron en la etapa anterior, indica todos los caminos posibles que debe seguir el robot para encontrar el tesoro.

Dibuja la trayectoria en el laberinto y anota la cantidad de pasos que debe dar el robot para llegar al objetivo. Los pasos se representan con la cantidad de celdas que transita.

\*Desafío para proponer al grupo de estudiantes

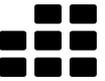
Se hace énfasis en que de todas esas posibles soluciones hay una que es la más corta y la podemos etiquetar como la solución más eficiente al laberinto. Un algoritmo se considera eficiente *si utiliza la menor cantidad de recursos posibles para resolver el problema en cuestión.*

**DESAFÍO 5.4 := ¡Encontrar el camino más corto!**

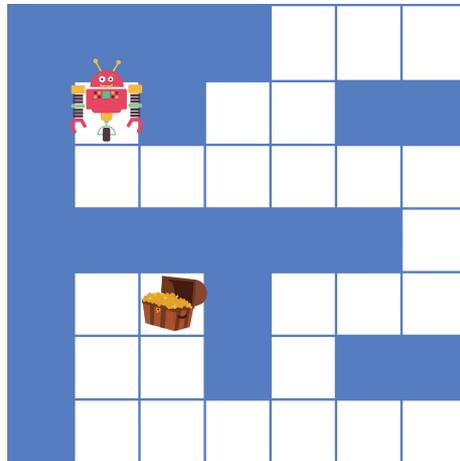
Encuentra los caminos más cortos de los niveles creados en la etapa anterior, mediante el siguiente procedimiento para calcular la distancia utilizando la celda como unidad:

1. Comenzamos desde el tesoro. Esta celda está a una distancia de 0 movimientos del final, por lo tanto se marcará con un 0.
2. Marcamos con un 1 las celdas adyacentes al tesoro (al lado), ya que están a una distancia de 1 movimiento del tesoro.
3. Ahora, marcamos con un 2 las celdas adyacentes a las celdas marcadas con 1, ya que están a una distancia de 2 movimientos del tesoro.
4. Continuamos con este proceso hasta llegar al robot.
5. El número de la celda con el que se marca al robot indica la cantidad de movimientos que este debe realizar para llegar al tesoro. En el caso siguiente, si el robot está en una celda marcada con 16, significa que debe realizar 16 movimientos para llegar al tesoro.

\*Desafío para proponer al grupo de estudiantes



Se muestra un ejemplo de aplicación del algoritmo anterior ([VER GIF](#)):



}

> **Continúa el viaje por RoboGarden () {**

Los estudiantes deberán completar las aventuras 3 y 4 del viaje "Siente la diversión en la fiesta de cumpleaños".

}

ALT+C

Duración aproximada: 10 minutos

En esta instancia deberán escribir tres ideas principales que aprendieron, dos preguntas que les quedan pendientes y una imagen que represente lo trabajado.

En [Anexo 1](#) se brinda el recurso para imprimir si se desea trabajar de esta forma.

3 ideas

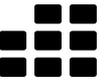


2 preguntas



1 imagen





**> Preguntas claves de la etapa () {**

¿Cuándo una solución es más eficiente que otra?  
¿Qué diferencia hay entre los bloques de programación “pintar” y “color en la celda”? ¿Qué sucede cuando utilizamos estos dos bloques consecutivos en un programa? ¿Cambia la salida según el orden de los bloques?  
¿Qué relación hay entre el bloque “mover distancia” y los bloques “dar un paso adelante” y “dar un paso atrás”?

}

ALT+PC

### > **Metacognición: elementos clave**

Pensamiento computacional y programación:

- Evaluación. Al evaluar las distintas soluciones a los laberintos.
- Pensamiento algorítmico. Para interpretar los distintos procedimientos de la etapa.
- RoboGarden:
  - ◆ Mover distancia (admite números positivos y negativos)
  - ◆ Pintar color en la celda
  - ◆ Soltar el objeto
  - ◆ Obtener el color de la celda

\*Los conceptos y habilidades de PC que se introducen.

\*Los bloques nuevos de programación de la etapa.

### > **Profundización**

Para estudiantes que se hayan interesado con la etapa es posible profundizar en los distintos métodos para resolver algoritmos.

\*Para estudiantes que quieran profundizar.



