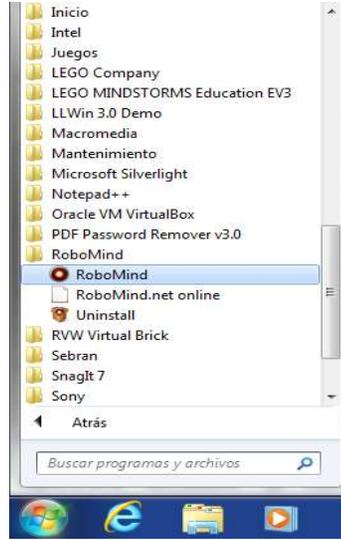


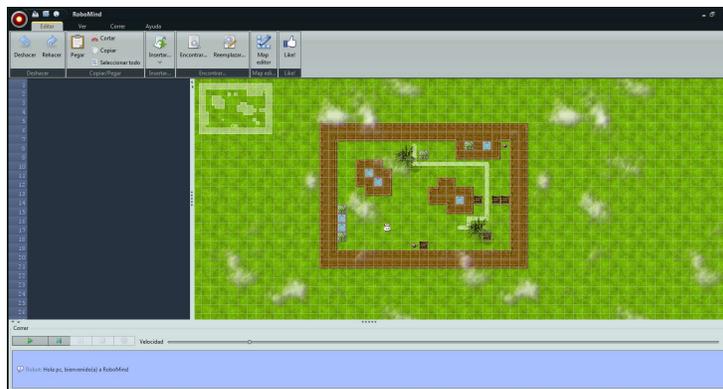
Antes de iniciar con la descripción del entorno de la aplicación, te compartiré una de las opciones que tienes para iniciarla, este proceso se describe tomando como referencia el sistema operativo Windows 7.

1. Da clic en el botón “Iniciar”,
2. Después selecciona “Todos los programas”
3. Busca la carpeta “RoboMind”
4. Da clic en “RoboMind”, puedes identificarlo con el siguiente icono .
5. Ahora solo espera que la aplicación inicie.



para

En la siguiente pantalla se muestra el entorno de la aplicación RoboMind.



En la parte superior izquierda de la aplicación, encuentras las herramientas que te permiten guardar un archivo, crear uno nuevo, editar, ver, entre otras. Algunas de estas opciones son similares a las que te ofrecen otras aplicaciones como Word, PowerPoint.

Conocerás algunas de las herramientas que te ofrece RoboMind.



En la parte superior encuentras tres íconos de acceso rápido .



Te permite guardar el archivo activo.



Abrir un archivo almacenado o guardado en tu equipo.



Abrir un mapa. Los mapas te sirven para resolver retos o “Challenges”, es decir probar tus habilidades en la creación de programas para robots.

Botón principal

Si das clic en el ícono de la aplicación , tendrás disponibles las siguientes opciones:



- Nuevo. Crear un nuevo archivo.
- Abrir. Abrir un archivo existente.
- Guardar. Guardar el archivo activo.
- Guardar como. Guardar el archivo activo con otro nombre.
- Print. Imprimir el archivo activo.
- Exportar. En algunos casos es posible exportar el código generado a un robot físico como EV3 o NTX (LEGO).
- Traducir guion. Esta opción sirve para cambiar el lenguaje en el que están inscritas las instrucciones de tu programa.
- Abrir mapa. Abrir un mapa para que crees un programa y des solución al reto que este plantea. También puedes crear tus propios mapas.
- Guardar mapa como. Guardar el mapa activo para una edición o modificación posterior.
- Configuración. Configurar el entorno de la aplicación.
- Salir. Cerrar el archivo activo y/o salir de la aplicación.

Configuración de la aplicación

La opción “Configuración” te permite personalizar algunos aspectos de RoboMind, como el idioma o la apariencia de tu robot.

Si deseas personalizarla, da clic en el botón principal, da clic en “Configuración”, para activar la ventana de configuración.

La ventana de configuración te ofrece cinco opciones: General, Ver, Editor, Sound y License.



General.

En la pestaña “General” puedes configurar el lenguaje. Si por alguna instalaste la aplicación en otro idioma, en “Lenguaje” y selecciona el idioma al deseas cambiar la aplicación.

Definición del guion. Esta opción te permite cambiar el idioma en el que se presentan las instrucciones utilizadas generar un programa. De igual manera, para cambiar el idioma, da clic opción para cambiar el idioma.



razón
da clic
que

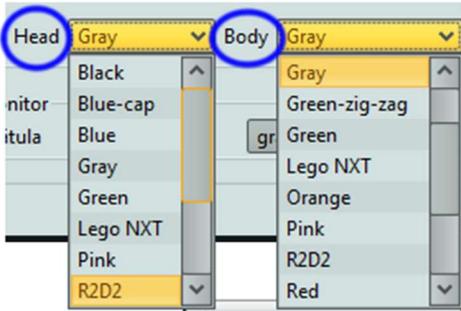
al
en esta

Ver

La opción “Ver” te permite configurar el robot y el escenario donde se moverá tu robot.

Puedes cambiar el “Head” o cabeza y el “Body” cuerpo de tu robot.

Por default, tu robot está configurado con la opción “Gray, tanto para la cabeza como para el cuerpo.



La imagen de la izquierda te muestra las diferentes opciones que tienes para configurarlo, prueba diferentes opciones y combinaciones para tu robot. En mi caso seleccionaré las opciones

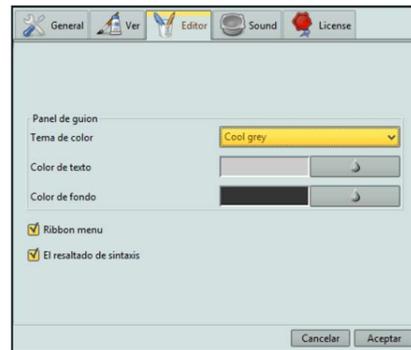
“R2D2” para la cabeza y el cuerpo.

El resultado de mi configuración se muestra en la imagen de la derecha.



Editor

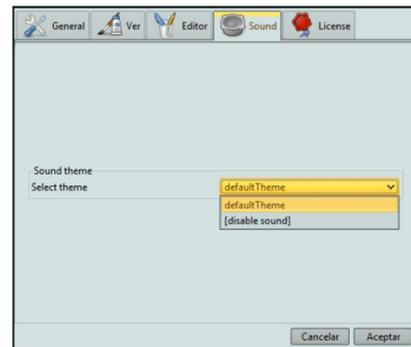
En la pestaña del editor, te permite cambiar los colores del entorno de la aplicación, como el color del tema, ofrece algunas combinaciones ya establecidas. También puedes modificar solo el color del texto o el color del fondo. Prueba diferentes opciones de combinación de colores para personalizar tu aplicación.



colores que te color

Sound (sonido)

La pestaña de sonido solamente ofrece dos opciones, por default o desactivar el sonido. Prueba y escoge la que desees para esta opción.

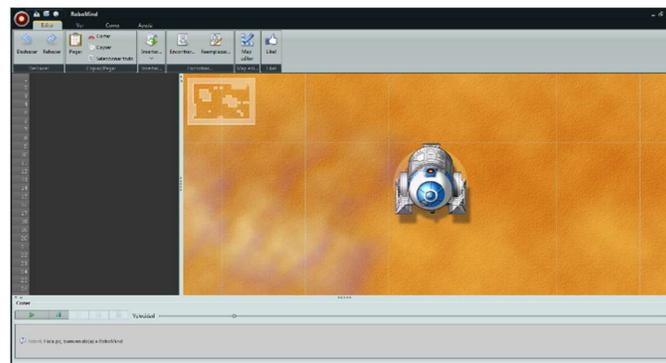


el tema opción

Si realizaste algunos cambios, presiona el botón “Aceptar”. Es posible que te pida reiniciar la aplicación para realizar los cambios en la configuración. Presiona el botón “Aceptar”.

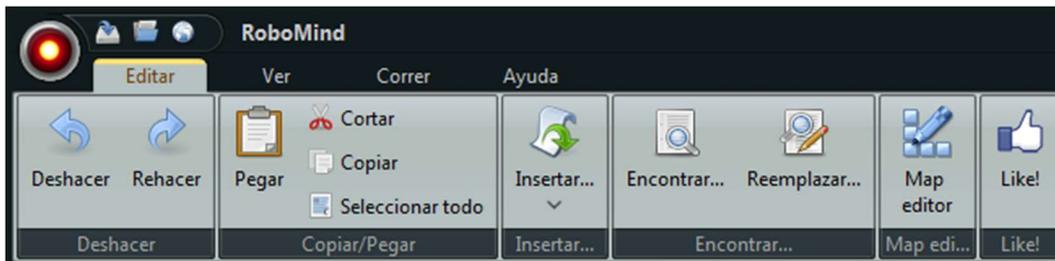


En la imagen siguiente puedes ver los cambios una vez que se aplicó la configuración.



Pestañas de la aplicación RoboMind

RoboMind cuenta con cuatro pestañas (menú) que te permitirán interactuar con la aplicación, así como generar código que te permita manipular tu robot.



Las pestañas son: Editar, Ver, Correr y Ayuda.

Editar

En esta pestaña puedes encontrar seis grupos de comandos. Estos grupos de comandos, como ya se ha mencionado son similares en su funcionamiento al de otras aplicaciones, los grupos de comandos son:

Deshacer

Este grupo de comandos cuenta con dos comandos, Deshacer y Rehacer.

Copiar/Pegar

En este grupo de comandos encontrarás los comandos: Pegar, Cortar, Copiar, Seleccionar todo.

Insertar

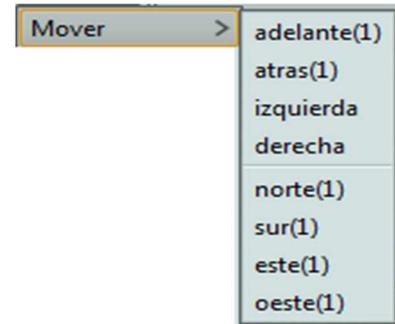
Quizá te preguntarás en qué momento empezarás a programar y que herramientas utilizaras para ello. En este grupo de comandos, al igual que en otras aplicaciones puedes insertar algunos elementos u objetos, los elementos que insertas aquí son referentes al código de programa, los cuales te servirán para crear programas para tu robot.

Estas instrucciones o códigos de programa se encuentran divididos en categorías.



Mover

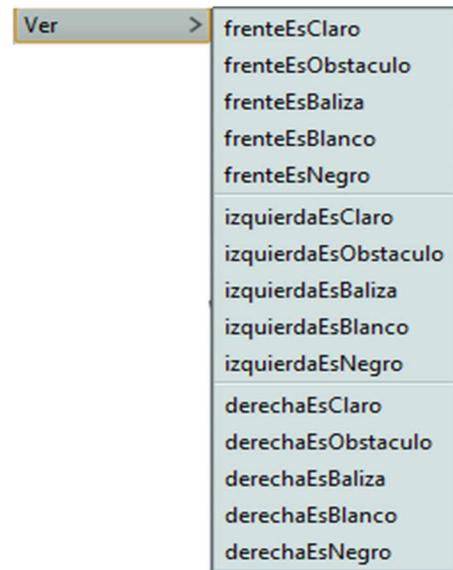
En esta categoría, encontrarás los comandos que harán que tu robot se mueva en diferentes direcciones, ya sea hacia *adelante*, *atrás*, *izquierda*, *derecha*, *norte*, *sur*, *este* u *oeste*.



Ver

Esta categoría te permite dar cierta autonomía al robot, ya que después de verificar algunas de las situaciones que incluye, puedes indicarle que realice alguna acción.

Dentro de los comandos que incluye esta categoría están: *frenteEsClaro*, *frenteEsObstáculo*, *frenteEsBaliza*, etc. Su interpretación no es nada complicada, aquí tendrías que preguntar si cumple alguna de las condiciones que realice una acción, claro, esto mediante la utilización combinada de estos comandos.



Pintar

En esta categoría, los comandos que incluye *pintarBlanco*, *pintarNegro* y *detenerPintar*.

Al indicarle que realice algunas de estas acciones, a través de estos comandos, pintará

una línea blanca o negra, mientras se mueve en cierta dirección. Si deseas indicarle que deje de pintar deberás utilizar el comando *detenerPintar*.

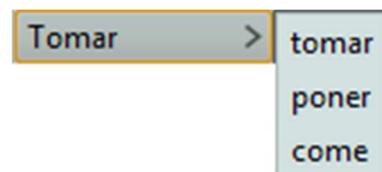


son

una

Tomar

Esta categoría incluye tres instrucciones o comandos, *poner* y *comer*.

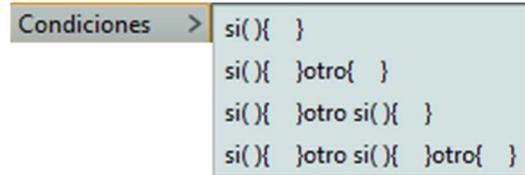


tomar,

El primero de ellos indica al robot que tome algún objeto que encuentre, el segundo que ponga en un lugar determinado un objeto que ha recogido anteriormente, también puedes indicarle que coma algún objeto que encuentre.

Condiciones

Se utilizan estas expresiones para determinar si un resultado o valor es verdadero o falso, que a su vez determinará la ejecución de ciertas acciones.

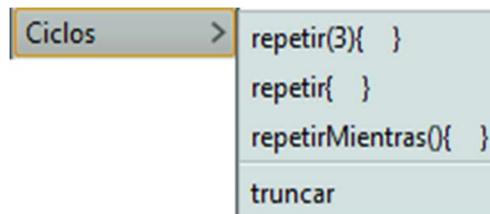


Las instrucciones que existen en esta categoría son:

- `si() { }`
- `si() { } otro{ }`
- `si() { } otro si() { }`
- `si() { } otro si() { } otro{ }`

Ciclos

Los ciclos te permiten repetir una o varias instrucciones cierta cantidad de veces o mientras se cumpla alguna condición.



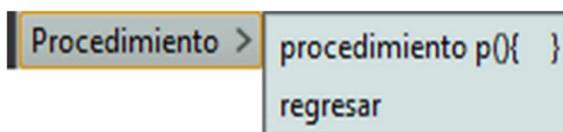
Los comandos que aquí se incluyen son:

repetir(3){}, repetir{ }, repetirMientras(){ } y *truncar*.

Procedimiento

¿Qué es un procedimiento? De una manera breve podemos decir que los procedimientos son sub programas dentro de un programa, los cuales puedes crear cuando necesitas usar una secuencia de instrucciones en más de una ocasión. Esto te permite, entre otras cosas, generar menos líneas de programa.

Los comandos o instrucciones que incluye esta categoría son: *procedimiento p(){ }* y *regresar*.



esta

lanzarMoneda

Cuando utilizas la instrucción *lanzarMoneda* te permite tomar una **lanzarMoneda** decisión al azar, y con base a ese resultado realizar otras acciones. El valor que devuelve o calcula lo podemos definir como falso o verdadero, o como el tradicional juego de lanzar la moneda al aire, cara o sol, dependiendo del resultado realizamos algo.

fin

Este comando permite poner fin a la ejecución de un programa. **fin**

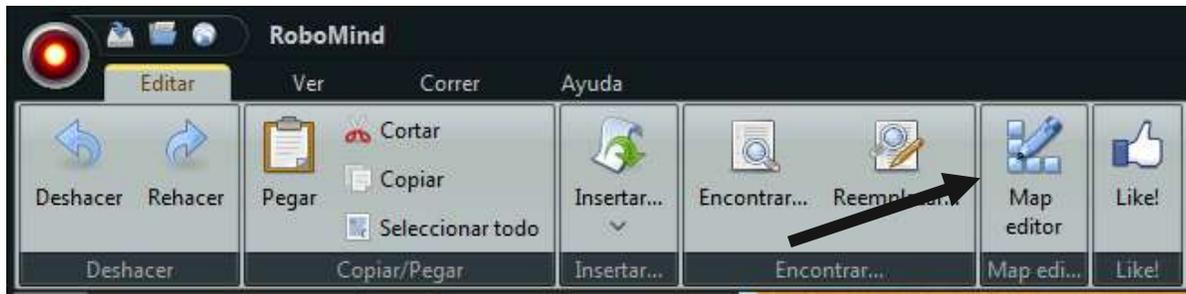
El siguiente grupo de comandos es el de encontrar

Encontrar

Este grupo de comandos consta de dos comandos, Encontrar y Reemplazar. Estos te permiten encontrar y/o reemplazar por otra palabra, un texto en específico.



Map editor



Este comando, te permite modificar escenarios existentes o crear uno nuevo.

Al modificar un escenario o crear uno nuevo, podrás generar nuevos retos.

Ver

La pestaña **Ver**, está integrada por cuatro grupos de comandos: Acercamiento, Mostrar cuadrícula, Crear una instantánea y Sound.

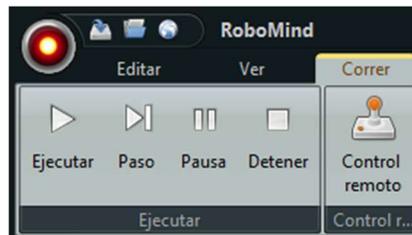


Los comandos de esta pestaña te permiten manipular el escenario, ya sea que necesitas acercar o alejar el escenario para apreciar más a detalle sus elementos, mostrar una cuadrícula, el radar, tomar una instantánea o desactivar los sonidos.

Explora todos y cada uno de estos elementos para que veas y analices que acciones realiza cada uno de ellos.

Correr

Esta pestaña tiene dos grupos de comandos: ejecutar y control remoto.



El menú correr sirve para controlar la ejecución de las instrucciones.

En el grupo de comandos ejecutar se encuentran los comandos:

Ejecutar

Inicia el funcionamiento del robot.

Paso

Hace que ejecuten las instrucciones o comandos uno a la vez.

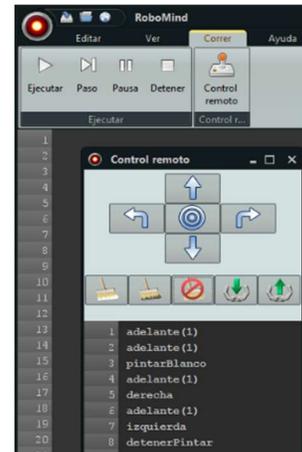
Pausa y Detener

Permiten detener o terminar respectivamente el seguimiento de las instrucciones.

Control remoto

En este menú también se puede controlar el robot manualmente mientras se va generando el código de las instrucciones que se dan al robot. Ese código se puede copiar y pegar en la ventana secuencias de comandos.

Si tu presionas uno de los botones que se incluyen en la ventana control remoto, te va generando el comando que se requiere, vez que terminas de crear el programa o las instrucciones que deseas que realice, las puedes copiar y pegar el editor de programas de la aplicación.



le
de
de
una

Ayuda

Finalmente, no menos importante, la ventana ayuda te ofrece tres opciones

- Tópicos de ayuda. Te ofrece ayuda sobre el uso de esta aplicación.
- RoboMind en línea. Al presionar el ícono de este comando te envía a la aplicación en línea. Esta opción la usaremos posteriormente para realizar nuestros primeros programas para robots.
- Acerca de RoboMind. Te ofrece información referente a la versión de RoboMind.



En este momento, ya conoces el entorno de la aplicación RoboMind, la cual te permitirá iniciar en el mundo de la robótica educativa. Recuerda que esta opción es para cuando la quieras utilizar con tus alumnos esta aplicación y no cuentes con acceso a internet.

El siguiente paso es realizar un registro en la página RoboMind Academy, la cual servirá para iniciar en el proceso de programación y uso de simuladores. Una vez concluido este curso, podrás utilizar esta aplicación, ya que habrás desarrollado las habilidades básicas para el uso de esta herramienta a través del portal RoboMind Academy.

Actividad. Registro y exploración de RoboMind Academy, resolviendo mis primeros retos.

Objetivo: conocer el entorno de RoboMind Academy y realizar los primeros programas que te permitan resolver los retos planteados en esta aplicación web.

Qué se requiere para esta actividad:

- De la lección tres, una impresión de pantalla donde muestres que has seguido los pasos y resuelto el reto.
- De las lecciones cuatro y cinco, anexarás los algoritmos que se crearon para resolverlos, así como impresión de pantalla donde se pueda observar que estos han sido resueltos.

Para realizar y resolver estas actividades, apóyate en el “Tema V. Programando un robot” de esta antología.

Cuando hayas finalizado la lección cinco y hayas recabado las evidencias solicitadas, guárdalo con el nombre `Actividad7_PrimerNombrePrimerApellido`, por ejemplo, `Actividad7_JesúsAlvarado`.

Envíalo a través de la plataforma para su revisión.

Tema V. Programando un robot

Desarrollando mis primeros programas

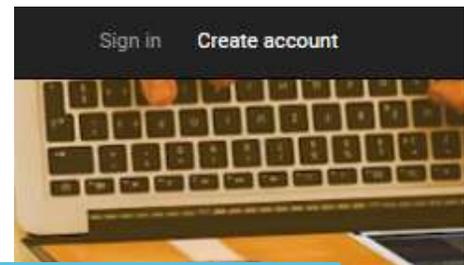
Para empezar a crear tus primeros programas en esta aplicación lo harás utilizando la versión que ofrece RoboMind en línea (RoboMind Academy).

Lo primero que debes hacer es registrarte en su plataforma, para ello abre tu navegador y en la barra de direcciones escribe la siguiente dirección:

<https://www.robomindacademy.com/go/robomind/home>

Ahora realiza lo siguiente:

1. Da clic en “Create account” (Crear una cuenta), que se encuentra ubicada en la parte superior derecha de la página. Para registrarte necesitas una cuenta de correo personal a la que puedas acceder en cualquier momento.
2. Escribe el correo electrónico en lo cuadro de texto que corresponden a “Email” y “Repeat Email” deben de coincidir de lo contrario te enviará en mensaje de error.
3. En el campo “Password” escribe una contraseña, la cual usarás para ingresar a RoboMind Academy. En “Repeat Password” a inscribir la contraseña que acabas de escribir en “Password”, deben de ser las mismas.
4. Presiona el botón “Register” (Registrar).
5. Después de esto recibirás un correo, donde te indica que debes completar el proceso de registro. Para completarlo da clic en la liga o hipervínculo “Click here to finalize your



RoboMind Academy Registration

Email

Repeat Email

Password

Repeat Password

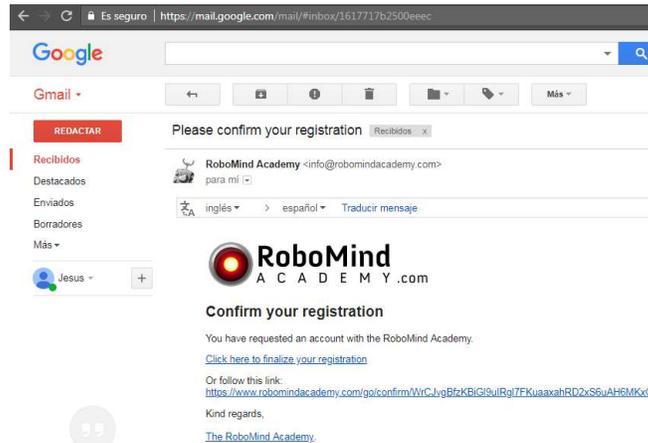
Use a unique password for RoboMind Academy

Register

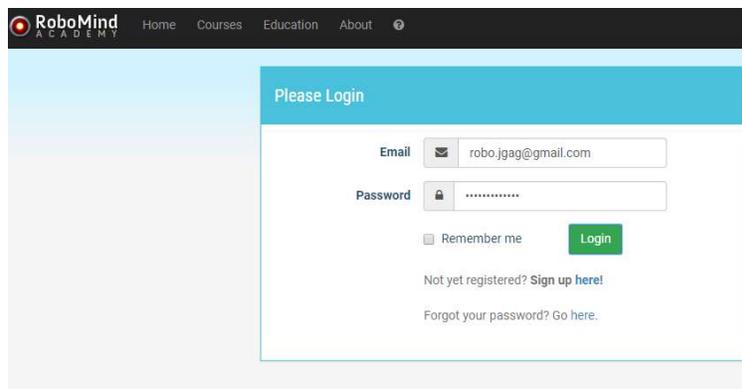
Email”,

vuelve

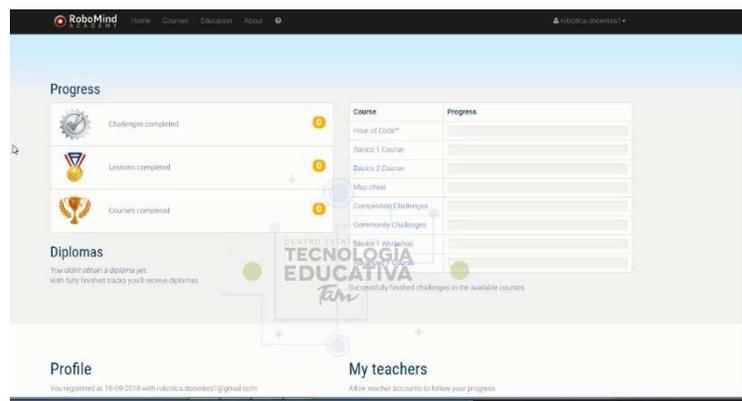
registration” (Clic aquí para finalizar el registro), esto te llevará nuevamente a la página de RoboMind.



6. Al dar clic te enviará a la pantalla de Login. Escribe el correo y la contraseña que utilizaste para registraste. Presiona el botón “Login” para ingresar.

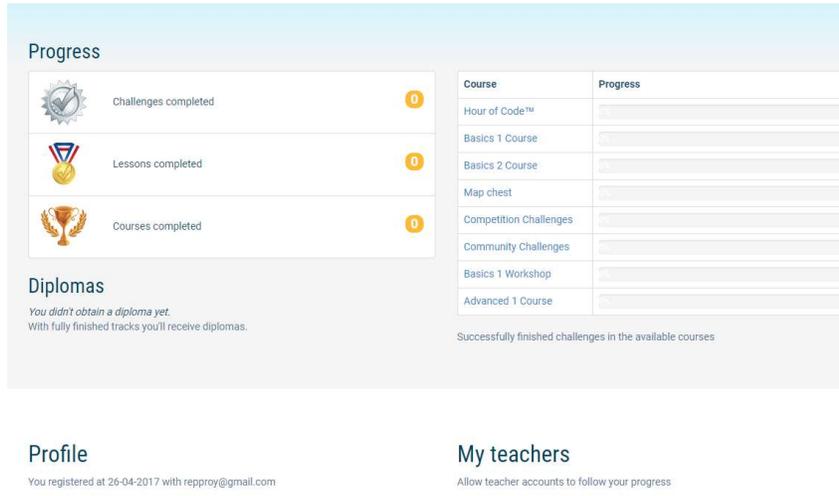


7. Una vez que ingresas a la página, verás tu página de perfil. Entre otra información encontrarás la clave de activación de RoboMind, como se muestra en la imagen inferior.



Entorno de la aplicación RoboMind Academy

Una vez que ingresas a la plataforma de RoboMind Academy, verás una imagen similar a la siguiente, donde podrás ver tu progreso y avances, entre otros aspectos. A esta pantalla se le denomina “Profile” o perfil.



Progress

- Challenges completed
- Lessons completed
- Courses completed

Diplomas
You didn't obtain a diploma yet.
With fully finished tracks you'll receive diplomas.

Profile
You registered at 26-04-2017 with repproy@gmail.com

My teachers
Allow teacher accounts to follow your progress

Course	Progress
Hour of Code™	0%
Basics 1 Course	0%
Basics 2 Course	0%
Map chest	0%
Competition Challenges	0%
Community Challenges	0%
Basics 1 Workshop	0%
Advanced 1 Course	0%

Successfully finished challenges in the available courses

En la sección derecha puedes ver los cursos que puedes tomar y el avance que logres en cada uno de ellos.

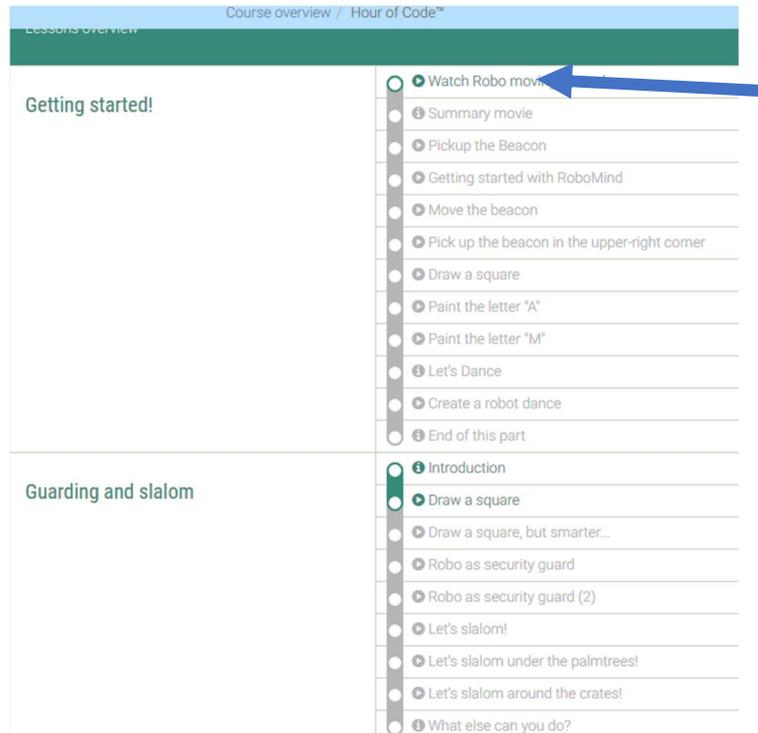
En este caso, como es la primera vez que ingresamos no contamos con ningún avance.

En “Robótica para Docentes I”, trabajaremos con el curso “Hour of Code™” (Hora de código), lo cual te permitirá iniciar con el aprendizaje de la programación para robots.

Para ingresar al curso da clic en “Hour of Code™”. Una vez que ingreses te darás cuenta que está dividido en dos categorías:

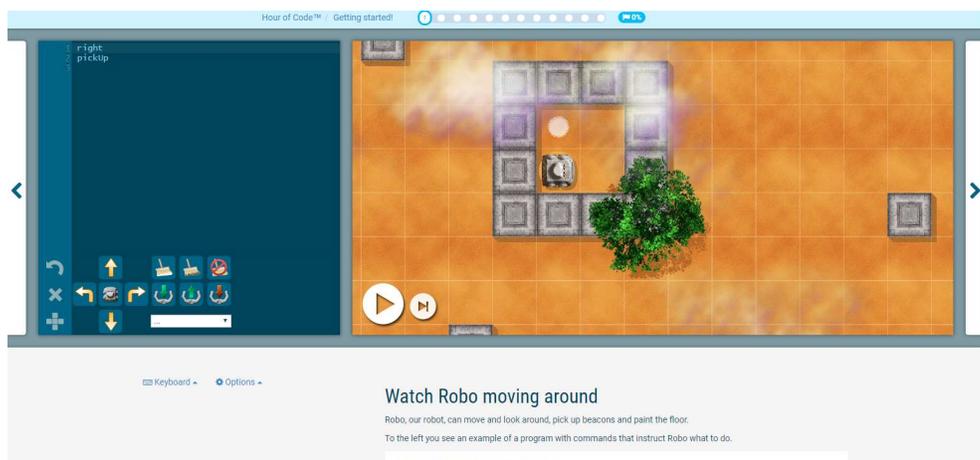
- “Getting started!”
- "Guarding and slalom”

Para iniciar da clic en “Watch Robo moving around” (Mira a Robo Moverse) de la categoría “Getting started!”.



Una vez que presionas el hipervínculo, te muestra un simulador donde podrás ir resolviendo las lecciones.

Describiré brevemente el entorno del simulador.



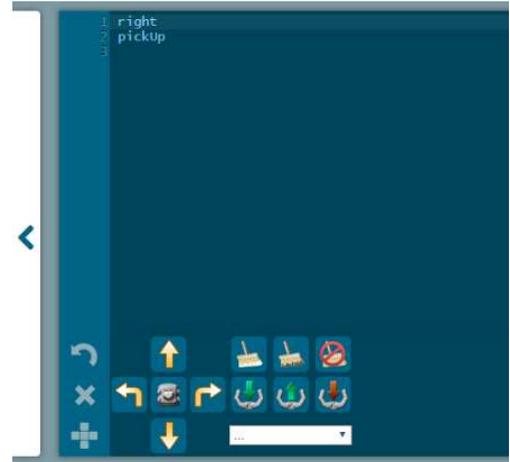
En la parte superior podrás ver la barra de progreso, esta te permite visualizar la lección en la que estás trabajando y moverte a cada una de las lecciones.



En la parte izquierda del simulador, puedes ver el área de código, en ella irás agregando las instrucciones que necesites para un programa.

Para agregar el código en esta sección tienes dos opciones. La primera de ellas es escribirlas manualmente, es decir, como si estuvieras escribiendo en un documento de texto.

La segunda opción, es más sencilla, solo tienes que presionar uno de los íconos, dependiendo de la acción que quieras indicar, y que se encuentran en la parte inferior de esta sección.



Antes de continuar describiremos brevemente estos elementos.

-  Deshace la última acción que realizaste.
-  Borra todo el código.
-  Muestra u oculta el asistente de código.



Los íconos de la izquierda (asistente de código) te permiten insertar código de programa, en lugar de escribirlas mediante el teclado, sólo es necesario que presiones uno de ellos, dependiendo de la acción que requieras.

Los íconos de las “flechas” se usan para insertar código e indicar al robot que realice ciertas acciones como avanzar, retroceder, girar a la izquierda o derecha. El ícono de robot  lo puedes usar para ubicar y visualizar el robot en el escenario.

Los íconos de las brochas te  sirven para ordenarle al robot que pinte una línea, ya sea blanca o negra, también para indicarle que deje de pintar.



Estos íconos sirven para ordenarle al robot que tome, suelte o coma un objeto.



Al presionar el menú tipo “combo”, que se encuentra debajo de los íconos que se utilizan para insertar instrucciones, se desplegará una lista de comandos o instrucciones que puedes agregar a tu programa.

Estos comandos, son similares a los que se describieron anteriormente en la aplicación de escritorio de RoboMind.



Al presionar este ícono iniciará el programa, es decir irá ejecutando y realizando cada una de las líneas de comando que hayas generado en la sección de código. Es similar al botón Play de un reproductor de video. Cada vez que realices modificaciones a tu programa y quieras ver sus resultados presiona este botón.



Ejecuta el programa línea por línea, desde la primera línea del programa.



Cuando presionas el botón Play, los iconos cambian a los siguientes iconos detener o pausar la ejecución del programa.

En la parte lateral del simulador se encuentran estos íconos   , los cuales te permiten avanzar o regresar al siguiente reto.

Si aún no has visto que es lo que realiza el robot, de acuerdo a las instrucciones que tiene en el área de código, presiona el botón Play.

Después de esto presiona el botón  para avanzar a la siguiente actividad.

En esta secuencia de lecciones, la siguiente es un video, donde te muestra algunas de las cosas que puedes hacer con tu robot.

Después de terminar ver el video, da clic en el botón  para avanzar a la siguiente lección. Verifica en la barra de progreso que te encuentres en la lección 3.

NOTA: en algunas ocasiones, después de trabajar en la elaboración de un programa para resolver un reto, puede suceder que no ejecute ninguna acción después de presionar

el botón , te recomiendo que actualices la página para que se cargue correctamente. Esto lo puedes hacer presionando la tecla “F5” o dando clic en el botón actualizar del navegador.



Cada vez que avanzas a la siguiente lección, este inicia automáticamente.



Resolviendo retos

Lección tres. “Pickup the Beacon”

La lección 3 se llama “Pickup the Beacon” (Recoge la baliza). En esta lección se requiere que el robot recoja la baliza u objeto . Esto lo puedes ver en las primeras líneas del código del programa.

```

1  # Let the robot pickup the beacon
2  # in the right corner
3
4  right
5  forward
6  forward
7  left
8  forward(2)
9  pickup
10
    
```

Las líneas que inician con un signo de “#” se usan para realizar comentarios en el programa. En este caso te indican lo que debe cumplir el robot, con base al código que generes. En este caso ya cuenta con unas líneas de código.

Para verificar si con esas líneas de código se cumple el reto, presiona el botón Play  para ver la ejecución del código línea por línea.

Como pudiste observar, estas líneas de código no son suficientes para cumplir el reto, puedes notar que no avanzó lo necesario para llegar al objetivo, en este caso lo que necesitas hacer es agregar las líneas de código que hacen falta, analicemos las líneas que ya nos muestra el reto.

```

1  # Let the robot pickup the beacon
2  # in the right corner
3
4  right
5  forward
6  forward
7  left
8  forward(2)
9  pickup
10
    
```

En la imagen de la izquierda puedes ver el código, en la parte izquierda de cada instrucción nos muestra un indicador  para que puedas identificar de una manera más sencilla que línea de código está realizando el robot.

El primer movimiento que realiza el robot es girar a su derecha (right), presiona nuevamente el botón para ejecutar el programa paso a paso .

La siguiente línea es avanzar “forward”, en la siguiente nuevamente avanza, posteriormente gira a su izquierda (left). Nuevamente encontramos la instrucción avanzar con una variante “forward(2)”, esto indica que el robot avanzará dos veces; escribirlo de esta manera nos ahorra líneas de código, es como si escribiéramos dos veces forward, similar a las líneas 5 y 6 del

programa; finalmente le indica que recoja (pickup) la baliza, pero como no llega, no la puede recoger.



Analicemos en programa de acuerdo al mapa donde se tiene que resolver el reto. Antes de continuar comentaré lo siguiente:

Esta imagen  en el mapa indica el punto de partida del robot, observa además que el mapa está

cuadrulado  como si fueran coordenadas,

así que lo podemos utilizar como referencia para saber qué acción realiza en cada posición, así que iniciemos el recorrido.

El código es el siguiente:

1. right
2. forward
3. forward
4. left
5. forward(2)

Los comandos o líneas de código que corresponden a giros, recoger, soltar, comer, pintar o dejar de pintar se ejecutan en la posición que se encuentra ubicado. Para recoger o soltar necesita estar frente al objeto o frente a la posición en la que desea soltar el objeto.

De acuerdo a lo anterior el código indica lo siguiente:

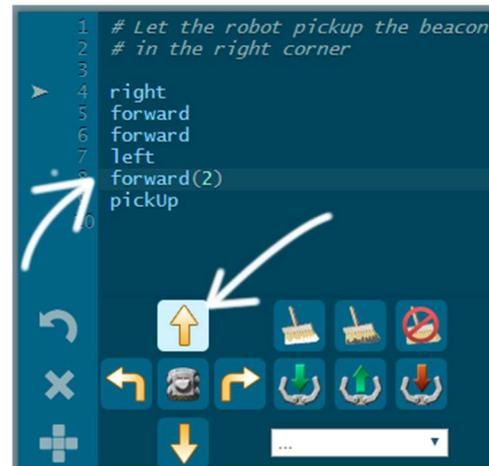
1. Gira a la derecha en su posición de inicio
2. Avanza a la siguiente posición
3. Nuevamente vuelve a avanzar
4. Gira a la izquierda en la posición actual
5. Avanza dos posiciones o cuadros (*forward(2)*)
6. En la posición actual recoge la baliza.



Con las líneas de código de este programa no se puede resolver este reto, ya que aún no está frente al objeto que tiene que recoger, le falta avanzar una posición o cuadro más y dar un giro a la derecha, esto antes de indicarle que recoja la baliza.

Para agregar el código o instrucciones que le faltan realiza lo siguiente:

1. Da clic en la sección del código del programa
2. Para indicarle que avance una posición más se puede hacer de dos maneras:
 - a) Cambiar el parámetro "2" de *forward(2)* a *forward(3)*, es decir, cambiando el dos por el tres.
 - b) Agregar una instrucción *forward*, para ello coloca el puntero del mouse antes de *forward(2)* y da clic en el ícono de la instrucción *forward*.



tres.

Prueba el programa, presiona el botón Play para que inicie.



Una vez que se ejecuta el programa, podrás darte cuenta que el robot está al mismo nivel de la baliza, ahora, solo hace falta indicarle que de un giro a la derecha para que el robot quede frente a la baliza antes de agregar la instrucción para que la recoja.

Para insertar el código coloca el puntero del mouse antes del comando o instrucción

“pickUp”, posteriormente da clic en el ícono del comando “right”. El código modificado es el siguiente:

1. right
2. forward
3. forward
4. left
5. forward
6. forward(2)
7. right
8. pickUp

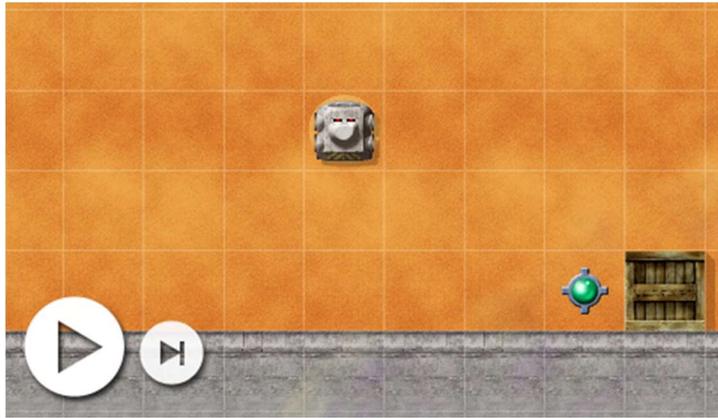
Para probar si efectivamente se puede resolver el reto con las modificaciones que se hicieron al programa, presiona el botón Play.

Si el código que creaste en tu computadora es igual al anterior, se resolverá el reto, el robot después de recoger la baliza gira sobre su eje para indicar que se ha resuelto con éxito reto. Felicidades, has creado tu primer programa.

Puedes avanzar al siguiente reto.

Lección cuatro. “Pick up another beacon yourself”

El reto que se plantea en la lección 4 es que debes recoger la baliza que se encuentra en la parte inferior del mapa. En la imagen siguiente puedes ver una parte de este mapa.



Como puedes observar, para recoger la baliza el robot tiene que desplazarse a la parte inferior del mapa.

Para resolver este reto primero realizaremos un algoritmo, que son los pasos a seguir para resolver un problema. Estos pasos dependen de quien los realice, ya que están basados en la lógica de cada individuo.

Este reto lo resolvería de la siguiente manera:

1. Girar a la derecha
2. Avanzar tres posiciones
3. Girar a la derecha
4. Avanzar una posición
5. Recoger baliza

Es hora de transformar nuestro algoritmo en un programa. Utilizando el algoritmo que se elaboró, agregaremos los comandos o instrucciones para el robot. En la siguiente tabla encontrarás la codificación, en la parte izquierda se encuentran los pasos a seguir (algoritmo) y en la parte derecha el comando que indica la acción para el robot (programa).

Girar a la derecha	derecha	
Avanzar tres posiciones	adelante(3)	
Girar a la derecha	derecha	
Avanzar una posición	adelante	
Recoger baliza	tomar	

Ahora da clic en la sección de código de programa para agregar los comandos. Utiliza los íconos para insertar el código. Una vez insertado el código, este debe ser similar al siguiente:

```

1 derecha
2 adelante
3 adelante
4 adelante
5 derecha
6 adelante
7 tomar
8
  
```

Una vez que hayas completado el código del programa, verifica si funciona correctamente dando clic en el botón play  para ver si se cumple el reto. Como resultado del programa el robot debe recoger la baliza.

Cumplido el reto puedes avanzar a la lección número cinco, da clic en la barra de progreso en el punto correspondiente a la lección número cinco.



Lección cinco. "Move the beacon"

En esta lección el reto que se debe cumplir es recoger la baliza, y ponerla en la esquina inferior izquierda. En la imagen inferior puedes ver la ruta sugerida para resolver el ejercicio.



Observa en la sección de código que se mantienen las instrucciones que se utilizaron para resolver la lección cuatro.

```
1 # Let the robot pick up the
2 # beacon to its lower right.
3 derecha
4 adelante(3)
5 derecha
6 adelante(2)
7 tomar
```

Para resolver esta lección elaboraremos el algoritmo y posteriormente agregaremos las instrucciones que complementen el programa y dar solución a este reto. Si deseas ver o comprobar cuáles son las acciones o pasos que realiza el robot presiona el botón play .



Como puedes ver, con el código actual solamente logramos recoger la baliza. De acuerdo con mi lógica, los pasos que hacen falta para resolver el ejercicio son los siguientes:

1. Avanzar un paso
2. Girar a la izquierda
3. Avanzar siete posiciones
4. Soltar la baliza

Las instrucciones que complementan el programa serían las siguientes:

Avanzar un paso	adelante	
Girar a la izquierda	derecha	
Avanzar siete posiciones	adelante(7)	
Soltar la baliza	poner	

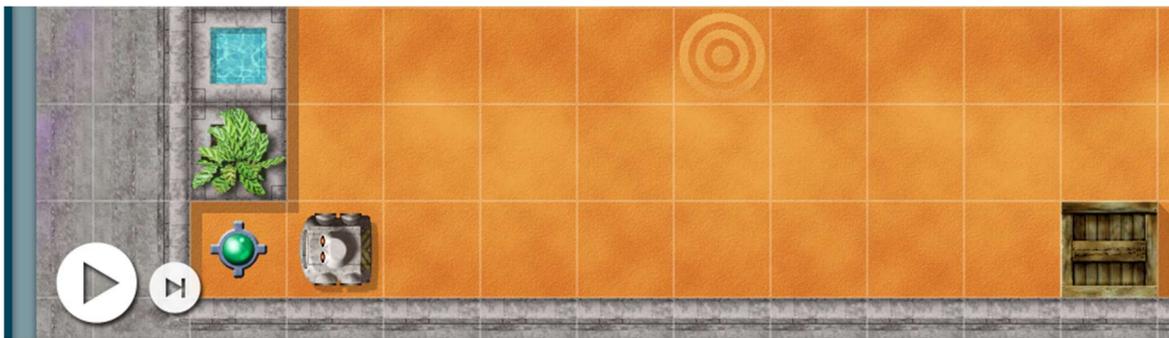
El código completo en la sección de códigos será:

```

1 # Let the robot pick up the
2 # beacon to its lower right.
3 derecha
4 adelante(3)
5 derecha
6 adelante(2)
7 tomar
8 adelante
9 derecha
10 adelante(7)
11 poner
12

```

Después de escribir el código, presiona al botón Play  para verificar que se resuelva el reto. Como resultado el robot debe colocar o soltar la baliza en la esquina inferior izquierda del mapa, como se muestra en la imagen siguiente:



Una vez resuelto el reto anterior, puedes avanzar a la lección seis, para ello da clic en la barra de progreso, lección 6. En este momento llevas un avance del 25%.