# Etapa 6. Creación de un juego de laberintos

Tiempo estimado: 160 minutos

## Resumen de la etapa

En esta etapa se crean las reglas del juego en RoboGarden. Para esto, previamente se trabaja el concepto de *eventos*.

ALT+P

Duración aproximada: 30 minutos

### > **Simón dice () {**

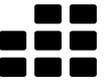
Para introducir el concepto de *evento*, se le propone a la clase la siguiente dinámica: el grupo se sienta en ronda (se puede seleccionar un subgrupo de estudiantes para que lleven adelante la dinámica); cada docente será quien modera, es decir, según lo que diga o haga el grupo estudiantes deberán actuar.

### **REGLAS DEL JUEGO**

Cuando la persona que modera:

- levante la mano derecha, cada estudiante tiene que moverse un asiento a la derecha.
- levante la mano izquierda, cada estudiante tiene que moverse un asiento a la izquierda.
- levante las dos manos, cada estudiante se tiene que parar.

\* Variantes: se puede cambiar la persona que indique las reglas. También se puede cambiar los eventos disparadores, por ejemplo, cuando aplauda o cuando se siente.



Luego de jugarlo varias veces, se plantean las siguientes preguntas:

¿Cuándo se mueven a la derecha el grupo de estudiantes? ¿Y a la izquierda? ¿Se mueven a la derecha con otra instrucción? Si quien modera indica dos instrucciones en simultáneo (por ejemplo, aplaudir y sentarse), ¿qué sucede?

\*Para reflexionar con el grupo en clase

A partir de las preguntas anteriores, se introduce el concepto de evento. *Los eventos son acciones que habilitan o desencadenan la ejecución de un programa.* En RoboGarden un evento que se utiliza es presionar el botón "Ejecutar", al presionarlo se ejecuta el programa y cada bloque le indica al robot qué hacer.

}

## > Descubriendo eventos ( ) {

Se reflexiona con el grupo acerca de las acciones que podemos hacer en la computadora y los comportamientos que se desencadenan.

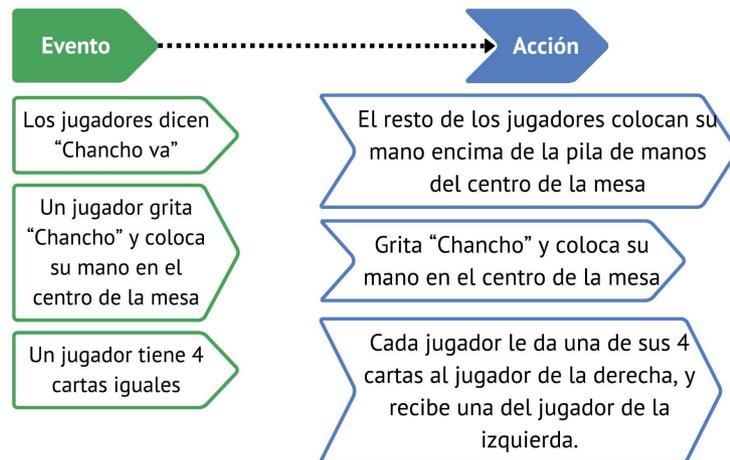
\*Para reflexionar con el grupo en clase

Ejemplos: clic sobre un botón, doble clic sobre un botón, arrastrar íconos, presionar teclas. Incluso pueden utilizarse otras entradas, como la cámara o el sonido.

### DESAFÍO 6.1 := ¿Qué desencadena cada evento?

#### ¿Qué desencadena cada evento?

Para este desafío utilizaremos el ejemplo del juego "Chanco va". Teniendo en cuenta las reglas del juego, **une** cada evento con la acción que desencadena



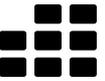
Imagina otros eventos de la vida cotidiana que desencadenan acciones.



Desafío 6.1

\*Desafío para proponer al grupo de estudiantes

}



ALT+T

Duración aproximada: 120 minutos

### > **Un paso a la vez () {**

Se programan las reglas del juego final. El programa que deben crear el grupo de estudiantes indica las instrucciones que sigue el robot.  
Para esto trabajaremos en la sección de proyectos, en la pestaña "Editor de lógica".

#### **DESAFÍO 6.2 := Moviendo el robot**

Retoma el proyecto con los niveles del juego en RoboGarden y programa al robot para que se mueva según las indicaciones del usuario.

Sugerencia: Prueba el bloque "when joystick is right pressed".

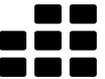
\*Desafío para proponer al grupo de estudiantes

Para programar los movimientos del robot se utilizan los eventos "when joystick is right/left/up/down pressed". Dentro de este evento, se puede colocar el bloque correspondiente "slide right/left" o "step forward/backward".

En esta instancia, la dificultad se centra en incorporar los eventos al programa, por lo tanto, se propone al grupo que prueben el juego con estos eventos solamente.



}



## > Esquivando y recolectando () {

### **DESAFÍO 6.3 := ¡Ojo con los obstáculos!**

Modifica el programa anterior, de forma que la persona usuaria pierda si el robot colisiona con un obstáculo, y gane puntos si recolecta objetos deseables.

Sugerencia: Prueba los bloques de eventos “when Robo collects” y “when Robo collides with”.

\*Desafío para proponer al grupo de estudiantes

Se utilizan bloques de la categoría Jugabilidad para que cada jugador sume puntos, avance de nivel, pierda o gane. Para indicar que el robot pierde el juego al tocar una pared, se utiliza el bloque “game over”. Por otro lado, para indicar que el robot alcanza el objeto final y debe avanzar de pantalla, se utiliza el bloque “go to next level”.

Si el grupo de estudiantes desean incluir monedas que el robot debe recolectar, pueden utilizar el bloque “change player score by” cuando el robot las encuentre. Asimismo, si se desea eliminar los elementos que se recolectan, se puede utilizar el bloque “remove object”.

}

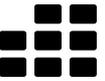
ALT+C

Duración aproximada: 10 minutos

**> Preguntas claves de la etapa ( ) {**

¿Qué bloques nuevos se introdujeron en la etapa?  
¿Lo aprendido en los viajes realizados en RoboGarden en las etapas anteriores resultó de utilidad al momento de programar un juego?  
¿Qué actividades de la vida cotidiana se relacionan con eventos?

**}**



ALT+PC

### > Metacognición: elementos clave

Pensamiento computacional y programación:

- Evaluación. Se depura el programa en cada instancia probando las distintas versiones del juego.
- Pensamiento algorítmico. Se programan las reglas del juego.
- RoboGarden:
  - ◆ When joystick is right/left/up/down pressed
  - ◆ When Robo collects
  - ◆ When Robo collides with
  - ◆ Change player score by
  - ◆ Go to next level
  - ◆ Go to x \_ y \_
  - ◆ Remove object

\*Los conceptos y habilidades de PC que se introducen.

\*Los bloques nuevos de programación de la etapa.

## > **Profundización**

Para aumentar el nivel de complejidad del juego se puede proponer el siguiente desafío:

### **DESAFÍO 6.4 := Avanzar en los niveles**

Modifica el programa anterior, de forma que cada usuario avance a través de los niveles cuando llegue al objetivo en cada instancia.

Prueba los bloques dentro de la categoría Gameplay.

\*Desafío para proponer al grupo de estudiantes

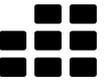
Como último nivel de complejidad del juego, se puede programar que, cuando el nivel actual sea el máximo de niveles diseñado, se gane el juego. Para esta opción se utilizan los bloques "if then else", "get current level" y "game complete", que se encuentran en la categoría Gameplay. Un posible programa se encuentra en el [Anexo 3](#).

Recuerden que, cuando se avanza al siguiente nivel, el robot se encuentra en la celda donde finalizó el nivel anterior. Por lo tanto, se puede utilizar el bloque "go to x \_ y \_" para ubicar al robot en la celda que se quiera.

## > **Trabajo interdisciplinario**

Los bloques en esta instancia no están en español, lo que permite el trabajo en conjunto con la asignatura Inglés.

\*Con esta nota se busca el trabajo interdisciplinario entre distintas asignaturas.



# Etapa 7. Socialización de los juegos

Tiempo estimado: 80 minutos

## Resumen de la etapa

Finalmente, para dar por finalizado el proyecto, el grupo de estudiantes comparten con la clase los juegos creados. Se utiliza la sección "Proyectos" de RoboGarden y, si se desea, se pueden descargar los juegos y utilizarlos en los celulares.