Actividad. Lección cuatro. “Pick up the beacon in the upper-right corner”

En esta lección el reto es que diseñes un programa que te permita recoger la baliza que se encuentra en la parte superior derecha del mapa.



Instrucciones:

Para resolver esta actividad debes de realizar lo siguiente:

1. En un documento de texto (Word) nuevo elabora el algoritmo que te indique los pasos a seguir para resolver el ejercicio de la lección cuatro.
2. Una vez que hayas finalizado el algoritmo, guarda el documento en la carpeta de actividades de este curso de la siguiente manera: Actividad8_PrimerNombrePrimerApellido. Por ejemplo, Actividad8_JesúsAlvarado.
3. En la página de RoboMind Academy, utiliza la sección de código para agregar los comandos o instrucciones que necesitas para resolver el ejercicio, estas instrucciones deben ir acorde a tu algoritmo, tal como lo realizamos en las lecciones anteriores.
4. Una vez que termines de agregar los comandos, prueba el programa. En caso de ser necesario, realiza las correcciones necesarias.
5. Una vez que resuelvas el ejercicio planteado en la lección seis, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo. Recuerda que para que el reto sea marcado como resuelto, debes poder avanzar a la siguiente lección.
6. Guarda nuevamente el documento con las modificaciones que acabas de realizar. Envía el archivo para su revisión a través de la plataforma.

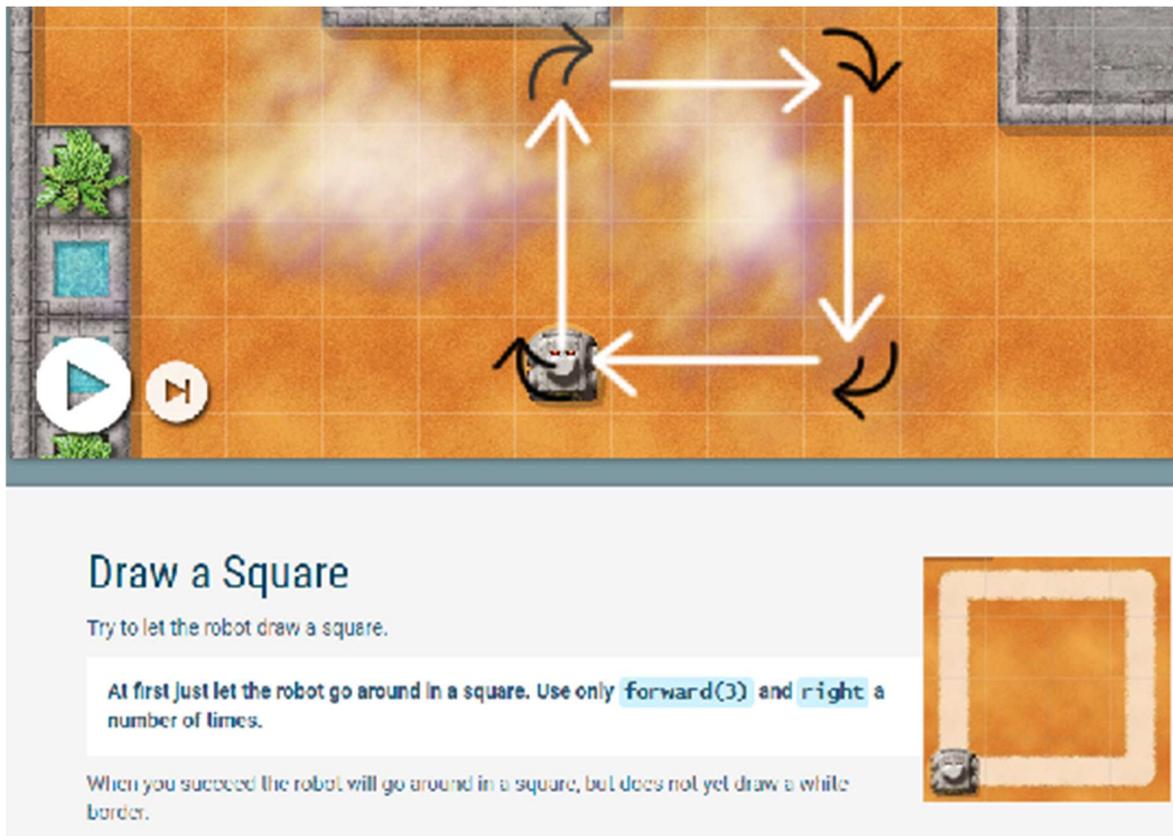
Felicidades, acabas de crear tu primer programa para robot. Es hora de continuar aprendiendo un poco más.

Práctica. Lección siete. “Draw a Square”

El reto a resolver en la lección siete, es dibujar un cuadro en el mapa. Las dimensiones del cuadro a dibujar deben de ser 3 x 3 (tres cuadros por lados), además debe dibujar o pintar una línea blanca.

Solo debes utilizar la instrucciones `forward(3)` y `right`, es decir, avanzar 3 posiciones y girar a la derecha.

Al igual que en los ejercicios anteriores, lo primero que se debe realizar es el algoritmo que permita dar respuesta al reto.



Draw a Square

Try to let the robot draw a square.

At first just let the robot go around in a square. Use only `forward(3)` and `right` a number of times.

When you succeed the robot will go around in a square, but does not yet draw a white border.

En la imagen anterior puedes ver los pasos que debe seguir el robot para cumplir el reto. Como ya se mencionó, el robot debe de pintar un cuadro de 3x3, resolveremos este ejercicio de la manera en la que lo hemos estado haciendo, utilizando instrucciones básicas, agregando el comando `pintarBlanco`. Realicemos el algoritmo.

Como primer punto debemos definir si el primer paso que hay que hacer es avanzar o empezar a pintar, en este caso lo primero que se necesita hacer es indicarle al robot que primero pinte, el algoritmo sería el siguiente:

Empezar a pintar

Avanzar tres posiciones

Girar a la derecha

Dejar de pintar

El código del programa, tomando como referencia el algoritmo, sería el siguiente:

Empezar a pintar	pintarBlanco	
Avanzar tres posiciones	adelante(3)	
Girar a la derecha	derecha	
Avanzar tres posiciones	adelante(3)	
Girar a la derecha	derecha	
Avanzar tres posiciones	adelante(3)	
Girar a la derecha	derecha	
Avanzar tres posiciones	adelante(3)	
		

Girar a la derecha

derecha



Dejar de pintar

detenerPintar

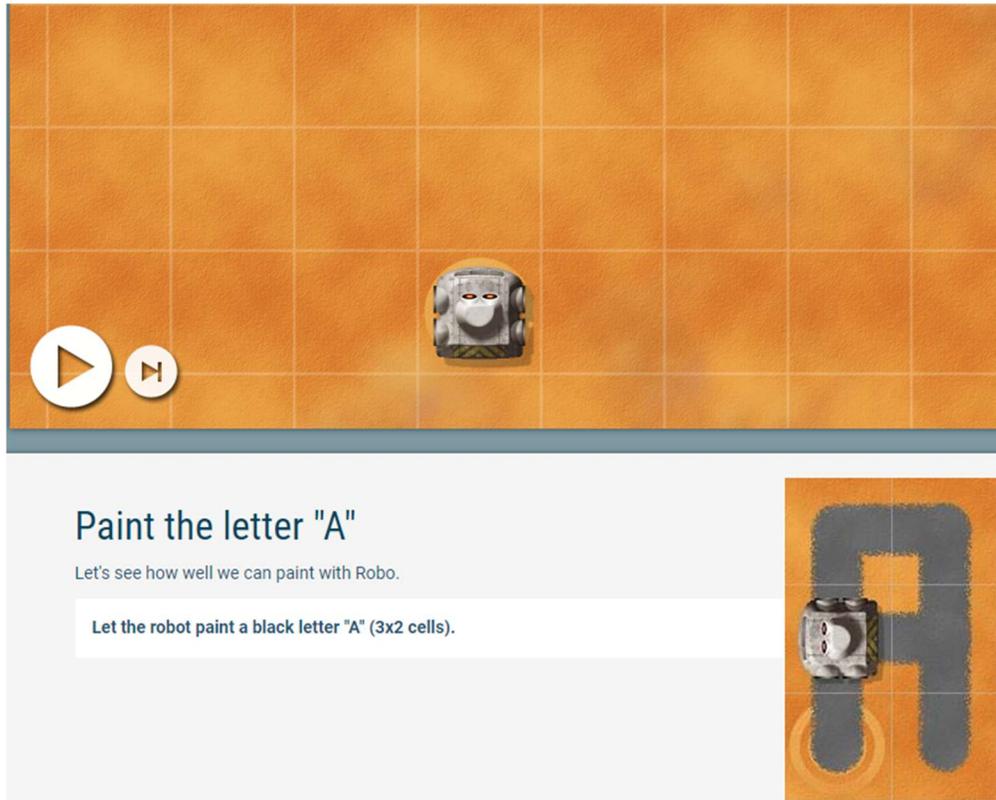
Es hora de verificar el programa, da clic en el botón Play  para ejecutarlo, observa los resultados. Si el código de tu programa lo creaste igual al de la lectura, debe de cumplir el reto.

Actividad. Resolver las lecciones ocho, nueve y once.

Es hora de aplicar lo aprendido, para ello resuelve las lecciones ocho, nueve y once. Los retos que debes cumplir en cada uno de ellos son los siguientes:

Lección ocho. “Paint the letter "A"”

El reto a cumplir en la lección ocho es pintar en negro una letra “A”.

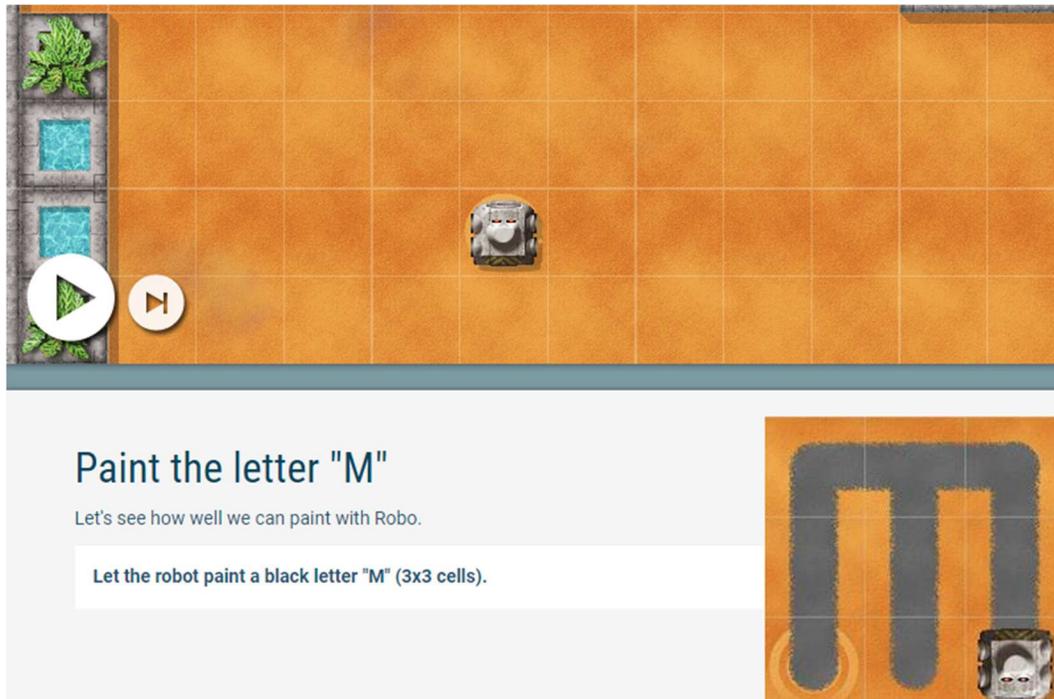


En la imagen anterior puedes ver un ejemplo de cómo quedaría la letra “A” en el mapa una vez que generes el programa para este reto, así como las medidas que debe abarcar (3x2).

Lección nueve. “Paint the letter "M"”

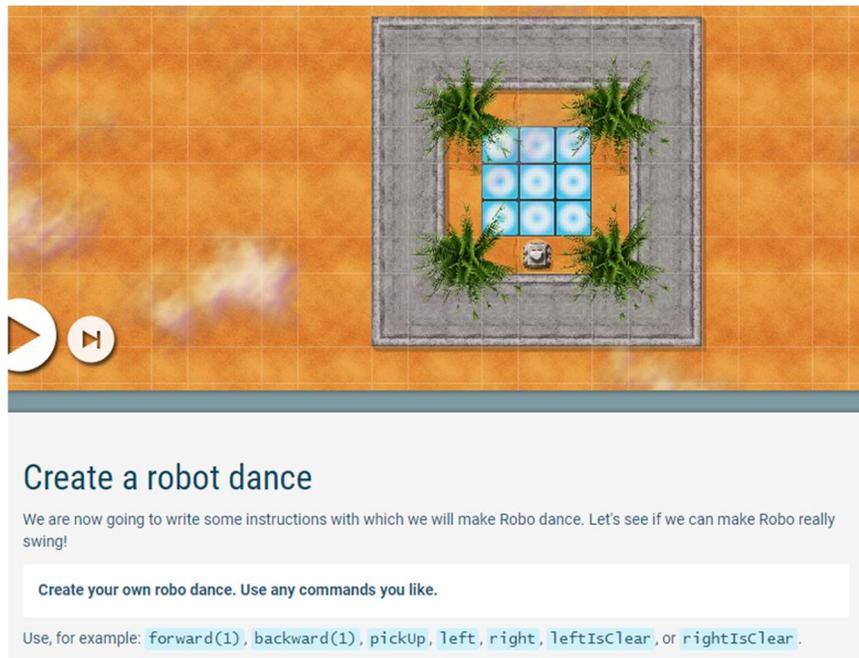
El reto a cumplir en la lección nueve es pintar la letra “M”, tal y como se muestra en la imagen siguiente.

Para este reto se deben utilizar tres celdas (3x3). Recuerda elaborar el algoritmo que te permita resolver esta lección.



Lección once. “Create a robot dance”

El objetivo de esta lección es hacer que tu robot “baile”. Para ello debes utilizar, al igual que en las lecciones anteriores, los comandos básicos de movimientos. En el mapa puedes ver la “pista” de baile, procura que el robot se mueva por toda la pista, utiliza la cantidad de comandos que consideres necesarios.



Para realizar esta actividad, donde debes dar solución a las tres lecciones (ocho, nueve y once) considera y realiza lo siguiente:

- Recuerda que debes tener resuelta la lección siete para pasar a la siguiente.
1. Abre un documento de texto nuevo (Word).
 2. Elabora un algoritmo que te permita crear el programa para dar solución al reto planteado en cada lección.
 3. Una vez que hayas finalizado el algoritmo, guarda el documento en la carpeta de actividades de este curso de la siguiente manera: Actividad8_PrimerNombrePrimerApellido. Por ejemplo, Actividad8_JesúsAlvarado.
 4. En la página de RoboMind Academy, utiliza la sección de código para agregar los comandos o instrucciones que necesitas para resolver el ejercicio. Estas instrucciones deben ir acorde a tu algoritmo, tal como lo realizaste en las lecciones anteriores.
 5. Una vez que termines de agregar los comandos, prueba el programa.
 6. Resuelto el ejercicio planteado en la lección, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo.
 7. Guarda nuevamente el documento con las modificaciones que acabas de realizar.
 8. Repite los pasos del dos al siete (2 – 7) hasta resolver las tres lecciones.

9. Cuando hayas resuelto las lecciones, envía el archivo para su revisión a través de la plataforma / Guárdalo en la Carpeta de Actividades de este curso.

Al finalizar la lección once, habrás concluido la primera parte de estos retos. Debes tener tu insignia que lo acredite. FELICIDADES.



Estructuras de programación.

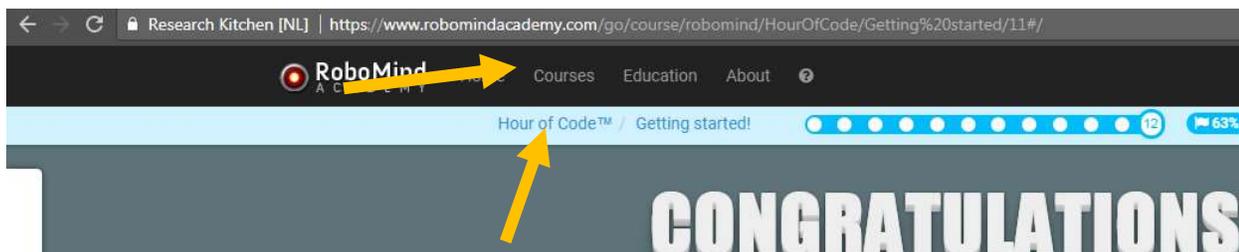
De manera general podemos decir que las estructuras de programación son parte fundamental de cualquier lenguaje. Sin ellas, las instrucciones de un programa solo podrían ejecutarse en el orden en que están escritas (orden secuencial). Las estructuras de control o de programación permiten modificar este orden.

Hay dos categorías de estructuras de control:

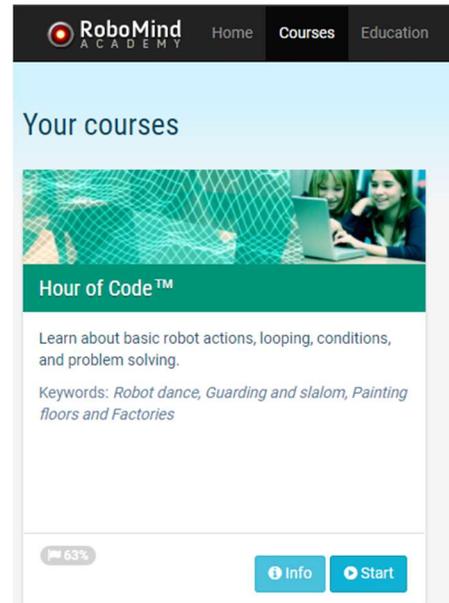
Condicionales o bifurcaciones. permiten que se ejecuten conjuntos distintos de instrucciones, en función de que se verifiquen determinadas condiciones, pueden ser una o varias.

Bucles o repeticiones. permiten que se ejecute repetidamente un conjunto de instrucciones, ya sea un número pre-determinado de veces, o bien hasta que se cumpla una determinada condición.

Pongamos en práctica lo anterior. Para ello continuemos con la secuencia de aprendizaje de la programación para robots, da clic en el menú “Courses” o en “Hour of Code”.



Ubica el curso “Hour of Code” y da clic en el botón “Start”



Da clic en “Introduction” de la categoría “Guarding and slalom”

Getting started!

●	Summary movie	👁
●	Pickup the Beacon	✓
●	Getting started with RoboMind	✓
●	Move the beacon	✓
●	Pick up the beacon in the upper-right corner	✓
●	Draw a square	✓
●	Paint the letter "A"	✓
●	Paint the letter "M"	✓
●	Let's Dance	👁
●	Create a robot dance	✓
●	End of this part	👁
●	Introduction	👁
●	Draw a square	✓
●	Draw a square, but smarter...	

Guarding and slalom

Una vez que ingreses en la lección “Introduction” te sugiero la explores, para ello utiliza el botón siguiente.

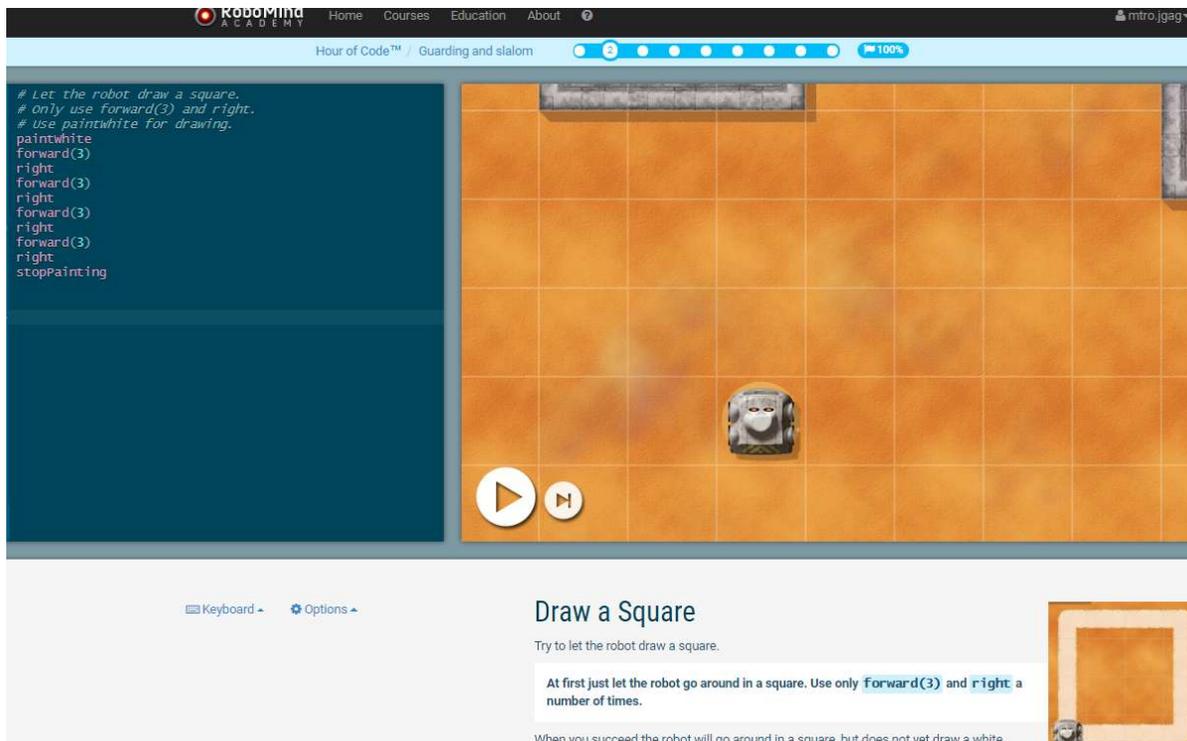
En esta lección se aborda los temas y retos referente al uso de estructuras de programación.



Después de ver y analizar la lección “Introduction” continúa con la lección 2.

Resumiendo, la lección “Introduction” al utilizar las instrucciones de estructura como “repeat” nos sirve para ejecutar continuamente una cantidad “n” de veces, una o varias instrucciones, esto como ya se mencionó, nos ayuda, entre otras cosas, a escribir menos cantidad de líneas de código.

El código para resolver la lección dos, sería similar al de la imagen siguiente:



Hour of Code™ / Guarding and slalom

```
# Let the robot draw a square.
# Only use forward(3) and right.
# Use paintwhite for drawing.
paintwhite
forward(3)
right
forward(3)
right
forward(3)
right
forward(3)
right
stopPainting
```

Draw a Square

Try to let the robot draw a square.

At first just let the robot go around in a square. Use only `forward(3)` and `right` a number of times.

When you succeed the robot will go around in a square, but does not yet draw a white

Para aprender y practicar un poco sobre el uso de estas instrucciones avanza a la lección tres “Draw a square, but smarter...”

Práctica. Lección tres. “Draw a square, but smarter...”

Para resolver esta lección también es necesario pintar un cuadro blanco en el mapa, hasta aquí no debes tener ningún problema para resolverlo, ya que este reto lo realizaste en una lección anterior.

En esta lección se sugiere que utilices la instrucción repeat (repetir). Es muy seguro que en esta lección ya tengas un código base para resolver el reto, sin embargo, te sugiero que crees uno nuevo. Lo primero que harás es elaborar el algoritmo para resolver el ejercicio.

Empezar a pintar

Avanzar tres posiciones

Girar a la derecha

Avanzar tres posiciones

Girar a la derecha

Avanzar tres posiciones
Girar a la derecha
Avanzar tres posiciones
Girar a la derecha
Dejar de pintar

Si analizas el algoritmo anterior, hay pasos que se repiten cuatro veces, estos son: Avanzar tres posiciones y Girar a la derecha, es en esta parte donde puedes utilizar las instrucciones de estructura como repetir. Utilizando las instrucciones de repetición y las que ya se tienen, el algoritmo quedaría de la siguiente manera:

Sin usar instrucciones o estructuras de repetición

Empezar a pintar

Avanzar tres posiciones
Girar a la derecha
Avanzar tres posiciones
Girar a la derecha
Avanzar tres posiciones
Girar a la derecha
Avanzar tres posiciones
Girar a la derecha

Dejar de pintar

Usando estructuras de Repetición

Empezar a pintar

Repetir 4 veces

Avanzar tres posiciones

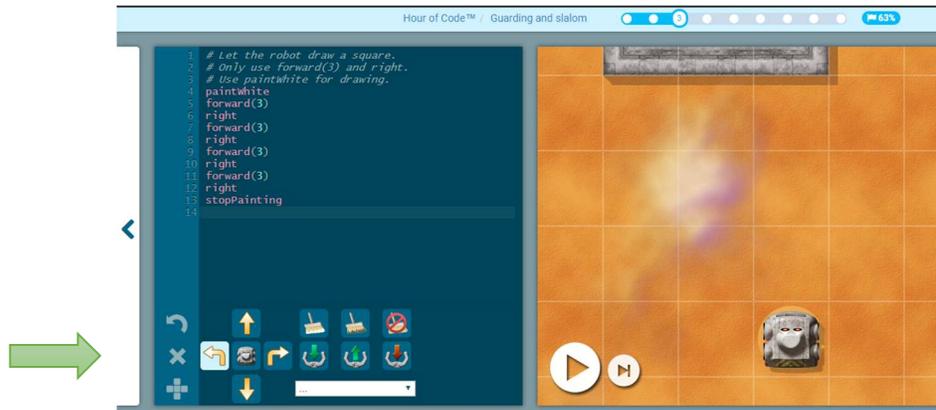
Girar a la derecha

Dejar de pintar

Si comparas los algoritmos, al utilizar la estructura de repetición “Repetir”, la cantidad de pasos se reduce considerablemente.

Es hora de crear el código tomando como referencia el algoritmo que se acaba de elaborar, realiza lo siguiente:

1. Si ya existe código en la lección, utiliza la opción  para borrar el código existente.



2. Presiona el icono para agregar la instrucción pintarBlanco
3. Ahora agregaremos la instrucción repetir, da clic en el combo seleccionar la instrucción repetir.

para

Nuestro código va así:

pintarBlanco

repetir

{

}



4. Da clic al final de la instrucción repetir y abre paréntesis, escribe “4” que es el número de veces que se repetirán las instrucciones que están dentro de los corchetes “{}”, cierra paréntesis, el código debe quedar así:

pintarBlanco

Repetir(4)

{

}

5. Ahora da clic en una de las líneas dentro de los corchetes para insertar las instrucciones que deseamos se repitan. Recuerda apoyarte en el algoritmo.
6. Presionamos el ícono “adelante” .
7. Da clic al final de la instrucción “adelante” abre paréntesis, escribe 3 y cierra paréntesis.

8. Presiona el ícono “derecha” .
9. Da clic en la línea que se encuentra después del corchete cerrar “}”. Si no hay una línea posterior, da clic después del corchete y presiona “enter”.
10. Da clic en el ícono “detenerPintar” .
11. El código generado para esta lección debe ser el siguiente

```

1  # Let the robot draw a square.
2  # Only use forward(3) and right.
3  # Use paintWhite for drawing.
4  pintarBlanco
5  repetir(4)
6  {
7  adelante(3)
8  derecha
9
10 }
11 detenerPintar
12

```

12. Para verificar que tu código funciona y se resuelve el reto, da clic en el botón Play .

Actividad. Resolver las lecciones cuatro, cinco y seis.

Lección cuatro. “Robo as security guard”

Recuerda que se está trabajando en la categoría “Guarding and slalom” del curso “Hour of Code”.

Resuelve el reto planteado en la lección 4, “Robo as security guard”.

¿Qué necesitas hacer Robo para resolver el reto?



El objetivo de este reto es que Robo se ubique en la posición 1, y de una vuelta completa a la figura, como se muestra en la imagen anterior.

- Recuerda que debes resolver la lección actual para pasar a la siguiente.
1. Abre un documento de texto nuevo (Word)
 2. Elabora un algoritmo que te permita crear el programa para dar solución al reto planteado en cada lección.

3. Una vez que hayas finalizado el algoritmo, guarda el documento en la carpeta de actividades de este curso de la siguiente manera: Actividad10_PrimerNombrePrimerApellido. Por ejemplo, Actividad10_JesúsAlvarado.
4. En la página de RoboMind Academy, ubica la lección cuatro. “Robo as security guard” y utilizando la sección de código agrega los comandos o instrucciones que necesitas para resolver el ejercicio. Estas instrucciones deben ir acorde a tu algoritmo, tal como lo realizamos en las lecciones anteriores.
5. Una vez que termines de agregar los comandos, prueba el programa. El código de tu programa debe incluir las instrucciones: adelante, izquierda y repetir.
6. Una vez que resuelvas el ejercicio planteado en la lección, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo.
7. Guarda nuevamente el documento con las modificaciones que acabas de realizar.
8. Envía el archivo para su revisión a través de la plataforma.

Lección cinco. “Robo as security guard (2)”

Resuelve el reto planteado en la lección 5, “Robo as security guard”.

¿Qué necesita hacer Robo para resolver el reto?



El objetivo de este reto es que Robo se ubique en la posición 1, y de tres vueltas completa a la figura, como se muestra en la imagen anterior.

- Recuerda que debes resolver la lección actual para pasar a la siguiente.
1. Abre el documento de texto que creaste para esta actividad (Actividad10_PrimerNombrePrimerApellido)
 2. Elabora un algoritmo que te permita crear el programa para dar solución al reto planteado en esta lección.
 3. Una vez que hayas finalizado el algoritmo, guarda el documento con las modificaciones o cambios que has realizado.
 4. En la página de RoboMind Academy, ubica la lección cinco. “Robo as security guard (2), utiliza la sección de código para agregar los comandos o instrucciones que necesitas para

resolver el ejercicio. Estas instrucciones deben ir acorde a tu algoritmo, tal como lo realizamos en las lecciones anteriores.

5. Una vez que termines de agregar los comandos, prueba el programa. El código de tu programa debe incluir las instrucciones: adelante, izquierda y repetir.
6. Una vez que resuelvas el ejercicio planteado en la lección, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo. (Actividad10_PrimerNombrePrimerApellido)
7. Guarda nuevamente el documento con las modificaciones que acabas de realizar.

Tips Para resolver este reto debes utilizar otra instrucción repeat.

Recuerda que los corchetes "{...}" indican que lo que se escribe dentro de ellos se repite las veces que se indique.

Lección seis. "Let's slalom!"

Resuelve el reto planteado en la lección 6, "Let's slalom!".

¿Qué necesita hacer Robo para resolver el

Lo primero que debe hacer Robo, es ubicarse en la posición marcada en el mapa.

Posteriormente realizar el recorrido marcado por los cuadros celestes, al llegar final recoger la baliza de la parte media.



reto?

al

Como puedes ver en la sección de código muestra tres segmentos de código:

```

1  # Let the robot follow the path
2  # with the blue tiles and pick
3  # up the beacon at the end.
4
5  # 1. Go to the start:
6
7  # 2. How many times to repeat:
8  repeat(0)
9  {
10     forward(2)
11     right
12     forward(2)
13     left
14     forward(2)
15     left
16     forward(2)
17     right
18 }
19
20 # 3: Pick up the beacon:
21
    
```

1. Go to the start: (recuerda que el signo “#” se utiliza para indicar que la línea que se escribe es un comentario)

2. How many times to repeat:

```

repeat(0)
{
    forward(2)
    right
    forward(2)
    left
    forward(2)
    left
    forward(2)
    right
}
    
```

3: Pick up the beacon:

Como puedes ver, ya tienes parte del código para resolver el reto, sin embargo, necesitas verificar que es lo que hace falta agregar para resolverlo, para ello realiza lo siguiente:

1. Abre el documento de texto que creaste para esta actividad. (Actividad10_PrimerNombrePrimerApellido)
2. Elabora un algoritmo que te permita crear el programa para dar solución al reto planteado en cada lección. Considera:
 - ¿Qué pasos debe realizar Robo para llegar al punto de partida?

- Una vez que ha llegado al punto de partida, ¿Necesito modificar el código que tengo dentro del ciclo “repeat”?
3. Una vez que hayas finalizado el algoritmo, guarda el documento.
 4. En la página de RoboMind Academy, verifica que estés ubicado en la lección seis. “Let's slalom!”, y utiliza la sección de código para agregar los comandos o instrucciones que necesitas, así como para complementar lo que ya se tiene. Estas instrucciones deben ir estructuradas con base a tu algoritmo.
 5. Una vez que termines de añadir los comandos, prueba el programa. El código de tu programa debe incluir las instrucciones que has utilizado hasta el momento (adelante, izquierda y repeat, etc)
 6. Una vez que resuelvas el ejercicio planteado en la lección, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo.
 7. Guarda nuevamente el documento con las modificaciones que acabas de realizar.
 8. Cuando hayas resuelto las tres lecciones, envía el archivo (Actividad10_PrimerNombrePrimerApellido) para su revisión a través de la plataforma.

Actividad. Resolver lecciones siete y ocho.

Lección siete. “Let's slalom under the palmtrees!”

Resuelve el reto planteado en la lección 7 “Let's slalom under the palmtrees!”.

¿Qué necesita hacer Robo para resolver el

En este caso, debe realizar el recorrido al de los azulejos o cuadros azules, solo ahora debe hacerlo a través de las palmeras e iniciar desde el punto marcado.



“Let's

reto?

similar

que

Así que

modifica el código que analiza el código que ya se tiene y realiza las modificaciones necesarias para resolver el reto de esta lección.

Realiza lo siguiente:

1. Abre un documento nuevo de texto nuevo (Word)
2. Elabora un algoritmo que te permita crear el programa para dar solución al reto planteado en cada lección. Considera:
 - ¿Qué pasos debe realizar Robo para llegar al punto de partida?
 - Una vez que ha llegado al punto de partida, ¿Necesito modificar el código que tengo dentro del ciclo “repeat”?
3. Finalizado el algoritmo, guarda el documento en la carpeta de actividades de este curso de la siguiente manera: Actividad11_PrimerNombrePrimerApellido, ejemplo “Actividad11_JesúsAlvarado”.
4. En la página de RoboMind Academy, verifica que estés en la lección siete. “Let's slalom under the palmtrees!”, y utiliza la sección de código para agregar y/o modificar los comandos o instrucciones que necesitas. Estas instrucciones deben ir acorde a tu algoritmo.

5. Una vez que termines de agregar los comandos, prueba el programa. El código de tu programa debe incluir las instrucciones que has utilizado hasta el momento (adelante, izquierda y repeat, etc).
6. Utiliza el botón Play para probar el código generado. Una vez que resuelvas el ejercicio planteado en la lección, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo.
7. Guarda nuevamente el documento con las modificaciones que acabas de realizar.
8. Envía el archivo para su revisión a través de la plataforma.

Lección ocho. “Let's slalom around the crates!”

Resuelve el reto planteado en la lección 8
“(Let's slalom around the crates!”.

¿Qué necesita hacer Robo para resolver
reto?

En este caso, debe realizar el recorrido
similar al de los azulejos o cuadros azules,
que ahora debe hacerlo a través de las



el

solo
cajas

e iniciar desde el punto marcado. Así que modifica el código que analiza el código que ya se tiene y realiza las modificaciones necesarias para resolver el reto de esta lección.

Para ello realiza lo siguiente

1. Abre el documento de texto que creaste para esta actividad (Actividad11_PrimerNombrePrimerApellido).
2. Elabora un algoritmo que te permita crear el programa para dar solución al reto planteado en cada lección. Considera:
 - ¿Qué pasos debe realizar Robo para llegar al punto de partida?
 - Una vez que ha llegado al punto de partida, ¿Necesito modificar el código que tengo dentro del ciclo “repeat”?
3. Una vez que finalices el algoritmo, guarda el documento con las modificaciones que realizaste.

4. En la página de RoboMind Academy, utiliza la sección de código para agregar y/o modificar los comandos o instrucciones que necesitas. Estas instrucciones deben ir acorde a tu algoritmo.
5. Una vez que termines de agregar los comandos, prueba el programa. El código de tu programa debe incluir las instrucciones que has utilizado hasta el momento (adelante, izquierda y repeat, etc).
6. Utiliza el botón Play para probar el código generado. Una vez que resuelvas el ejercicio planteado en la lección, realiza una impresión de pantalla y cópiala en el documento en el que realizaste el algoritmo.
7. Guarda nuevamente el documento con las modificaciones que acabas de realizar.
8. Envía el archivo para su revisión a través de la plataforma.

Con esta actividad has desarrollado los principios básicos de la programación para robots.

Sin duda queda un camino largo que recorrer en el área de la Robótica Educativa, estamos trabajando para facilitarte este trayecto, y sobre todo que lo puedas poner en práctica. Continúa con el módulo 2.

Bloque 2

Tema I. Algoritmos y estructuras de programación

Algoritmos

De acuerdo al Diccionario de la Lengua Española (DLE) algoritmo se define de la siguiente manera:

1. m. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
2. m. Método y notación en las distintas formas del cálculo.

Algoritmo. En Matemática, ciencias de la Computación y disciplinas relacionadas, un algoritmo (del latín, dixit algorithmus y éste a su vez del matemático persa Al Juarismi) es un conjunto reescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien lo ejecute. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución. Los algoritmos son objeto de estudio de la algoritmia.

¿Qué es un algoritmo?

Un algoritmo es un conjunto de instrucciones detalladas paso a paso o una fórmula para resolver un problema o completar una tarea. En programación, los programadores escriben algoritmos que indica a la computadora cómo realizar una tarea.

Cuando se piensa en un algoritmo en su forma más general (no en términos de programación), éstos están por todas partes. Una receta para cocinar es un algoritmo, el método que se usa para resolver una suma o una división grande es un algoritmo, y el proceso de doblar una camisa o unos pantalones es un algoritmo. ¡Incluso nuestra rutina matinal se puede considerar un algoritmo!

Los beneficios del pensamiento algorítmico

El pensamiento algorítmico, o la habilidad de definir claramente los pasos para resolver un problema, es crucial en materias como las Matemáticas o las Ciencias. Utilizamos algoritmos todo el tiempo sin darnos cuenta, especialmente en Matemáticas. Para resolver un problema de división de gran tamaño, se aplican algoritmos que hemos aprendido para iterar a través de los dígitos del número que están dividiendo. Para cada dígito del dividendo (el número a dividir), se debe dividir, multiplicar y restar. El pensamiento algorítmico permite descomponer los problemas y conceptualizar soluciones en términos de pasos discretos de un procedimiento. (Edukative, 2016)

Características de los algoritmos

Las características fundamentales que debe cumplir todo algoritmo son:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar bien definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos.

La definición de un algoritmo debe describir tres partes: Entrada, Proceso y Salida. Por ejemplo, en el cálculo de la edad de una persona, conociendo su año de nacimiento, la definición del algoritmo quedaría de la siguiente manera:

- Entrada: la edad de la persona, información del año de nacimiento y el actual.
- Proceso: realizar la diferencia del año actual menos el año de nacimiento.
- Salida: visualización del resultado generado. Es decir, el resultado es la edad.

Tipos de Algoritmos

Dentro de los diferentes tipos de algoritmos que existen, podemos mencionar los siguientes:

Algoritmo Secuencial:

La estructura secuencial es aquella en la que una acción (instrucción) sigue otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso.

Algoritmo Selectivo

Estas estructuras se utilizan para TOMAR DECISIONES (llaman estructuras de decisión o alternativas). Lo que se hace es EVALUAR una condición, y, a continuación, en función del resultado, se lleva a cabo una opción u otra.

Algoritmo Repetición

Permite repetir varias veces un grupo de pasos, hasta que se satisfaga esta condición. La repetición puede programarse para un cierto número de veces.

Ejemplos de algoritmos cotidianos

Secuencial

Algoritmo para ir al cine

1. Ir al cine
2. Escoger la película
3. Hacer la fila
4. Pagar boleto
5. Ir a la entrada de las salas
6. Entregar el boleto
7. Buscar mi lugar
8. Sentarse
9. Ver la película

Cuando vas a la escuela por la mañana

1. Apagas el despertador.
2. Te tomas 5 minutos más.
3. Te despiertas.
4. Te das cuenta que no fueron 5 minutos, si no 20.
5. Te despiertas como rayo.
6. Te metes a bañar rápido.
7. Te secas.
8. Sales del baño
9. Buscas tu ropa.
10. Te vistes.
11. Sales de tu casa corriendo.
12. En el camino recuerdas que hay tarea.
13. Dices “En la escuela la hago”
14. Listo

Selectivo

1. Verificar cartelera

Repetición

1. Verificar cartelera
2. *Si exhiben la película que quiero ver,*

2. Si exhiben la película que quiero ver,

- 2.1. ir a paso 3 y continuar,**
2.2. si no, finaliza algoritmo.

3. Ir al cine
4. Escoger la película
5. Hacer la fila
6. Pagar boleto
7. Ir a la entrada de las salas
8. Entregar el boleto
9. Buscar mi lugar
10. Sentarse
11. Ver la película

- 2.1. ir a paso 3 y continuar,*
2.2. si no, finaliza algoritmo.

3. Ir al cine
4. Escoger la película
5. Hacer la fila
6. Pagar boleto
7. Ir a la entrada de las salas
8. Entregar el boleto

9. Repetir hasta encontrar

9.1. Buscar mi lugar

10. Sentarse
11. Ver la película

Estructuras de programación

De manera general podemos decir que las estructuras de programación son parte fundamental de cualquier lenguaje. Sin ellas, las instrucciones de un programa solo podrían ejecutarse en el orden en que están escritas (orden secuencial). Las estructuras de control o de programación permiten modificar este orden. Podemos decir que estas estructuras de programación son:

- Condicionales o bifurcaciones: permiten que se ejecuten conjuntos distintos de instrucciones, en función de que se verifique o no determinada condición.
- Bucles o repeticiones: permiten que se ejecute repetidamente un conjunto de instrucciones, bien un número pre-determinado de veces, o bien hasta que se verifique una determinada condición.

Estructuras selectivas

La especificación formal de algoritmos tiene realmente utilidad cuando el algoritmo requiere una descripción más complicada que una lista sencilla de instrucciones. Este es el caso cuando existen un número de posibles alternativas resultantes de la evaluación de una determinada condición. Las estructuras selectivas se utilizan para tomar decisiones lógicas; de ahí que se suelen denominar también estructuras de decisión o alternativas.

En las estructuras selectivas se evalúa una condición y en función del resultado de la misma se realiza una opción u otra. Las condiciones se especifican usando expresiones lógicas. La representación de una estructura selectiva se hace con palabras en pseudocódigo (if, then, else; en español si, entonces, si_no), con una figura geométrica en forma de rombo o bien con un triángulo en el interior de una caja rectangular. Las estructuras selectivas o alternativas pueden ser:

- simples,
- dobles,
- múltiples.

Alternativa simple (si-entonces / if-then)

La estructura alternativa simple si-entonces (en inglés if-then) ejecuta una determinada acción cuando se cumple una determinada condición. La selección si-entonces evalúa la condición y

- si la condición es verdadera, entonces ejecuta la acción S1 (o acciones caso de ser S1 una acción compuesta y constar de varias acciones),
- si la condición es falsa, entonces no hacer nada.

si (condición) entonces

acciones

fin-si

Alternativa doble (si-entonces-sino / if-then-else)

La estructura anterior es muy limitada y normalmente se necesitará una estructura que permita elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de una determinada condición. Si la condición C es verdadera, se ejecuta la acción S1 y, si es falsa, se ejecuta la acción S2

si (condición) entonces

acción S01

acción S02

acción Snn

si _ no

acción N01

acción N02

acción Nnn

fin _ si

Estructuras repetitivas

Las computadoras están especialmente diseñadas para todas aquellas aplicaciones en las cuales una operación o conjunto de ellas deben repetirse muchas veces. Un tipo muy importante de estructura es el algoritmo necesario para repetir una o varias acciones un número determinado de veces. Un programa que lee una lista de números puede repetir la misma secuencia de mensajes al usuario e instrucciones de lectura hasta que todos los números de un fichero se lean.

Las estructuras que repiten una secuencia de instrucciones un número determinado de veces se denominan bucles y se denomina iteración al hecho de repetir la ejecución de una secuencia de acciones.

Un ejemplo aclarará la cuestión. Supongamos que se desea sumar una lista de números escritos desde teclado —por ejemplo, calificaciones de los alumnos de una clase—. El medio conocido hasta ahora es leer los números y añadir sus valores a una variable SUMA que contenga las sucesivas sumas parciales. La variable SUMA se hace igual a cero y a continuación se incrementa en el valor del número cada vez que uno de ellos se lea. El algoritmo que resuelve este problema es:

```
algoritmo suma
inicio
    SUMA = 0
    leer(numero)
    SUMA = SUMA + numero
    leer(numero)
    SUMA = SUMA + numero leer(numero)
fin
```

y así sucesivamente para cada número de la lista. En otras palabras, el algoritmo repite muchas veces las acciones.

```
leer(numero)
SUMA = SUMA + numero
```

Tales opciones repetidas se denominan bucles o lazos. La acción (o acciones) que se repite en un bucle se denomina iteración. Las dos principales preguntas a realizarse en el diseño de un bucle son ¿qué contiene el bucle? y ¿cuántas veces se debe repetir?

Cuando se utiliza un bucle para sumar una lista de números, se necesita saber cuántos números se han de sumar. Para ello necesitaremos conocer algún medio para detener el bucle. En el ejemplo anterior usaremos la técnica de solicitar al usuario el número que desea, por ejemplo, N. Existen dos procedimientos para contar el número de iteraciones, usar una variable TOTAL que se inicializa a la cantidad de números que se desea y a continuación se decrementa en uno

cada vez que el bucle se repite (este procedimiento añade una acción más al cuerpo del bucle: $TOTAL = TOTAL - 1$), o bien inicializar la variable TOTAL en 0 o en 1 e ir incrementando en uno a cada iteración hasta llegar al número deseado.

algoritmo suma_numero

```
var
    entero : N, TOTAL
    real : NUMERO, SUMA
inicio
    leer(N)
    TOTAL = N
    SUMA = 0
    mientras TOTAL > 0 hacer
        leer(NUMERO)
        SUMA = SUMA + NUMERO
        TOTAL = TOTAL - 1
    fin_mientras
    escribir('La suma de los', N, 'números es', SUMA)
fin
```

Tipos de estructuras repetitivas

Dentro de los tipos de estructuras repetitivas puedes encontrar las siguientes:

Estructura mientras ("while")

La estructura repetitiva mientras (en inglés while o dowhile: hacer mientras) es aquella en que el cuerpo del bucle se repite mientras se cumple una determinada condición. Cuando se ejecuta la instrucción mientras, la primera cosa que sucede es que se evalúa la. Si se evalúa falsa, no se toma ninguna acción y el programa prosigue en la siguiente instrucción del bucle. Si la expresión booleana es verdadera, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión booleana. Este proceso se repite una y otra vez mientras la condición sea verdadera.