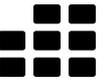
ALT+P

Duración aproximada: 40 minutos

**> Evaluación () {**

Se realiza una evaluación formativa, dentro de la plataforma SEA, sobre los conceptos aprendidos en el proyecto.

**}**



ALT+T

> Duración aproximada: 40 minutos

> **Salida a producción () {**

El grupo de estudiantes comparten sus juegos con la comunidad en RoboGarden, para esto es necesario que queden públicos en la plataforma.. En caso de contar con dispositivos con sistema operativo Android, pueden descargarlo y probarlo.

}

> **Presentación en muchos formatos () {**

Se sugiere la siguiente actividad para trabajar con la asignatura Lengua Española (texto argumentativo) y Arte - Comunicación Visual, Artes Visuales y Plásticas.

### **DESAFÍO 7.1 := Publicidad**

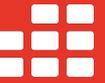
En grupos pequeños, crea un afiche, folleto o audiovisual promocionando el juego. Estas preguntas pueden ayudar en el diseño:

- ¿Cuál es el objetivo de esta publicidad? ¿Qué es lo que quieren comunicar?
- ¿A quiénes va dirigido?
- ¿Qué imágenes incorporarán? (Es muy importante, ya que impacta en la persona receptora y puede quedar bastante tiempo en la memoria de las personas).
- ¿Qué tipografías usarán? (Recordar que deben ser claras y legibles a distancia).
- ¿Qué datos incluirán?

Comparte el producto con el resto de los equipos.

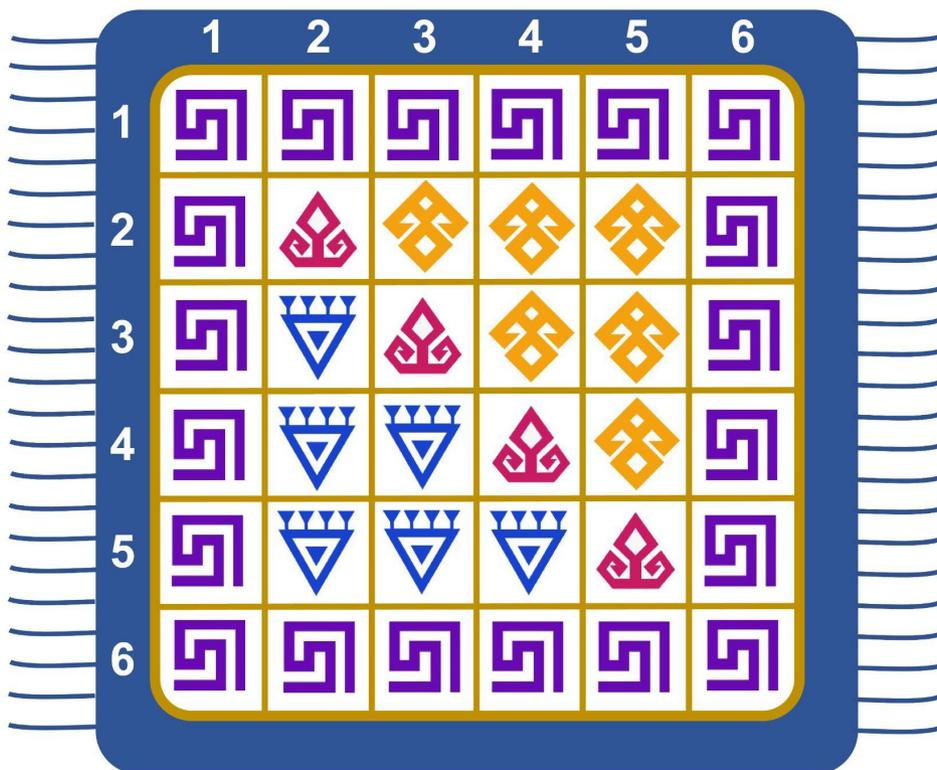
\*Desafío para proponer al grupo de estudiantes