Estructura hacer-mientras ("do-while")

El bucle hacer-mientras evalúa la expresión al comienzo del bucle de repetición; siempre se utiliza para crear bucle pre-test. Los bucles pre-test se denominan también bucles controlados por la entrada. En numerosas ocasiones se necesita que el conjunto de sentencias que

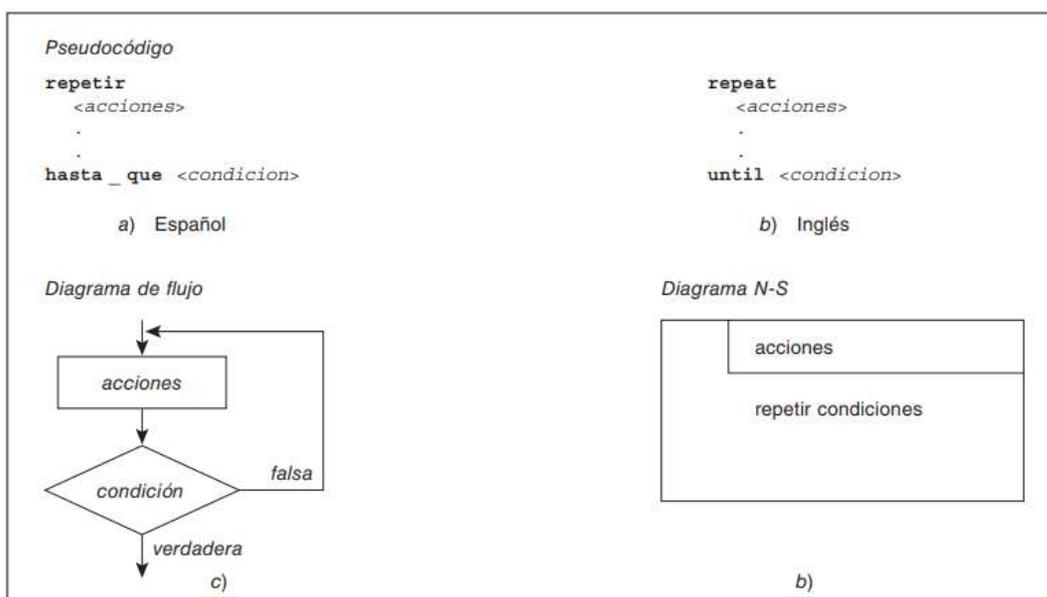
componen el cuerpo del bucle se ejecute al menos una vez sea cual sea el valor de la expresión o condición de evaluación. Estos bucles se denominan bucles post-test o bucles controlados por la salida.

Un caso típico es el bucle hacer-mientras (do-while) existente en lenguajes como C/C++, Java o C#. El bucle hacer-mientras es análogo al bucle mientras y el cuerpo del bucle se ejecuta una y otra vez mientras la condición sea verdadera. Existe, sin embargo, una gran diferencia y es que el cuerpo del bucle está encerrado entre las palabras reservadas hacer y mientras, de modo que las sentencias de dicho cuerpo se ejecutan, al menos una vez, antes de que se evalúe la expresión. En otras palabras, el cuerpo del bucle siempre se ejecuta, al menos una vez, incluso aunque la expresión booleana sea falsa.

Estructura repetir ("repeat")

Existen muchas situaciones en las que se desea que un bucle se ejecute al menos una vez antes de comprobar la condición de repetición. En la estructura mientras si el valor de la expresión booleana es inicialmente falso, el cuerpo del bucle no se ejecutará; por ello, se necesitan otros tipos de estructuras repetitivas.

La estructura repetir (repeat) se ejecuta hasta que se cumpla una condición determinada que se comprueba al final del bucle.



El bucle repetir-hasta_que se repite mientras el valor de la expresión booleana de la condición sea falsa, justo la opuesta de la sentencia mientras.

algoritmo repetir

```
var
real : numero
entero: contador
inicio
contador = 1
repetir
    leer(numero)
    contador = contador + 1
    hasta_que contador > 30
    escribir('Numeros leidos 30')
fin
```

En el ejemplo anterior el bucle se repite hasta que el valor de la variable contador exceda a 30, lo que sucederá después de 30 ejecuciones del cuerpo del bucle.

Diferencias de las estructuras mientras y repetir

- La estructura mientras termina cuando la condición es falsa, mientras que repetir termina cuando la condición es verdadera.
- En la estructura repetir el cuerpo del bucle se ejecuta siempre al menos una vez; por el contrario, mientras es más general y permite la posibilidad de que el bucle pueda no ser ejecutado. Para usar la estructura repetir debe estar seguro de que el cuerpo del bucle —bajo cualquier circunstancia— se repetirá al menos una vez. (Joyanes Aguilar, 2008)

Quizá suene un poco abrumador en un principio leer los tipos de estructuras selectivas y de repetición, en este curso trabajarás con las instrucciones IF (selectivas) y REPEAT (repetición), sin embargo es necesario que conozcas la un poco más acerca de este tema.

Actividad. Elaboración de un algoritmo cotidiano.

Esta actividad es muy sencilla, ya que debes elaborar un algoritmo en el que describas cuáles son los pasos para realizar alguna actividad cotidiana como preparar un platillo, lavar el coche, etc.

¿Qué debes incluir en tu algoritmo?

Debes incluir al menos uno de los dos tipos de estructuras que verás en este curso, selectivas (Si-If) y/o repetición (repeat). Estas complementadas con los pasos necesarios para llevar a cabo la tarea o actividad que describirás en el algoritmo.

Instrucciones para realizar la actividad.

1. Abre un documento nuevo de texto (Word).
2. Analiza y reflexiona sobre la actividad que describirás en tu algoritmo.
3. Describe paso a paso la secuencia en la que realizas la actividad. Recuerda que tu algoritmo debe incluir al menos una de los dos tipos de estructuras con las que trabajarás en este curso.
4. Una vez terminado el algoritmo, guarda el documento de texto de la siguiente manera: Actividadn+InicialesNombre, por ejemplo “actividadnnJGAG”.
5. Cierra el documento y envíalo a través de la plataforma para su revisión.

Recomendaciones:



Realiza una prueba de escritorio para verificar que tu algoritmo indique e incluya los pasos que deseas agregar

Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Tips:

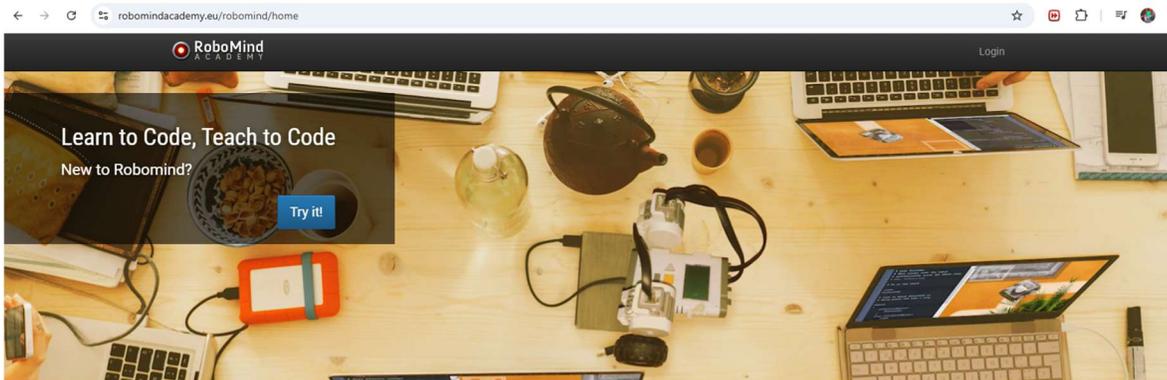


Las pruebas de escritorio son simulaciones del comportamiento de un algoritmo que permiten determinar la validez del mismo. Permiten detectar errores, omisiones o mejorar el algoritmo.

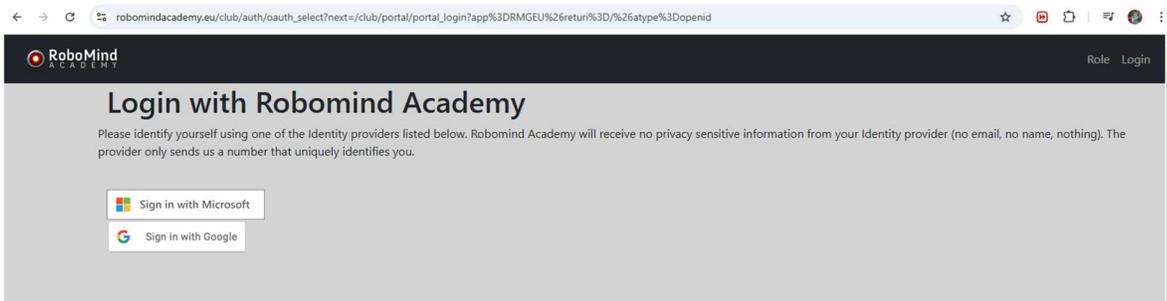
B2 Módulo II. Ciclos y condiciones con Robomind

Ingreso a la página Robomind Academy

Es hora de continuar resolviendo retos en la página de Robomind Academy, para ello abre tu navegador y en la barra de direcciones escribe la siguiente dirección: <https://www.robomindacademy.eu/robomind/home>, posteriormente da clic en “Login” para iniciar sesión.



Iniciar sesión, escribe el correo que utilizaste para crear tu cuenta en Robomind.

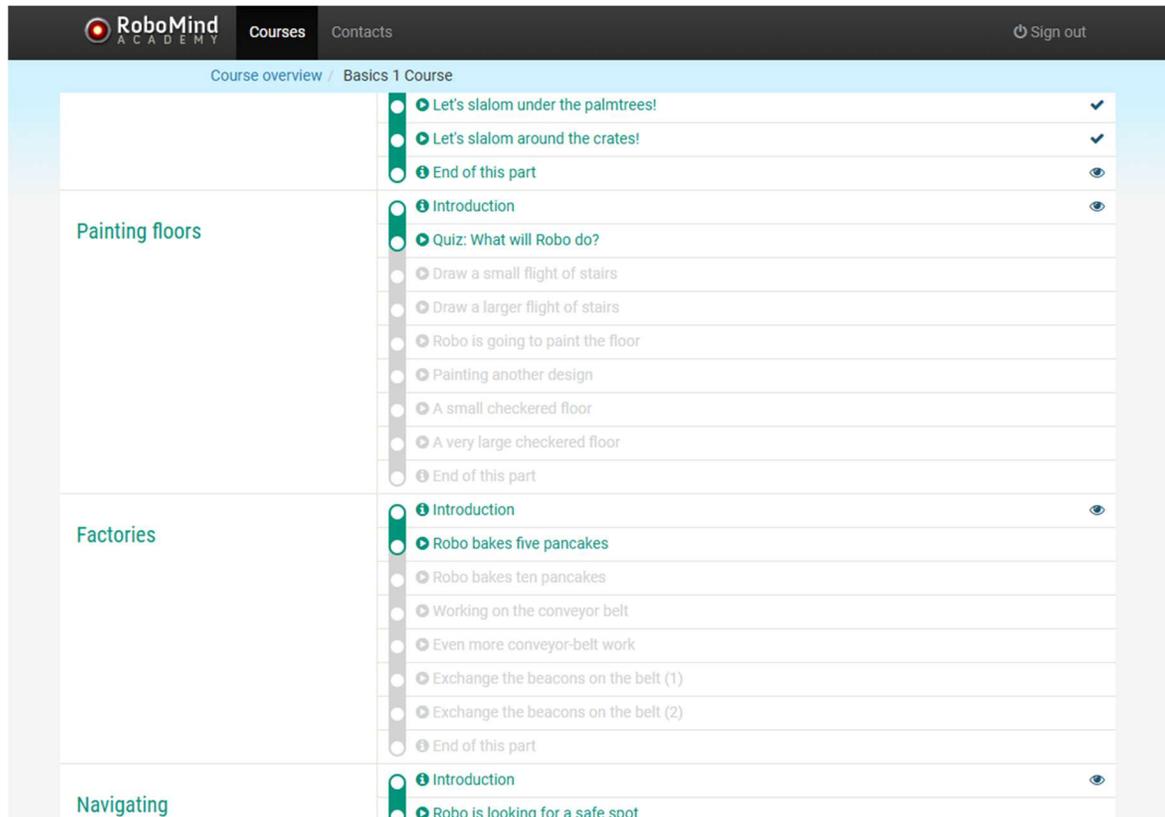


Una vez que inicias sesión, da clic en “Courses” y ubica el curso “Basics 1 Course”, da clic en “Start”. Al dar clic en este curso encontrarás las siguientes categorías o secciones:

- Getting started!
- Guarding and slalom
- Painting floors
- Factories
- Navigating
- Tracking and tracing

- Line following

Esto lo puedes ver en la imagen siguiente:



Bucles / Ciclos

Para poner en práctica la teoría vista en el tema de ciclos y bucles, trabajarás en algunas de las secciones con retos que te permitan fortalecer estos temas.

El primer tema que pondrás en práctica es el de ciclos o repeticiones, para ello selecciona la categoría o lección denominado “Factories” (fábricas). En esta parte del Bloque se simulan algunos de los procesos repetitivos que se pueden llevar a cabo en algunas fábricas, las cuales en muchos casos ya son realizadas por robots.

Para iniciar da ubica el curso
“Factories” y da clic en
“Introduction”

Factories

- Introduction
- Robo bakes five pancakes
- Robo bakes ten pancakes
- Working on the conveyor belt
- Even more conveyor-belt work
- Exchange the beacons on the belt (1)
- Exchange the beacons on the belt (2)
- End of this part

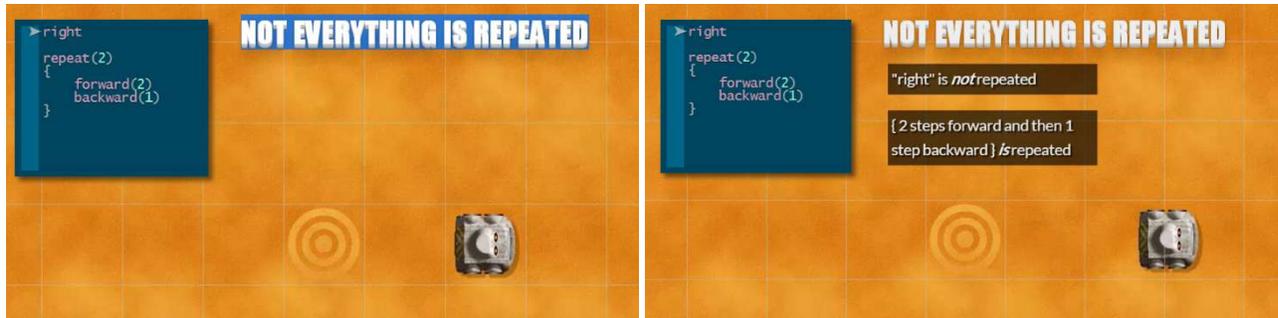
- Introduction
- Robo is looking for a safe spot
- Robo is looking for another safe spot
- Robo is looking for the beacon
- Robo is looking for the beacon again
- Another path ...
- Dis yourself through the beacons

Navigating

Como puedes ver el curso indica que se continúa trabajando con la instrucción “repeat” (repetir). Explora la introducción dando clic en el botón siguiente >.



En la información proporcionada en la información podrás encontrar que no todas las instrucciones es necesario repetirlas. Esto es importante que lo conozcas, ya que en algunos retos tendrás que determinar que instrucciones son las que deben de ir dentro de un ciclo.



En la siguiente imagen puedes encontrar otro ejemplo, donde se utilizan dos ciclos, y entre estos dos ciclos una instrucción (forward), que no se repite.

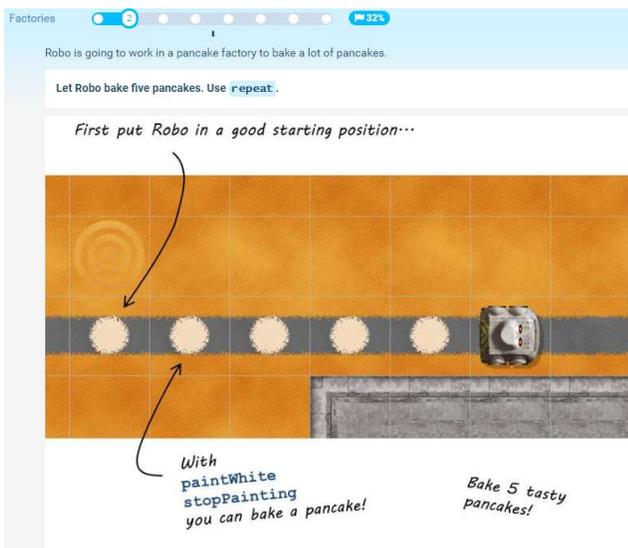
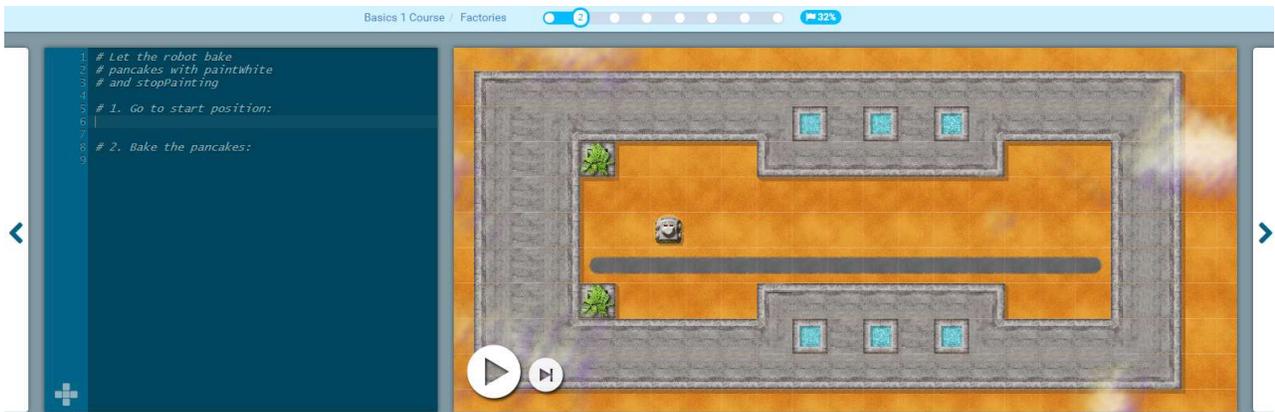


Finalmente podrás ver un video donde en el que verás algunos de los procesos repetitivos que puede realizar un robot.



Reto 2 “Robo bakes five pancakes” (Robo hornea cinco panqueques)

El objetivo de este reto es que Robo debe “hornear” cinco panqueques.



Como ya se mencionó, el objetivo de este reto es que Robo hornee cinco panqueques, pero antes debes llevarlo a la posición de inicio y que empiece a hornear los panqueques.

Debes determinar que instrucciones serán las que deben ir dentro de un ciclo y cuáles no.

Para empezar a resolver este reto realizaremos el algoritmo que nos permita desarrollar el programa.

De acuerdo con la posición de Robo y a la posición en la que debe estar para empezar con el proceso de fabricación utilizando la instrucción “repeat”, el algoritmo puede ser el siguiente:

1. Realizar dos giros a la derecha (o izquierda), para quedar frente a la línea negra
2. Avanzar una vez, esto le permite quedar encima de la línea negra.
3. Girar a la izquierda
4. Pintar blanco (hacer el panqueque)
5. Dejar de pintar
6. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)
7. Pintar blanco (hacer el panqueque)
8. Dejar de pintar
9. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)
10. Pintar blanco (hacer el panqueque)
11. Dejar de pintar
12. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)

13. Pintar blanco (hacer el panqueque)
14. Dejar de pintar
15. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)
16. Pintar blanco (hacer el panqueque)
17. Dejar de pintar
18. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)

Sin embargo, en este código no se ha utilizado la instrucción “repeat”. Para ello es necesario determinar cuáles serán las instrucciones que se repetirán cinco veces, que son los panqueques que debe realizar Robo. De acuerdo a mi lógica estas son las instrucciones que se deben repetir en el ciclo “repeat”

1. Realizar dos giros a la derecha (o izquierda), para quedar frente a la línea negra
2. Avanzar una vez, esto le permite quedar encima de la línea negra.
3. Girar a la izquierda
4. ***Pintar blanco (hacer el panqueque)***
5. ***Dejar de pintar***
6. ***Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)***
7. Pintar blanco (hacer el panqueque)
8. Dejar de pintar
9. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)
10. ***Pintar blanco (hacer el panqueque)***
11. ***Dejar de pintar***
12. ***Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)***
13. Pintar blanco (hacer el panqueque)
14. Dejar de pintar
15. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)
16. ***Pintar blanco (hacer el panqueque)***
17. ***Dejar de pintar***
18. ***Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)***

El algoritmo quedaría de la siguiente manera utilizando la instrucción “repeat”:

1. Realizar dos giros a la derecha (o izquierda), para quedar frente a la línea negra
1. Avanzar una vez, esto le permite quedar encima de la línea negra.
2. Girar a la izquierda
3. Repetir cinco veces

- 3.1. Pintar blanco (hacer el panqueque)
- 3.2. Dejar de pintar
- 3.3. Avanzar (se mueve a la siguiente posición para elaborar otro panqueque)

Cómo puedes ver el algoritmo se reduce considerablemente, por lo consiguiente las líneas de código o instrucciones que contenga el programa serán menos.

Ahora desarrollaremos el código para resolver este reto.

```
derecha(2)
adelante
izquierda
repetir(5)
{
    pintarBlanco
    detenerPintar
    adelante
}
```

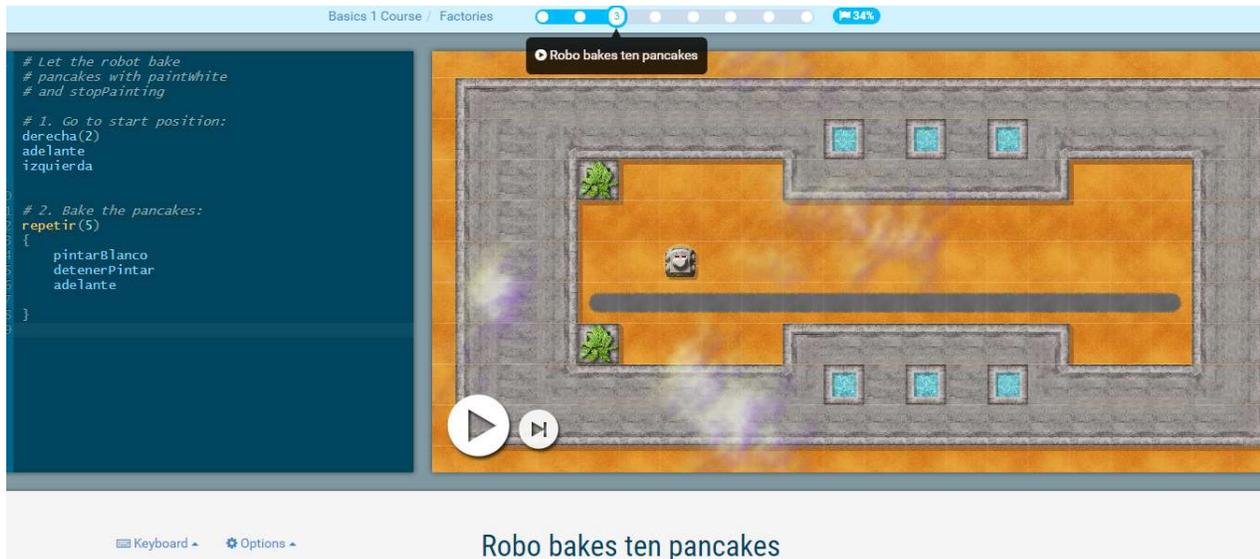
```
1 # Let the robot bake
2 # pancakes with paintWhite
3 # and stopPainting
4
5 # 1. Go to start position:
6 derecha(2)
7 adelante
8 izquierda
9
10
11 # 2. Bake the pancakes:
12 repetir(5)
13 {
14 pintarBlanco
15 detenerPintar
16 adelante
17 }
18
19
```

Probemos el código, recuerda que, si el código desarrollado cumple con el reto, Robo al final realiza un giro completo, además podrás avanzar al siguiente reto.

Resuelto el reto puedes avanzar al siguiente, da clic en el reto 3 de esta Bloque. El objetivo es que Robo realice diez panqueques. La aplicación automáticamente carga el código que elaboraste en el reto anterior, por lo tanto, sólo es necesario modificar un valor para que resuelvas el reto.

Actividad. Resolver reto tres.

Reto 3 “Robo bakes ten pancakes” (Robo hornea diez panqueques)



Instrucciones para resolver la actividad

1. Este reto requiere que Robo realice diez panqueques. Para ello identifica cuál es el valor que debes modificar para que se realice esta acción.
2. Una vez que hayas resuelto el reto, realiza una impresión de pantalla donde se pueda apreciar el reto resuelto
3. Abre un documento nuevo de texto (Word) y copia la impresión de pantalla.
4. Guarda el archivo con el nombre “Actividad02” más las iniciales de tu nombre, por ejemplo “Actividad02JGAG”.
5. Para finalizar envía el documento a través de la plataforma para su revisión.

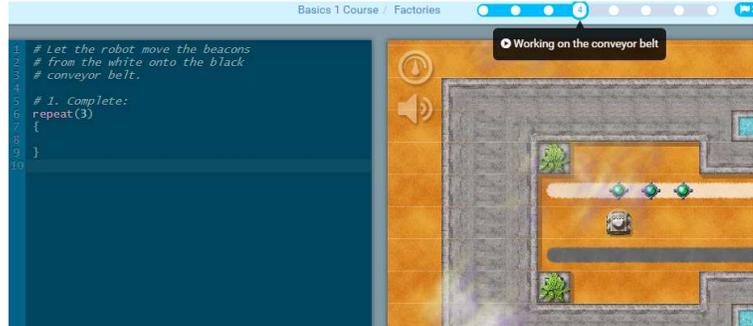
Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Reto 4. “Working on the conveyor belt” (Trabajando en la cinta transportadora)

El objetivo de este reto es que Robo tome las tres balizas que se encuentran en la línea blanca, una a la vez, y las coloque de la misma manera en la línea negra.



Como puedes ver, el reto ya incluye algunas instrucciones que debes utilizar, en este caso es la instrucción del ciclo “repeat (3)”.

Resolviendo el reto

Analizando el planteamiento del reto, y el mapa que se plantea en él, verás que hay tres “balizas”, por lo tanto, se usa en la instrucción “repeat” tres veces. Tomando como referencia esta base, elaboraremos el algoritmo para posteriormente elaborar el programa.

Algoritmo

1. Repetir tres veces
 - 1.1. Tomar la baliza
 - 1.2. Girar a la derecha (o izquierda) dos veces
 - 1.3. Soltar baliza
 - 1.4. Girar a la izquierda
 - 1.5. Avanzar
 - 1.6. Girar a la izquierda

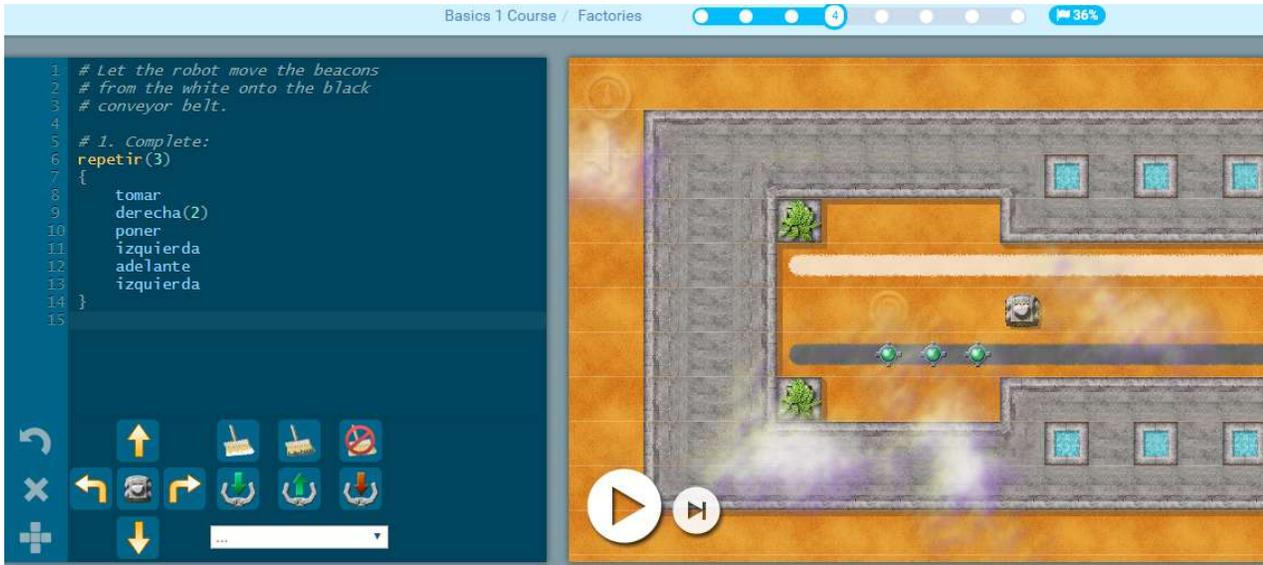
Código

```

4
5 # 1. Complete:
6 repetir(3)
7 {
8     tomar
9     derecha
10    derecha
11    poner
12    izquierda
13    adelante
14    izquierda
15
16 }

```

Escribe el código anterior y prueba el programa. Como puedes ver, Robo pasa las tres balizas a la “cinta” negra.



Una vez que hayas comprobado que el código funciona y podrás pasar al siguiente reto.

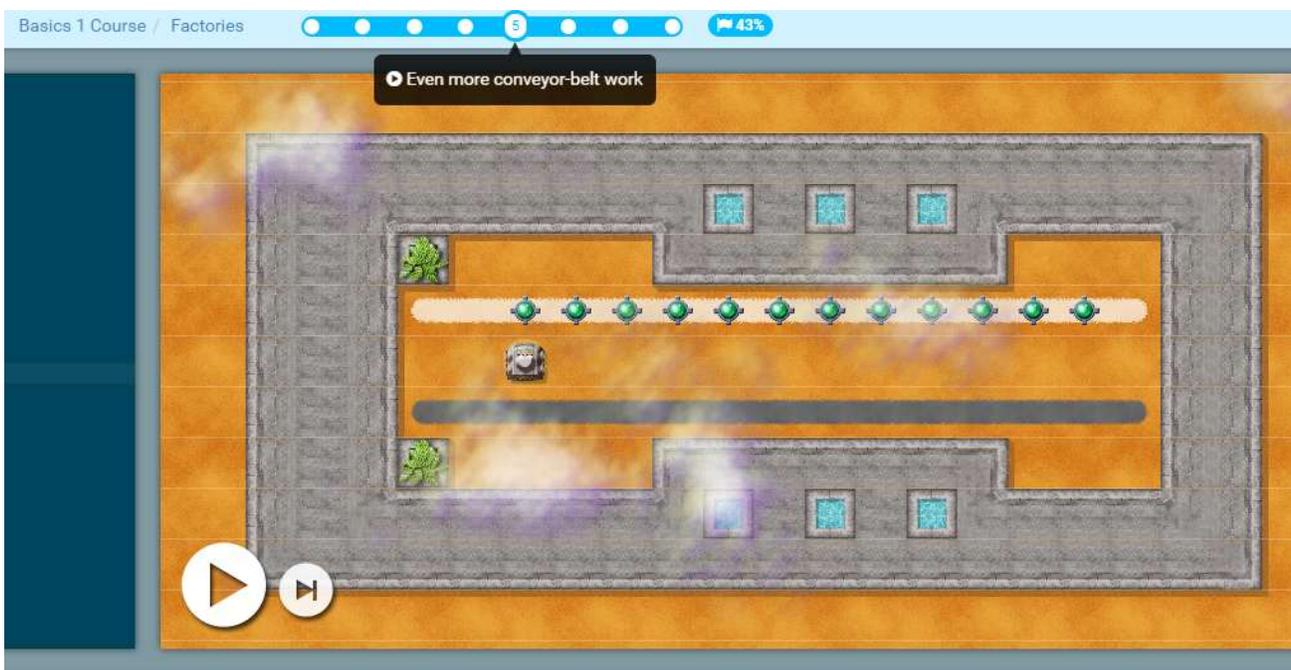
Actividad. Resolver retos cinco, seis y siete.

Resolver los retos:

- Cinco “Even more conveyor-belt work”. Objetivo de este reto es mover las balizas colocadas en la “cinta” blanca y moverlas a la “cinta” negra.
- Seis “Exchange the beacons on the belt (1)”
- Siete “Exchange the beacons on the belt (2)”

Reto cinco. “Even more conveyor-belt work” (Aún más trabajo de cinta transportadora)

Objetivo: Robo debe pasar todas las balizas que se encuentran en la “banda” blanca a la “banda” negra.



Después de resolver el reto cuatro, selecciona el reto cinco.

Una vez que se despliegue la información del reto, podrás ver que este ya incluye código, el cual te servirá como base para resolverlo. Analiza el código del programa y el mapa, después procede a resolverlo.

¿Qué debes incluir en esta actividad?

- Algoritmo
- Código que resuelve el reto

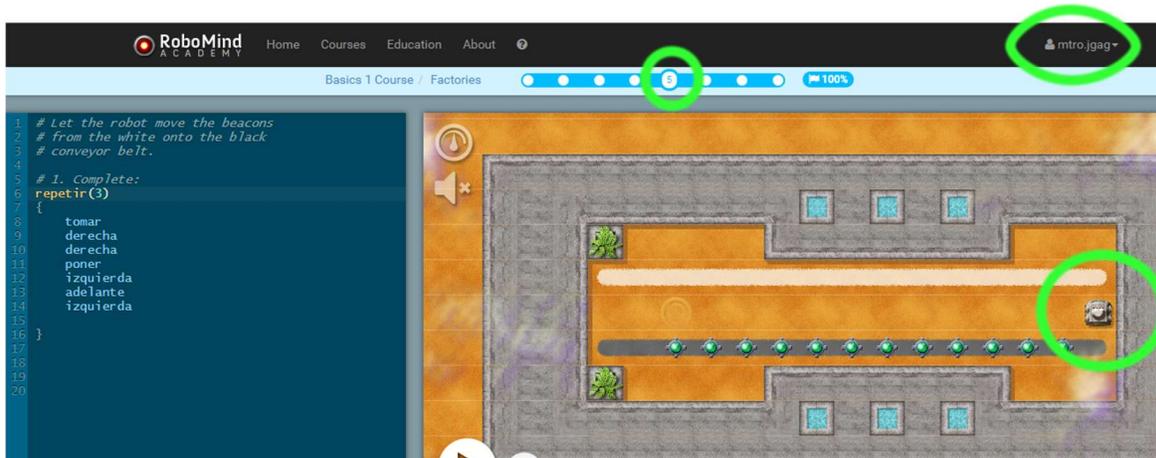
Instrucciones para realizar la actividad

1. Realiza el algoritmo que te permita resolver el reto planteado.
2. Después de realizar el algoritmo, elabora o modifica el programa que se requiere para resolver este reto.
3. Después de solucionarlo, abre un documento nuevo de texto.
4. Copia el algoritmo que usaste para resolver el reto.
5. Realiza una impresión de pantalla, donde se pueda apreciar que el reto está resuelto. Debe apreciarse el código que usaste, así como el mapa del reto.
6. Guarda el documento con el nombre Actividadnn+inicialesTuNombre, por ejemplo “Actividad03JGAG”

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.



Reto seis. “Exchange the beacons on the belt (1)” (Cambie las balizas en la banda(1))



Objetivo: Observa que las balizas se encuentran repartidas en las dos “bandas”, así que Robo debe pasar todas las balizas que se encuentran en la “banda” blanca a la “banda” negra y viceversa.

Después de resolver el reto cinco, selecciona el reto seis.

Una vez que se despliegue la información del reto, podrás ver que este ya incluye código (el del reto 5), el cual te servirá como base para resolverlo. Analiza el código del programa y el mapa, después procede a resolverlo.

¿Qué debes incluir en esta actividad?

- Algoritmo
- Código

Instrucciones para resolver la actividad

1. Realiza el algoritmo que te permita resolver el reto planteado.
2. Después de realizar el algoritmo, elabora o modifica el programa que se requiere para resolver este reto.
3. Después de solucionarlo, abre el documento que creaste en reto anterior.
4. Copia el algoritmo que usaste para resolver el reto.

5. Realiza una impresión de pantalla, donde se pueda apreciar que el reto está resuelto. Debe apreciarse el código que usaste, así como el mapa del reto.
6. Guarda el documento.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Tips para resolver el reto:



- Recuerda que la elaboración del algoritmo te auxiliará a resolver el reto.
- Debes utilizar dos ciclos (repeat), la cantidad de repeticiones depende de la cantidad de balizas que se encuentran en cada “banda”.
- Una vez que termina el primer ciclo, es necesario agregar una instrucción que te permita quedar frente a la otra “banda” para poder empezar a recoger las balizas restantes.



Reto siete. “Exchange the beacons on the belt (2)” (Cambie las balizas en la banda(2))



Objetivo: Observa que las balizas se encuentran alternadas en las dos “bandas”, Robo debe pasar las balizas que se encuentran en la banda blanca a la banda negra y viceversa.

Después de resolver el reto seis, selecciona el reto siete.

Una vez que se despliegue la información del reto, podrás ver que este ya incluye código (reto 6), el cual te servirá como base para resolverlo. Analiza el código del programa y el mapa, después procede a resolverlo.

¿Qué debes incluir en esta actividad?

- Algoritmo
- Código

Instrucciones para resolver la actividad

1. Realiza el algoritmo que te permita resolver el reto planteado.
2. Después de realizar el algoritmo, elabora o modifica el programa que se requiere para resolver este reto.
3. Después de solucionarlo, abre el documento que creaste en reto anterior

4. Copia el algoritmo que usaste para resolver el reto.
5. Realiza una impresión de pantalla, donde se pueda apreciar que el reto está resuelto. Debe apreciarse el código que usaste, así como el mapa del reto.
6. Una vez que resuelvas el reto, podrás avanzar al “reto” ocho, que en realidad es la medalla que confirma que concluiste satisfactoriamente el curso “Factories”. Realiza una impresión de pantalla de y cópiala en este documento.
7. Guarda el documento.
8. Envíalo a través de la plataforma para su revisión.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Tips para resolver el reto:



- Recuerda que la elaboración del algoritmo te auxiliará a resolver el reto.
- Puedes utilizar uno o dos ciclos (repeat) para resolver el reto.
- Si utilizas un ciclo, Robo debe recoger y mover alternadamente las balizas. Si utilizas dos ciclos, debes agregar entre ellos instrucciones o comandos que permitan a Robo ubicarse en la posición adecuada para que Robo realice el siguiente ciclo.

Reto resuelto:



Medalla de la lección:

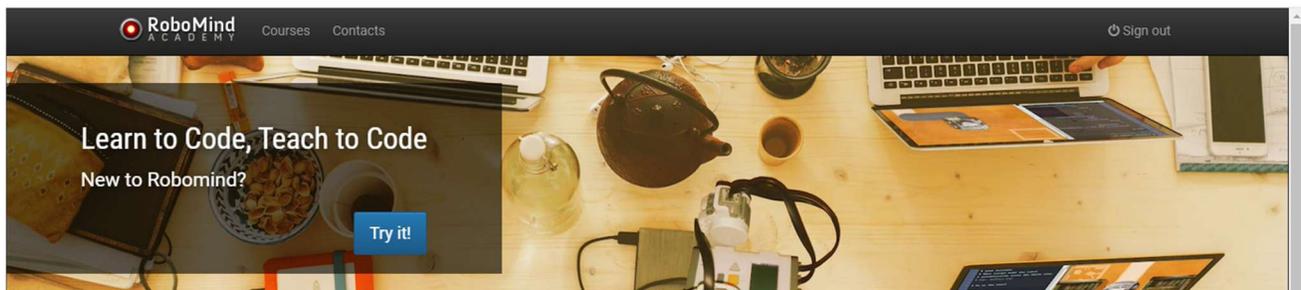


¿Cómo toma decisiones Robo?

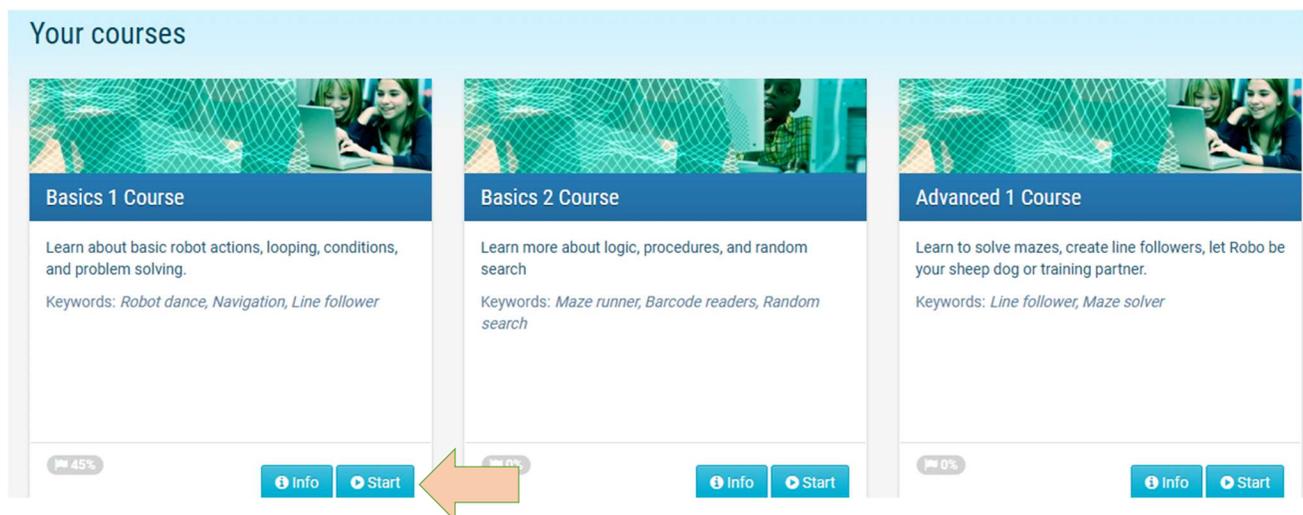
Para continuar practicando con la programación para robots, ahora trabajarás con la categoría denominado “Line following” del curso “Basics 1 Course”.

Para seleccionar la siguiente etapa, verifica que tu sesión este abierta en RoboMind Academy, de lo contrario, inicia sesión.

Una vez dentro de Robomind Academy da clic en el menu “Courses”.



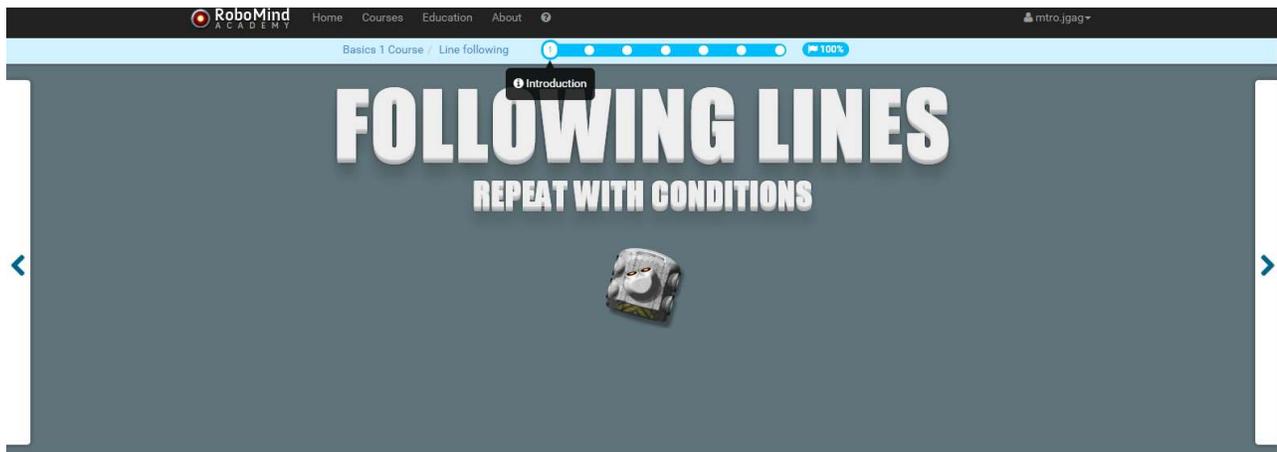
Posteriormente ubica y da clic en “Basics 1 Course” y da clic en el botón “Start”



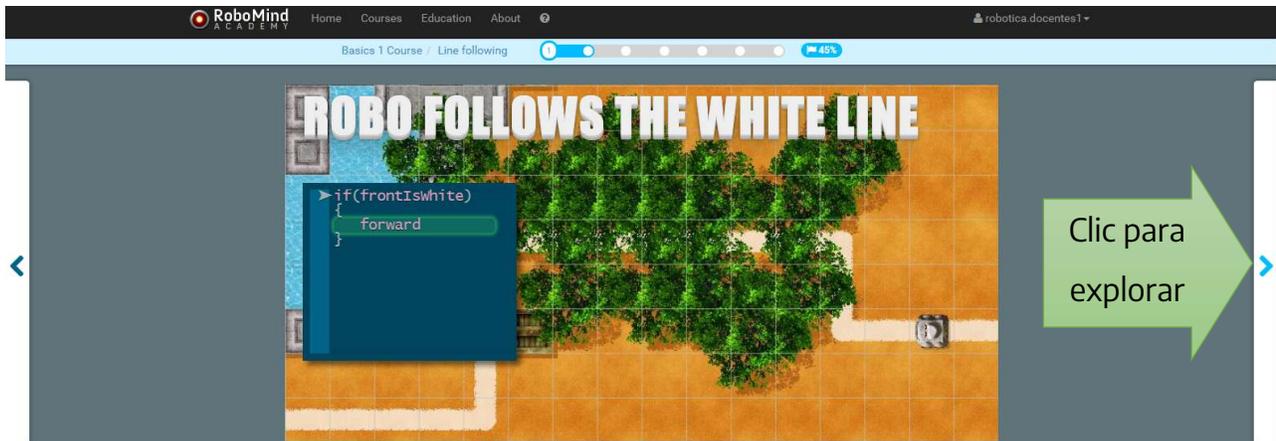
Posteriormente ubica la lección “Line following” (seguidor de línea) y da clic en “Introduction”, esta etapa se encuentra al final del “Basics 1 Course”.