}



# Soluciones a los desafíos

### La solución al desafío 3.1



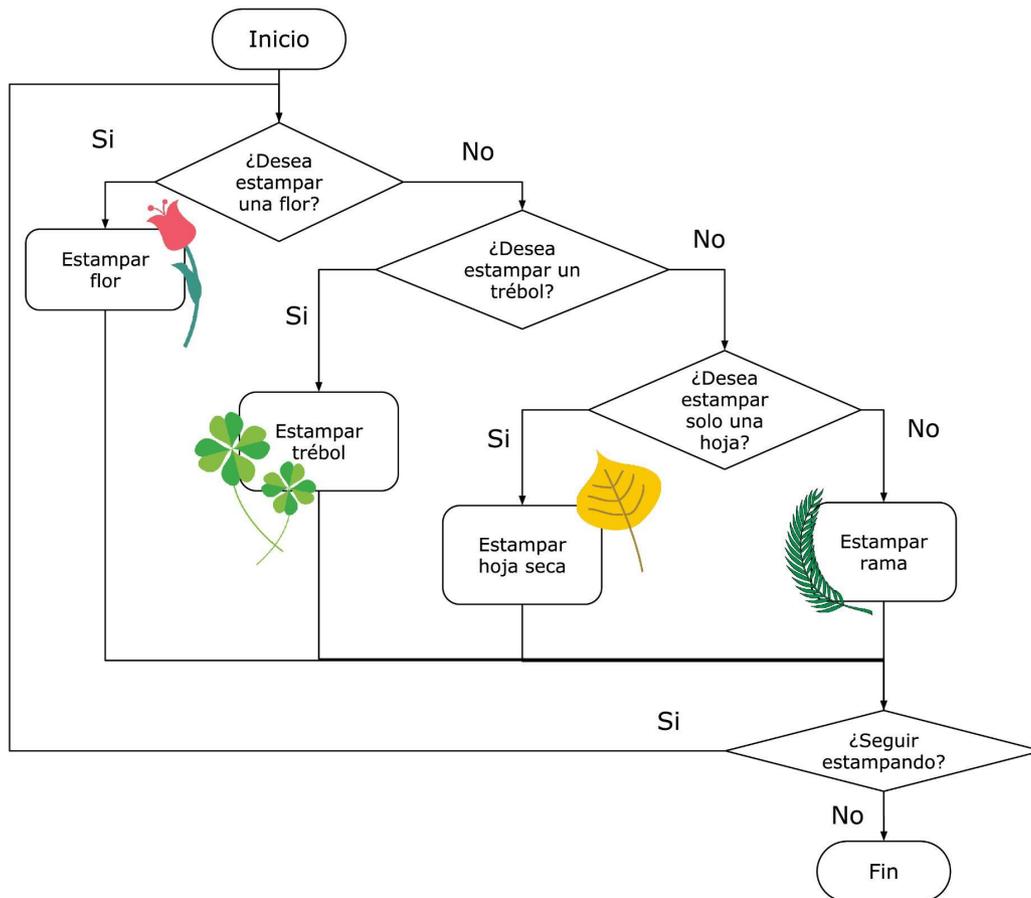


## Una posible respuesta al desafío 3.3

Observa el siguiente diagrama de flujo

¡Más estampados!

A partir del siguiente diagrama:



Desafío 3.3

**1** Completa con el estampado final:

Se desea estampar una flor, seguir estampando, estampar un trébol, y no seguir estampando.



Se desea estampar una hoja. Luego, estampar una rama. Por último, estampar una flor. No seguir estampando



**2** Escribe la entrada para que el patrón estampado sea el siguiente:



Estampar una hoja, estampar un trébol, estampar una rama, estampar una flor, estampar un trébol, estampar una hoja, estampar un rama, y no seguir estampando

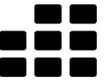
**3** Crea propio patrón para el telar utilizando el diagrama anterior.



**4** Diseña tu propio diagrama para poder tejer lo que desees.



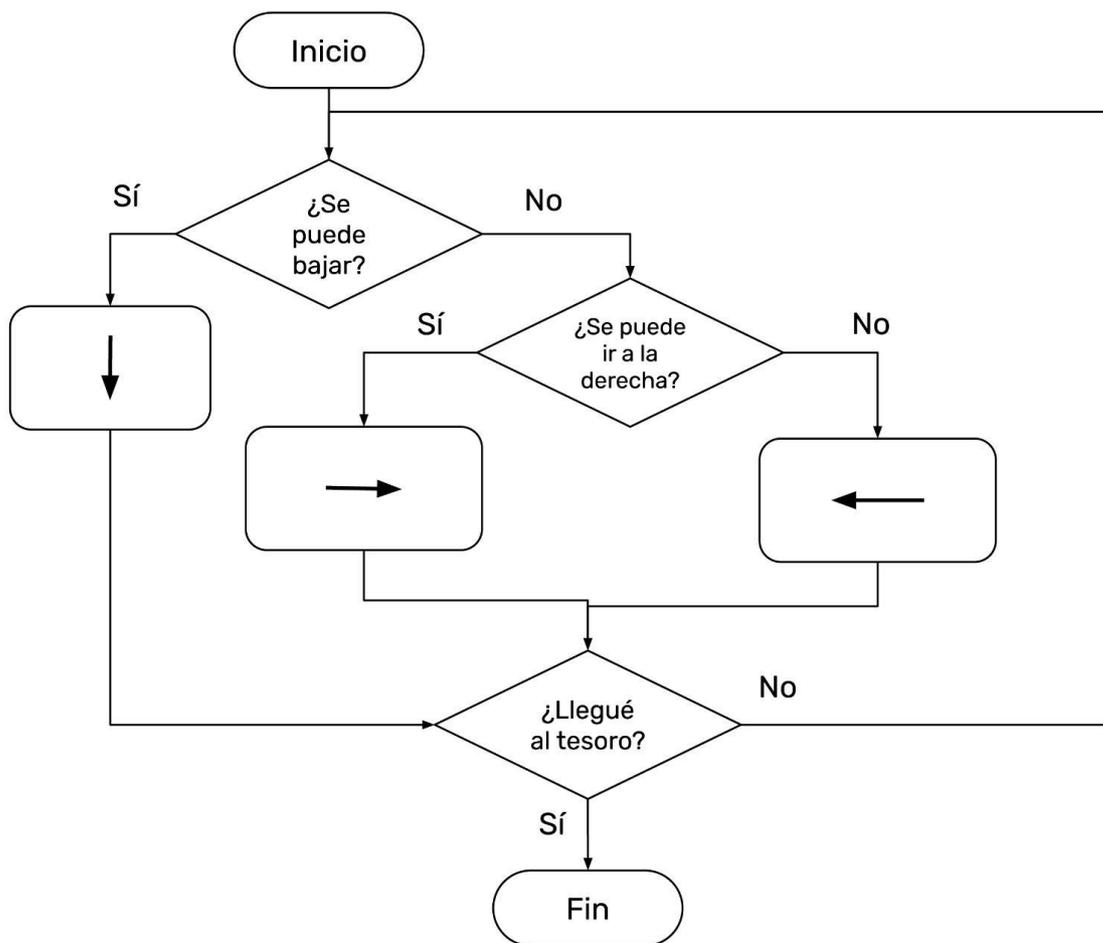
Desafío 3.3 



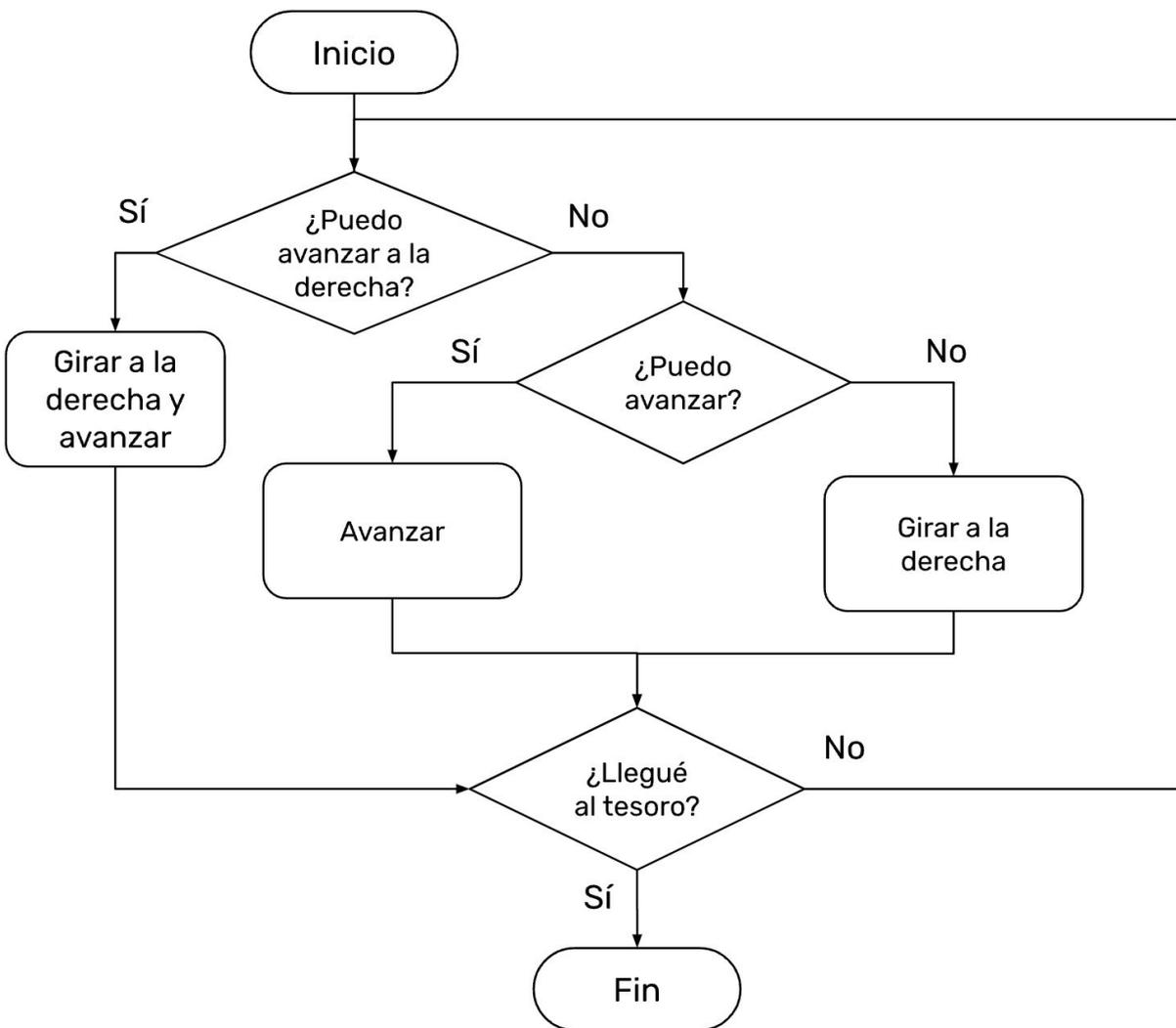
### Solución al desafío 4.1

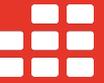
La solución al desafío es el número 4.