Course overview / Basics 1 Course	
	<ul style="list-style-type: none"> End of this part
Tracking and tracing	<ul style="list-style-type: none"> Introduction Robo is going to track and trace Robo is going to track and trace (2) Robo is going to track and trace (3) Robo is going to track and trace (4) Robo is going to track and trace (5) End of this part
	<ul style="list-style-type: none"> Introduction ← Robo follows a short line Robo follows a longer line Robo follows a very long line! Get the beacon back Get the beacon back (2) What else can you do?

Introducción.

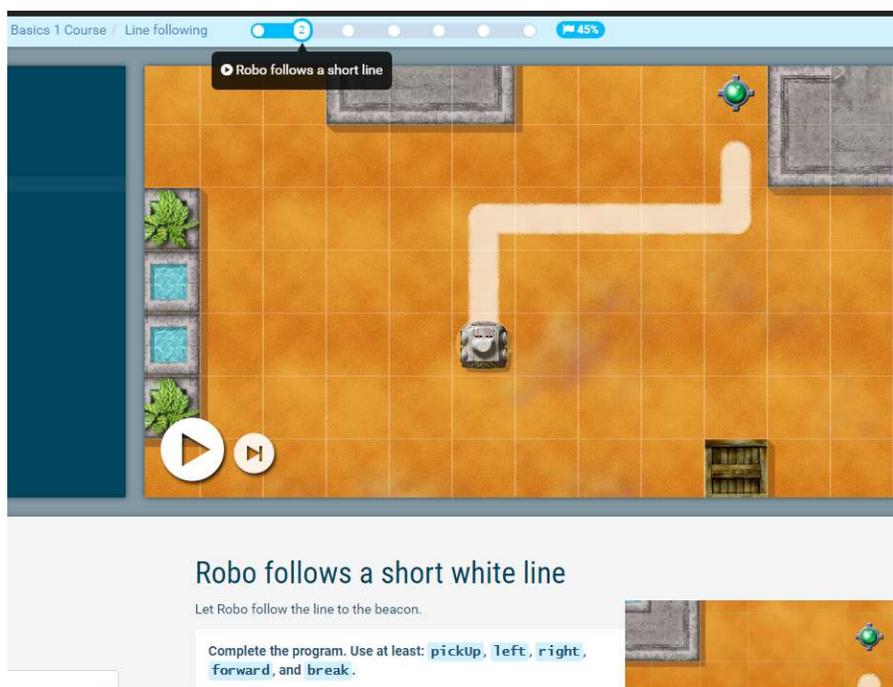


Explora la introducción, ya que en ella encontrarás información que te será útil para resolver los retos.



Una vez que hayas finalizado de ver la introducción, puedes avanzar al reto número dos. El objetivo de este reto es que sigas la línea blanca hasta llegar a la baliza y recogerla.

Se pide que utilices los comandos o instrucciones: pickUp, left, right, forward y break. Además de los comandos de condiciones “If”, los cuales se mencionan en la introducción.



Resolviendo el reto dos “Robo follows a short line” (Robo sigue una línea corta).

Como puedes ver en la imagen anterior, Robo debe seguir la línea blanca, llegar al final de ella y recoger la baliza que ahí se encuentra. Para ello utilizaremos la instrucción “IF”, que servirá o indicará a Robo que es lo que debe de hacer dependiendo si se cumple o no la condición marcada. Todas las instrucciones deben de ir dentro de un ciclo, que terminará cuando se cumpla la condición, que es recoger la baliza al final de la línea.

Para resolver este reto, primero elaboraremos el algoritmo, esto desde mi lógica.

Algoritmo

- ❖ Repetir hasta que llegue a la baliza
- ❖ Preguntar si enfrente es blanco
 - Si es verdadero, avanza
 - Fin (esta instrucción se repite mientras enfrente sea blanco) // Termina la pregunta
- ❖ //cuando enfrente no sea blanco, Robo se preguntará si a la derecha es blanco
- ❖ Preguntar si derecha es blanco
 - Si es verdadero, da vuelta a la derecha y avanza una vez
 - Fin // Termina la pregunta
- ❖ //Cuando no sea blanco enfrente, ni a la derecha Robo se preguntará si a la izquierda es blanco
- ❖ Preguntar si a la izquierda es blanco
 - Si es verdadero, gira a la derecha y avanza
 - Fin // Termina la pregunta
- ❖ //Cuando no sea blanco a la derecha, ni a la izquierda, ni enfrente, Robo se preguntará si enfrente se encuentra la baliza.
- ❖ Preguntar si enfrente está la baliza
 - Si es verdadero, recoger la baliza y salir del ciclo (terminar el programa)
 - Fin // Termina la pregunta

Es hora de codificar el algoritmo.

Como ya te habrás dado cuenta, el reto ya incluye código, el cual completaremos a partir del comentario "# 1. Complete:"

```
# Let the robot follow the
# white line to find the
# beacon and pick it up.
```

```
Repeat
{
# 1. Complete:
}
```

De acuerdo con el algoritmo, el código será el siguiente, antes activa el asistente de código, si es que está deshabilitado.

```
# Let the robot follow the
# white line to find the
# beacon and pick it up.
```

Repeat

```
{
# 1. Complete:
}
```



Posteriormente agrega el código tomando como base el algoritmo desarrollado.

Algoritmo

- ❖ Repetir hasta que llegue a la baliza
- ❖ Preguntar si enfrente es blanco
 - Si es verdadero, avanza

Código

```
# Let the robot follow the
# white line to find the
# beacon and pick it up.
```

Repetir

- Fin (esta instrucción se repite { mientras enfrente sea blanco) // # 1. Complete: Termina la pregunta si(frenteEsBlanco)
- ❖ //cuando enfrente no sea blanco, Robo se { adelante pregunta si a la derecha es blanco }
- ❖ Preguntar si derecha es blanco } si(derechaEsBlanco)
 - Si es verdadero, da vuelta a la derecha y avanza una vez derecha adelante
 - Fin // Termina la pregunta }
- ❖ //Cuando no sea blanco enfrente, ni a la derecha Robo se preguntará si a la izquierda es blanco si(izquierdaEsBlanco)
 - Si es verdadero, gira a la derecha izquierda adelante y avanza }
 - Fin // Termina la pregunta }
- ❖ //Cuando no sea blanco a la derecha, ni a la izquierda, ni enfrente, Robo se preguntará si enfrente se encuentra la baliza. si(frenteEsBaliza)
 - Si es verdadero, recoger la baliza y salir del ciclo (terminar el tomar programa) truncar
 - Fin // Termina la pregunta }

Este sería el resultado que se acaba de crear.



Una vez que se ha resuelto el reto dos, puedes avanzar al reto número tres.



En este reto se incorporan algunas instrucciones las cuáles te mostraré cómo puedes usarlas, recuerda que la lógica o forma de pensar de cada uno es diferente, espero te sirva como sugerencia, comencemos.

Lo primero es agregar la instrucción SI, esto nos servirá para realizar una pregunta, la colocaremos dentro del ciclo Repetir, ya que necesitamos que esto se repita indefinidamente o hasta que se cumpla otra condición, para ello da clic en el combo de instrucciones y selecciono “si()”.

Quedaría de la siguiente manera:

```

repetir
{
  # 1. Complete:
  si()
}
otro
{
}
}

```

Ahora coloco el cursor dentro del paréntesis de “si() y del combo de instrucciones busco el comando “frenteEsBlanco”, es decir, estoy preguntando si frente a Robo hay un color blanco.

Ahora coloco el cursor o puntero dentro de los corchetes, en esta sección colocaré todo lo que quiero que se realice si la instrucción es verdadera.

En este caso solo quiero que se mueva hacia adelante si al frente es claro, entonces doy clic en el icono de la instrucción adelante.

Mi código va así hasta el momento:



Continuando con el programa, necesito agregar otra pregunta, esto es para que en el caso de que no se claro enfrente, tenga una opción más para seguir avanzando. De la misma manera agrego otro “si() con sus respectivos corchetes. Esta instrucción “si” debe ir de igual manera dentro del ciclo repetir.

```

repetir
{
    # 1. Complete:
    si(frenteEsBlanco )
    {
        adelante
    }
}
    
```

Ahora voy a agregar la instrucción para que pregunte si a la derecha es blanco, en el caso de que sea claro a la derecha, entonces hay que indicarle que se mueva para que siga el claro, en ese caso primero tendría que girar a la derecha Robo y luego avanzar, agrego en ese orden las instrucciones correspondientes dentro de los corchetes. El código va como se muestra.

```

repetir
{
    # 1. Complete:
    si(frenteEsBlanco )
    {
        adelante
    }
    si(derechaEsBlanco )
    {
        derecha
        adelante
    }
}
    
```

Ahora necesito preguntar si a la izquierda el blanco, repito el procedimiento, pero ahora con las instrucciones necesarias que determinen que a la izquierda es claro y si lo es, que avance en esa dirección.

```

repetir
{
    # 1. Complete:
    si(frenteEsBlanco )
    {
        adelante
    }
    si(derechaEsBlanco )
    {
        derecha
        adelante
    }
    si(izquierdaEsBlanco )
    {
        izquierda
        adelante
    }
}
    
```

De acuerdo con mi algoritmo, el programa cumple hasta aquí con la parte para avanzar por la línea. Ahora necesito agregar el código que me permita identificar si enfrente está la baliza, la cual se encuentra al final de la línea blanca, y si la encuentra, que la recoja y que termine o salga del programa, el cual es el objetivo del reto

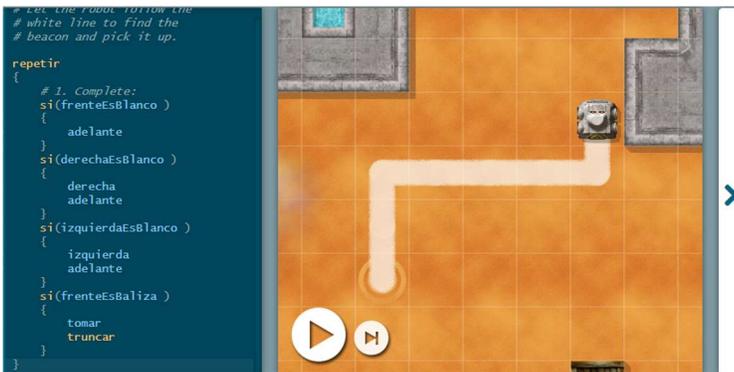
Entonces, agrego otra instrucción “si” y pregunto si en frente se encuentra la baliza, si es así, que la recoja y que interrumpa el programa. El código generado es de la siguiente manera:

Es hora de verificar que el programa desarrollado con base al algoritmo funcione correctamente, damos clic en el botón ejecutar.

El programa desarrollado resuelve el reto, avancemos al siguiente.

```

repetir
{
  # 1. Complete:
  si(frenteEsBlanco )
  {
    adelante
  }
  si(derechaEsBlanco )
  {
    derecha
    adelante
  }
  si(izquierdaEsBlanco )
  {
    izquierda
    adelante
  }
  si(frenteEsBaliza )
  {
    tomar
    truncar
  }
}
    
```



Reto tres “Robo follows a longer line” (Robo sigue una línea larga)

Una vez que ingreses al reto tres “Robo follows a longer line”, podrás ver que el programa que creaste en el reto anterior se ha cargado para este. El objetivo es que Robo recorra una línea más larga con el código o programa que desarrollaste para el reto anterior.



¿Será este código de programa el adecuado para resolver este reto? Es hora de verificarlo, presiona el botón “Play” para comprobarlo. En caso de que no funcione realizaremos los ajustes pertinentes, pero primero probemos el programa.

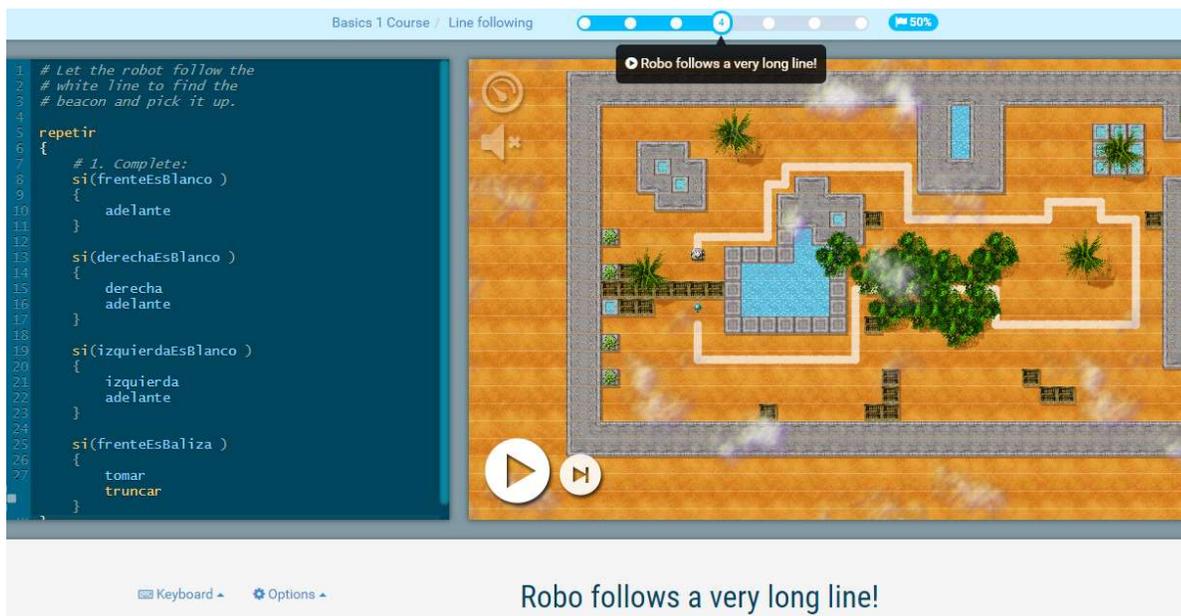


Después de ejecutar el programa, podrás ver que se resuelve el reto sin realizar modificación alguna. Ahora puedes avanzar al siguiente reto.

Reto cuatro “Robo follows a very long line!” (¡Robo sigue una línea muy larga!)

El objetivo de este reto es similar a los anteriores, es decir que Robo siga una línea, llegue al final de la misma y recoja la baliza.

De la misma manera que en el reto anterior, este ya cuenta con un programa, resta ver si con el código que tiene es suficiente para resolverlo, ya que la línea a seguir es más larga y la posición de inicio de Robo es diferente a los retos anteriores.



Ejecuta el programa y verifica si se resuelve el reto.



Después de ejecutar el programa, puedes verificar que se ha resuelto el reto. Esto significa que el programa que se creó en el reto dos sirve para seguir una línea blanca, no importando su longitud o forma del trazo, y que en el momento que encuentre la baliza la recoja.

Actividad. Resolver reto cinco “Get the bacon back”

Reto cinco “Get the bacon back” (Tomar la baliza y llevarla de regreso)

Objetivo: Recorrer la línea, llegar al final, tomar la baliza y llevarla a la posición donde inició el recorrido Robo.

Una vez que accedas al reto, podrás ver que se ha cargado el programa utilizado para los ejercicios anteriores. Analiza el código del programa y el mapa, después procede a resolverlo.

¿Qué debes incluir en esta actividad?

- Algoritmo
- Código

Instrucciones para resolver la actividad

1. Analizar el mapa y código para que elabores el algoritmo que te permita resolver el reto planteado.
2. Después de realizar el algoritmo, crea o modifica el programa que se requiere para resolver este reto.
3. Prueba el programa, si no se resuelve el reto repite los pasos uno y dos. Una vez que lo hayas solucionado, abre un documento nuevo de texto (Word).
4. Copia el algoritmo que usaste para resolver el reto.
5. Realiza una impresión de pantalla, donde se pueda apreciar que el reto está resuelto. Se debe visualizar el código que usaste, así como el mapa del reto.
6. Guarda el archivo de la siguiente manera: actividad04 + tus Iniciales, por ejemplo “actividad04)GAG”.
7. Cierra el documento.
8. Envíalo a través de la plataforma para su revisión.

Recomendaciones:

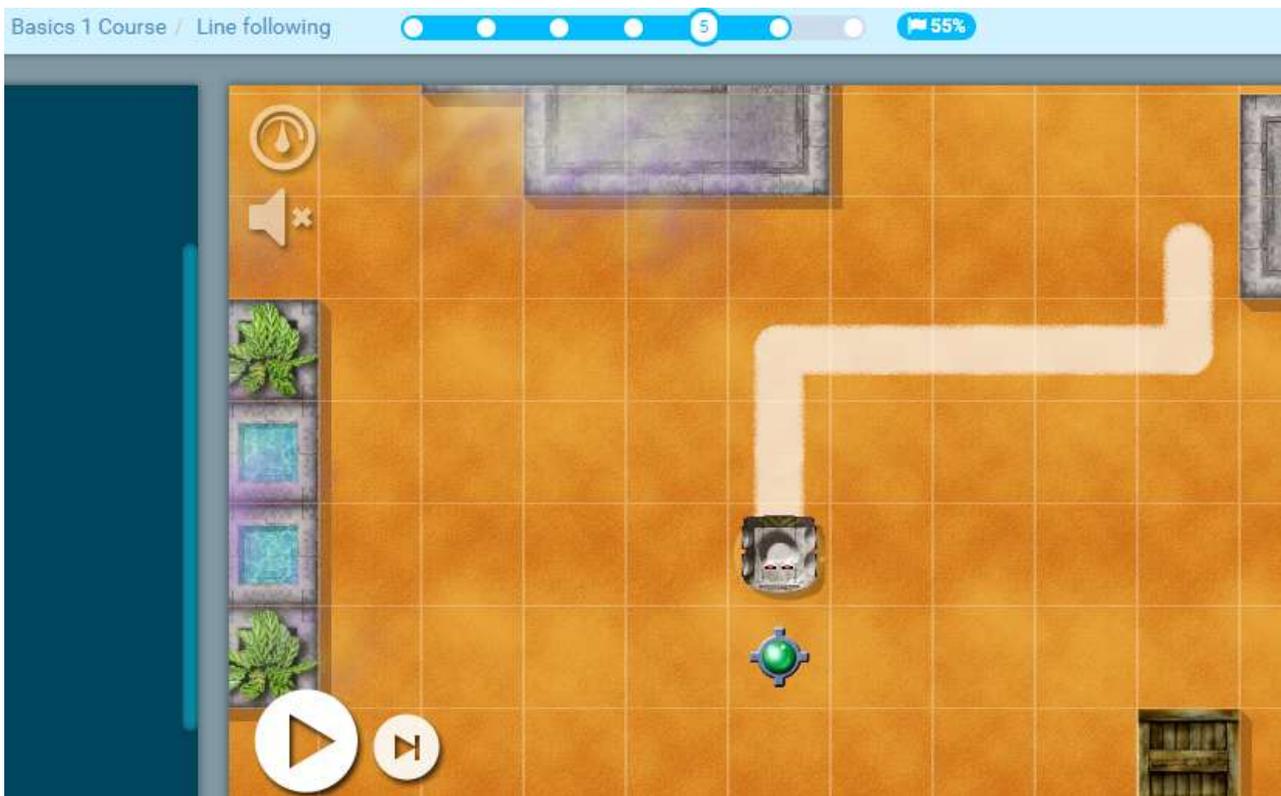


Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

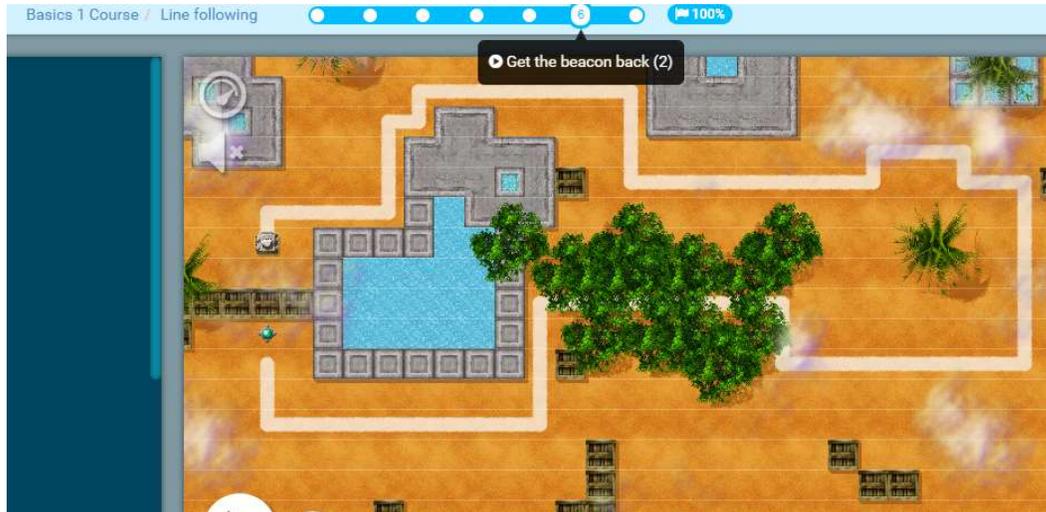
Tips para resolver el reto:



- Recuerda que la elaboración del algoritmo te auxiliará a resolver el reto.
- Debes dejar una marca para que cuando Robo regrese detecte que es el lugar donde inició su recorrido, avanzar para quedar sobre la marca y después dejar la baliza. En ese momento termina el programa.
- Cuando recoja la baliza debe tomar posición para continuar avanzando sobre la línea. Por lo tanto, debes eliminar la instrucción “break” o “truncar” y agregar la o las instrucciones que le permitan retomar el camino y regresar y dejar la baliza donde inició el recorrido.



Actividad. Reto seis “Get the bacon back (2)” (Tomar la baliza y llevarla de regreso (2))



Objetivo: Recorrer la línea, llegar al final, tomar la baliza y llevarla a la posición donde inició el recorrido Robo. El reto es similar al anterior, en este caso Robo inicia el recorrido en otra posición.

Una vez que accedas al reto, podrás ver que se ha cargado el programa utilizado para los retos anteriores. Analiza el código del programa y el mapa, después procede a resolverlo.

¿Qué debes incluir en esta actividad?

- Algoritmo
- Código

Instrucciones para resolver la actividad

1. Analizar el mapa y código para que elabores el algoritmo que te permita resolver el reto planteado.
2. Después de realizar el algoritmo, crea o modifica el programa que se requiere para resolver este reto.
3. Prueba el programa, si no se resuelve el reto repite los pasos uno y dos. Una vez que lo hayas solucionado, abre un documento nuevo de texto (Word).
4. Copia el algoritmo que usaste para resolver el reto.

5. Realiza una impresión de pantalla, donde se pueda apreciar que el reto está resuelto. Se debe visualizar el código que usaste, así como el mapa del reto.
6. Avanza al reto siete, realiza una impresión de pantalla de la insignia que obtuviste y pégalas en el documento de texto.
7. Guarda el archivo de la siguiente manera: actividad05 + tus Iniciales, por ejemplo “actividad05JGAG”.
8. Cierra el documento.
9. Envíalo a través de la plataforma para su revisión.