### Solución al desafío 4.2



## Una posible solución al desafío 5.2.





# Anexos

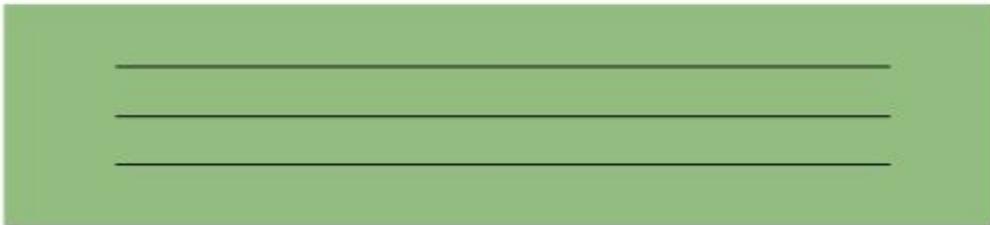
## Anexo 1

Estrategias para la metacognición:

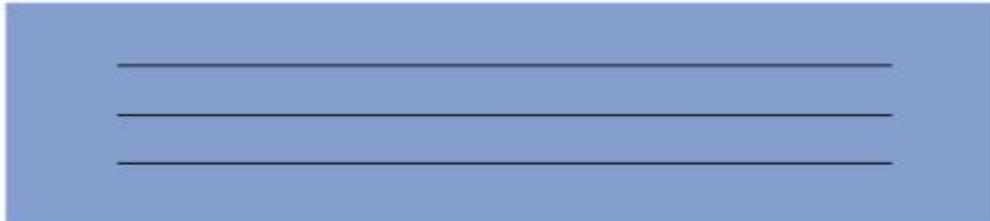
Cierre

Completa las frases de acuerdo a lo aprendido en la etapa:

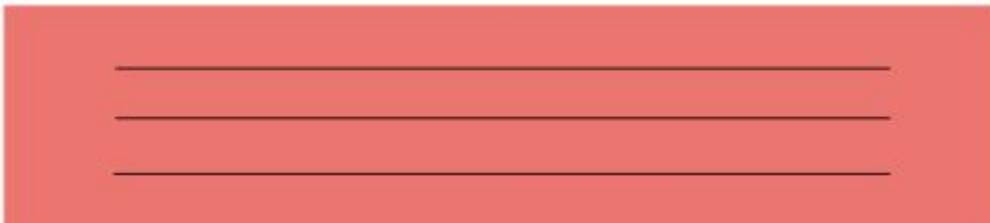
**De lo aprendido, le puedo explicar a alguien más de la clase...**

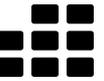


**De lo aprendido, puedo explicarle a alguien que no estuvo en clase...**



**Aún me queda por entender...**





3 ideas



2 preguntas

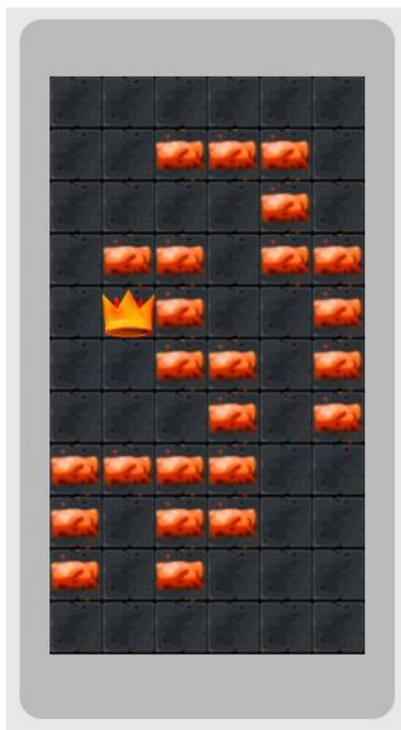
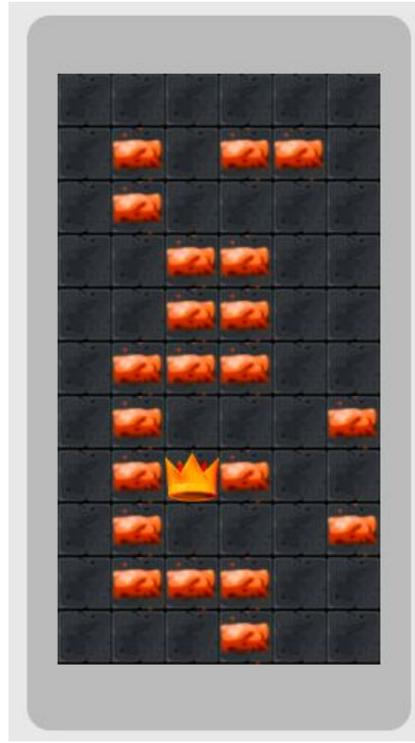
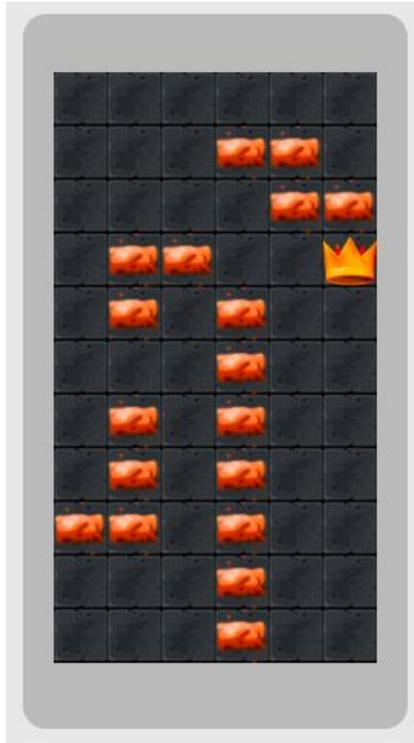


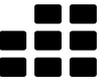
1 imagen



## Anexo 2

Ejemplos de niveles diseñados en RoboGarden:





### Anexo 3

Ejemplos de programación del juego en RoboGarden. En este caso se utiliza el joystick para mover al robot. Y se diseñaron 3 niveles, por lo tanto, cuando el usuario llega al 3 nivel se gana el juego.