Recomendaciones:

Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.



Tips para resolver el reto:

- Recuerda que la elaboración del algoritmo te auxiliará a resolver el reto.
- Debes dejar una marca para que cuando Robo regrese detecte que es el lugar donde inició su recorrido.
- Cuando recoja la baliza debe tomar posición para continuar avanzando sobre la línea. Por lo tanto, debes eliminar la instrucción “break” o “truncar” y agregar la o las instrucciones que le permitan retomar el camino y regresar y dejar la baliza donde inició el recorrido.
- Probablemente será necesario agregar una condición para detectar el color negro, ubicarse sobre la marca negra y soltar la baliza, en ese momento termina el ciclo.



B2 Tema III. Programando con Code

Propósito: desarrollar habilidades para elaborar rutinas de programación utilizando bloques de instrucciones que te permitan familiarizarte con la metodología utilizada en aplicaciones de programación como LEGO y mBlock.

En este módulo utilizarás la aplicación Web Code. (<https://code.org/>)



¿Qué es CODE?

CODE es una organización sin fines de lucro, fundada por los hermanos Hadi y Ali Partovi, que tiene como objetivo incentivar a la gente, en especial a los estudiantes de colegios a aprender sobre las Ciencias Computacionales.

La idea surgió porque en Estados Unidos hay una gran demanda de programadores: según datos de la organización, esta oportunidad de empleo se calcula tan grande que para 2020 habrá 1,4 millones de empleos relacionados con la informática frente a tan solo 400.000 estudiantes. La oferta crece a un ritmo que es el doble que, en otros campos laborales, pero en cambio, nueve de cada diez escuelas no ofrecen clases de programación.

Esta plataforma es una de las imprescindibles en el mundo de la programación. Plantea pequeños retos que deberemos ir resolviendo, jugando, con bloques de programación, para luego abrir a un mundo infinito de posibilidades a través de un editor ilimitado, y en el que podremos crear el contenido sobre la materia que queramos. Dispone de una infinidad de tutoriales adaptados a múltiples franjas de edades y multitud de recursos para profesores y docentes que deseen utilizar esta plataforma en sus clases. (Vargas Leguizamón, 2017)

Code.org es una organización sin ánimo de lucro, creada en el año 2013 que busca acercar la programación a todas las escuelas en el mundo, y especialmente a mujeres y colectivos minoritarios que desgraciadamente suelen ocupar un segundo lugar en este tipo de actividades. ¿Quieres saber más sobre ello?

Uno de los principales proyectos de Code.org es La Hora del Código, una iniciativa para que personas de todo el mundo se animen a crear tutoriales de una hora de duración, utilizando el lenguaje de programación visual de Google, Blockly. Un movimiento global que ya ha llegado a decenas de millones de estudiantes en más de 180 países. ¡Cualquier persona, en cualquier lugar del mundo, puede organizar una Hora del Código!

Algunos recursos que no te puedes perder

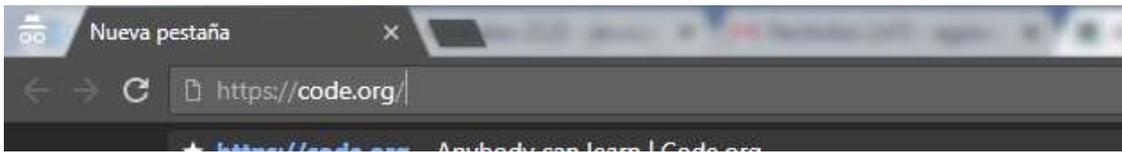
¿Quieres trabajar con tus estudiantes La Hora del Código? A continuación, te proponemos diversos artículos y recursos interactivos relacionados:

- La Hora del Código en el aula: ¿Cómo utilizar La Hora del Código en tus clases? No te pierdas esta pequeña guía con diferentes apartados para puedas sacar todo el provecho a esta iniciativa de Code.org. ¡Es muy fácil y sencillo!
- Star Wars, construye una galaxia con código: ¿Eres fan de esta ya clásica saga de ciencia ficción? Con este tutorial, aprenderás a programar droides y a crear tu propio juego de Star Wars en una galaxia muy, muy lejana...
- Minecraft: Utiliza los bloques de código para llevar a Steve o Alex en una aventura a través del mundo de Minecraft. Disponible en multitud de idiomas diferentes, ¡y para personas de 6 a 106 años!

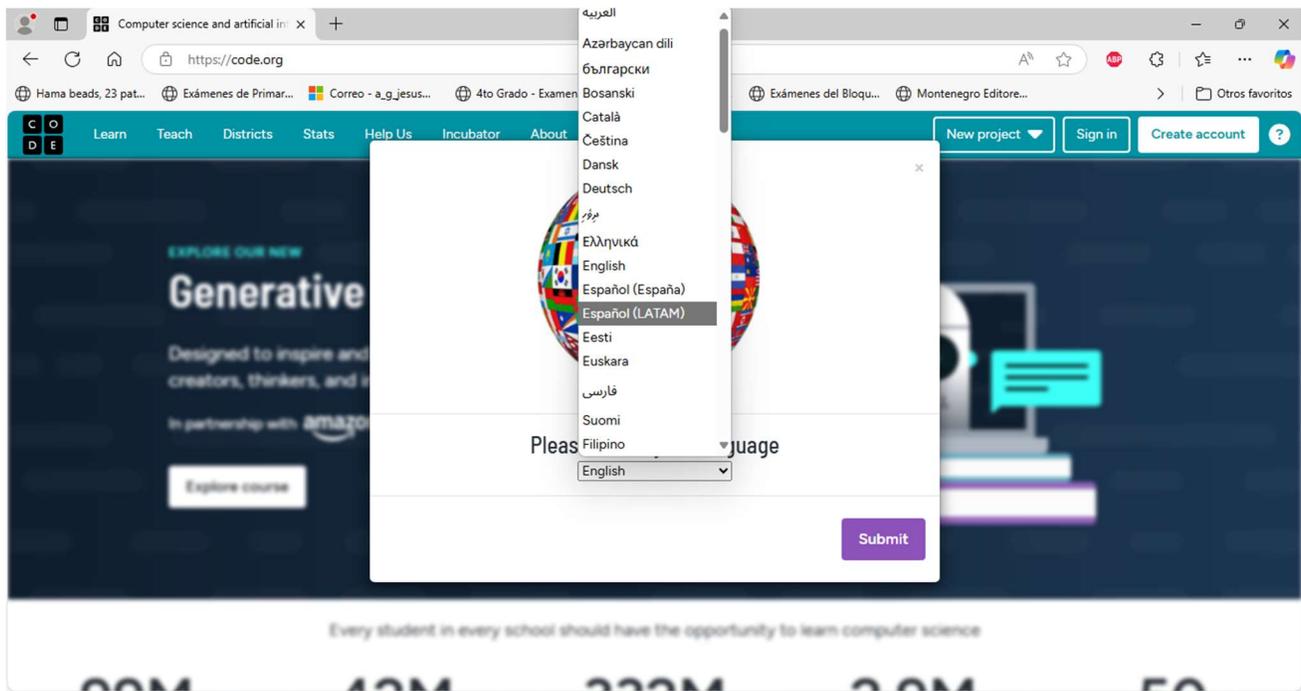
- ¡Da vida a tu robot con Blockly!: Si ya te has convertido en todo un experto de este lenguaje de programación, ¿te atreves a utilizarlo con un robot de verdad? Los robots educativos Dash & Dot utilizan la app Blockly como una herramienta más para el aprendizaje, ¡echa un vistazo! (Tiching, 2017)

Ingresar a code.org

Para ingresar a Code.org abre tu navegador y en la barra de direcciones escribe <https://code.org/> y presiona “Enter”.

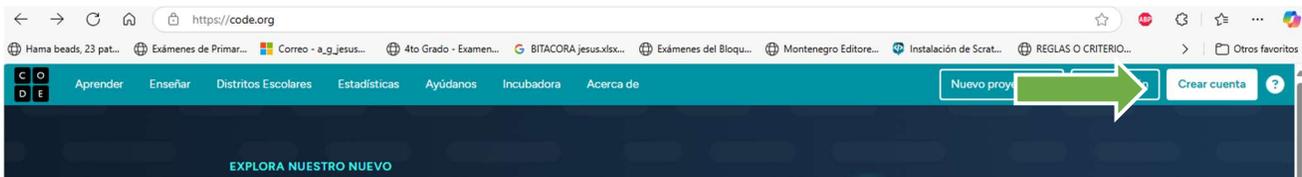


La primera vez que ingresas a la página te pide que selecciones el idioma en el que prefieres usar la aplicación, en mi caso selecciona “Español (LATAM)”.



Registro en code.org

Para guardar un registro del avance que realices en los diferentes cursos es recomendable que te registres en la aplicación, para ello da clic en “Crear cuenta”



Posteriormente puedes elegir entre cuenta estudiante o docente, selecciona “Registrarse como docente”.



Tienes varias opciones para crear tu cuenta, en esta ocasión crearás la cuenta utilizando una cuenta, que puede ser de Google, Microsoft, Facebook o Clever. Te indicaré el proceso para registrarte con una cuenta de Google que ya uses.

Iniciar sesión usando Google

En este curso, indicaré el proceso a seguir para iniciar sesión utilizando tu cuenta de Google.

Como primer paso da clic en “Registrarse con Google”.

Elija su método de inicio de sesión

Elige uno de los métodos de registro a continuación.

Regístrese con...

Simplifique el inicio de sesión con opciones sencillas de acceso único.

 Regístrese con Google

 Regístrese con Microsoft

 Regístrese con Facebook

 Regístrese con Clever

O regístrese con correo electrónico

Dirección de correo electrónico

Contraseña

o  Mínimo 6 caracteres

Confirmar contraseña

Te enviará a la siguiente pantalla. Al dar clic en el cuadro de texto te permitirá seleccionar la cuenta de Google que deseas utilizar, también puedes escribir el correo. Después de escribir el correo presiona el botón “Siguiente”.

 Acceder con Google



Inicia sesión

Ir a [Code.org](https://code.org)

Correo electrónico o teléfono

[¿Olvidaste el correo electrónico?](#)

Antes de usar Code.org, revisa su [política de privacidad](#) y [condiciones del servicio](#).

[Crear cuenta](#)

[Siguiente](#)

Escribe la contraseña y presiona nuevamente el botón “Siguiente”.

Acceder con Google



Te damos la bienvenida

 eqvd2019.04@gmail.com

Ingresa tu contraseña

Mostrar contraseña

Antes de usar Code.org, revisa su [política de privacidad](#) y [condiciones del servicio](#).

[¿Olvidaste la contraseña?](#) [Siguiente](#)

Te informará que si continúas Google compartirá algunos datos como tu nombre y correo electrónico, para continuar con el registro da clic en el botón “Continuar”.

Iniciar sesión con Google



Iniciar sesión en Code.org

 eqvd2019.04@gmail.com

Si continúas, Google compartirá tu nombre, tu dirección de correo electrónico, tu preferencia de idioma y tu imagen de perfil con Code.org. Consulta la [Política de Privacidad](#) y los [Términos del Servicio](#) de Code.org.

Puedes gestionar Iniciar sesión con Google en tu [cuenta de Google](#).

[Cancelar](#) [Continuar](#)

Ahora te pedirá algunos detalles como tu nombre dentro de code, tu función, etc. Complementa la información de acuerdo a tu perfil, después de eso da clic en el botón “Ir a mi cuenta”

Personalize su experiencia en Code.org con unos pocos detalles.

¿Cómo quieres que te llamen en Code.org?*

Sra. Coder

Esto es lo que verán sus estudiantes.

¿Cuál es su función principal?*

Seleccione un rol

¿En qué país se encuentra?*

Seleccione un país

¿Cuál es el nombre de su escuela u organización?*

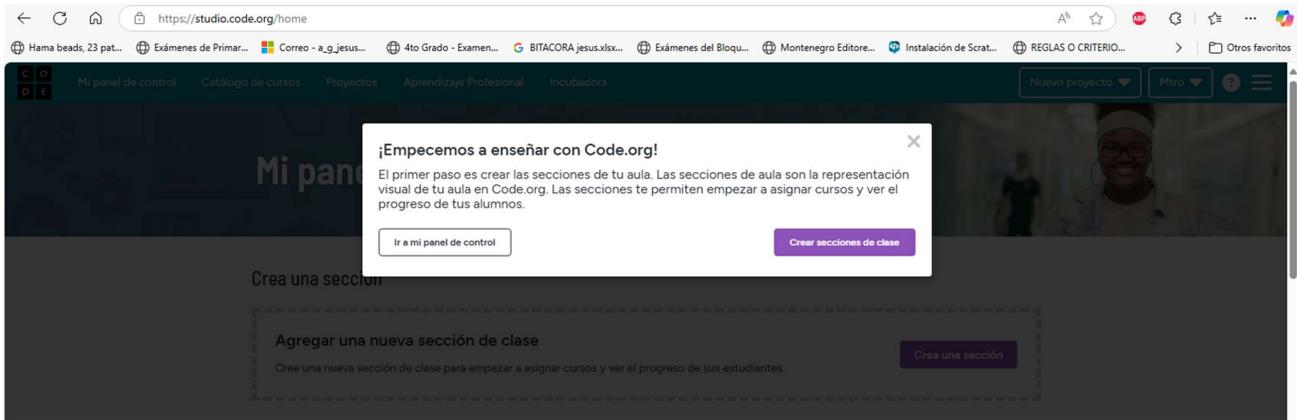
Manténganme informado:

Reciba correos electrónicos informativos sobre actualizaciones y oportunidades de Code.org (1-2 correos por mes)

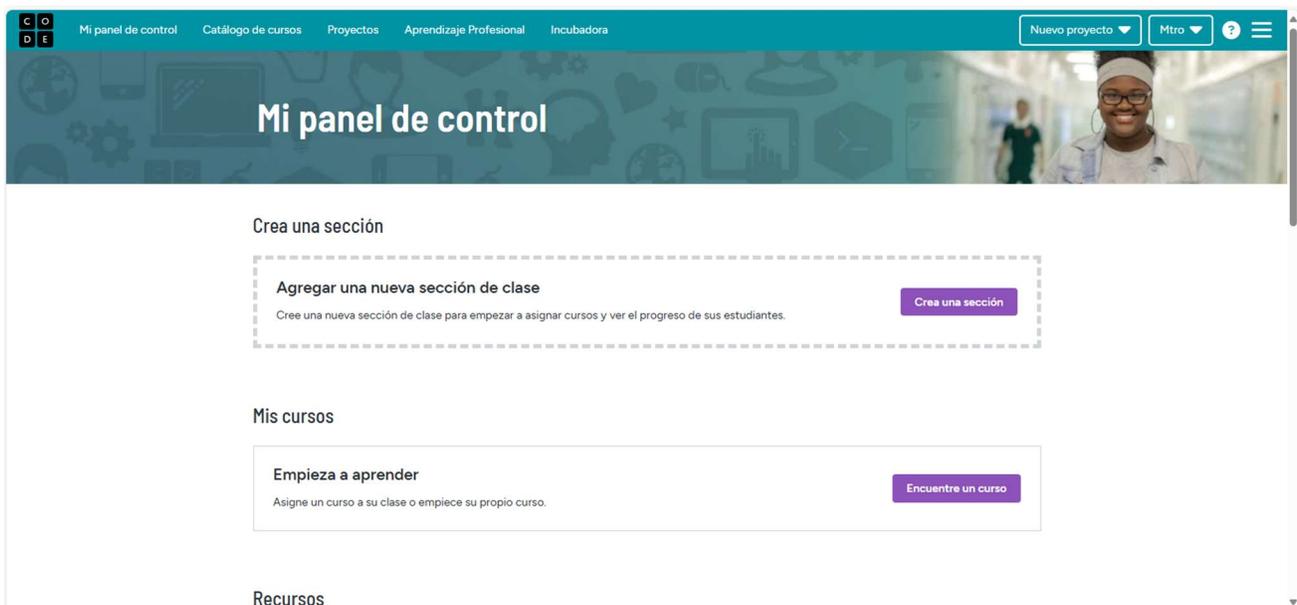
Nota: Después de crear su cuenta, recibirá un conjunto de correos electrónicos de incorporación para ayudarlo/a a comenzar.

[Ir a mi cuenta](#) →

Una vez que realizas el registro te mostrará la siguiente pantalla, da clic en “Ir a mi panel de control”.



Este es el “Dashboard” o tablero que visualizarás.



En “Mi panel de control” encontrarás algunas secciones como:

- “Crea una sección”. Donde podrás configurar tu aula virtual “Configurar tu sesión de clases”, y
- “Mis cursos”. En esta sección podrás realizar los cursos que la página te ofrece y se llevará un registro del avance que realices en los cursos.

En este momento trabajarás en la sección “Mis cursos”, ya que explorarás y realizarás algunos de los cursos con los que cuenta CODE.

Explora la pantalla de la página de Code.org

Si tu página aparece en inglés, y deseas cambiarla a español, desplázate a la parte inferior de esta, hasta el pie de página, ubica la opción para cambiar el idioma, selecciona el idioma que deseas utilizar.



Actividad 6. Exploración de CODE.

Indicaciones para realizar la actividad.

Una vez que te hayas registrado en CODE.org, explora las diferentes opciones que esta aplicación te ofrece a través del menú.



Qué debes hacer en esta actividad:

- Visitar las diferentes opciones que te brinda CODE a través de las fichas de menú: Mi panel de control, Catálogo de cursos, Proyectos y Acerca de. También explora los botones relacionados con la creación de proyectos “Crear” y la de tu cuenta, que en mi caso muestra “Robótica”, que es el nombre relacionado con mi cuenta.
- Realiza las anotaciones que consideres pertinentes y que te ayuden a ubicar las diferentes opciones que la página te muestra a través del menú.
- Si lo consideras necesario, agrega una impresión de pantalla por cada opción del menú.

Instrucciones para realizar la actividad

1. Abre un documento de texto nuevo (Word)
2. Escribe las anotaciones que consideres pertinentes, y que te puedan servir como referencia para facilitar tu navegación en esta plataforma.
3. Realiza impresiones de pantalla y pégalas en el documento. Estas impresiones de pantalla deben estar relacionadas a las notas que vayas generando.
4. La extensión del documento es libre, pero debes incluir o mencionar en tu documento todas las opciones que te ofrece CODE a través de su menú.
5. Al finalizar, guarda el archivo de la siguiente manera: actividad06 + tus Iniciales, por ejemplo “actividad06JGAG”.
6. Cierra el documento.
7. Envíalo a través de la plataforma para su revisión

Recomendaciones:



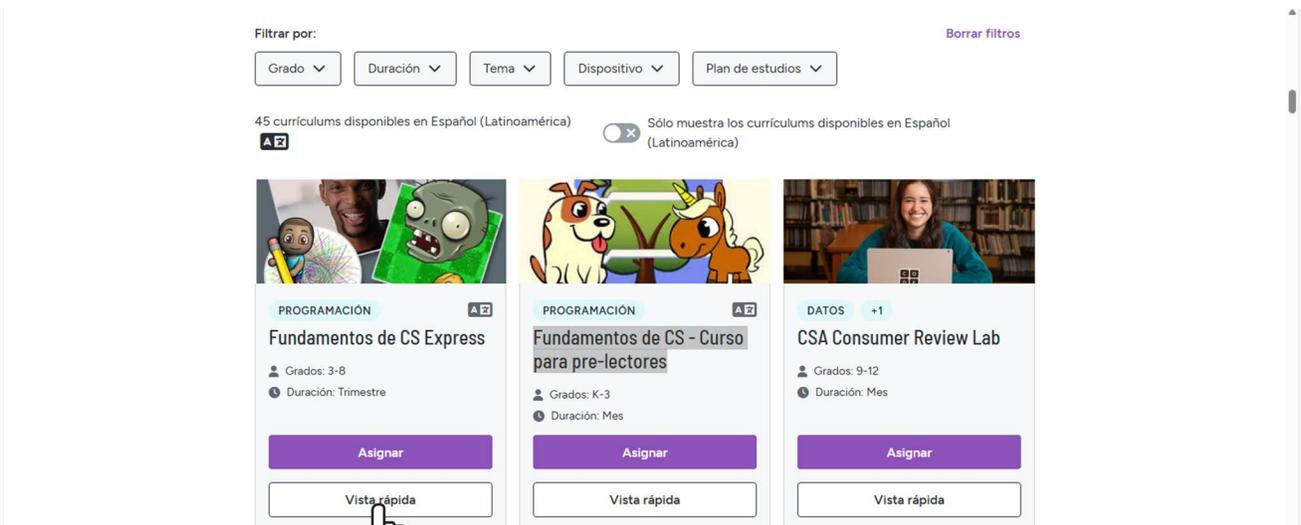
Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Trabajando con CODE

Para empezar a trabajar con CODE, utilizarás los ejercicios del “Fundamentos de CS - Curso para pre-lectores”, para ello da clic en “Catálogo de Cursos”.



Para empezar a familiarizarte con la interface de Code.org, ubica y da clic en “Fundamentos de CS - Curso para pre-lectores”. Esta actividad está marcada para edades de 4 a 8 años, sin embargo, te facilitará el trabajo en esta plataforma y posteriormente en cursos más avanzados.



Una vez ubiques el curso, da clic en “Vista rápida”. Después da clic en “Ver detalles del currículum”.

Filtrar por: Borrar filtros

Grado ▼ Duración ▼ Tema ▼ Dispositivo ▼ Plan de estudios ▼

45 currículums disponibles en Español (Latinoamérica) Sólo muestra los currículums disponibles en Español (Latinoamérica)

pace! Learn to create computer programs, solve problems, and work through fun challenges.

Code.org's CS Fundamentals: Kid Test... Más

KID TESTED
TEACHER APPROVED

0:05 / 3:48

Planes de lecciones

micro:bit Maker - Grade 2

Music Lab: Sesión Jam

● Computadora ● Chromebook ● Tableta ▲ Dispositivos móviles ● Sin conexión

Ver detalles del currículum Asignar a secciones de clase

Ahora podrás ver las diferentes lecciones, así como los diferentes ejercicios que conforman las diferentes lecciones, la primera lección que realizarás será “Lección 1: Aprende a arrastrar y soltar”, para iniciar con el ejercicio da clic en la opción “1”.

Curso Express para pre-lectores (2021)

Versión

2021 (Recomendado) ▼ ▼ Recursos del profesor ▼ Opciones de impresión

Ver página como: Estudiante Docente

Aprende los fundamentos de las ciencias de la computación y la seguridad en Internet. Al final del curso, crea tu propio juego o historia para compartir.

Mostrar todas las lecciones Ocultar todas las lecciones i ☰

▼ Secuencias

▼ Lección 1: Aprende a arrastrar y soltar

Esta lección dará a los estudiantes una idea de qué esperar cuando se dirigen al laboratorio de computación. Comienza con un breve debate sobre los modales en un laboratorio de computación y continúa con el uso de una computadora para completar desafíos en línea.

Ver el plan de lecciones

Enviar a estudiantes

1-12 Desarrollo de Habilidades

1 2 3 4 5 6 7 8 9 10 11 12

En las pantallas de las actividades podrás ver en la parte superior el ejercicio o “puzzle” que estás resolviendo; las instrucciones o qué es lo que requieres hacer en el ejercicio, así como algunos otros elementos para interactuar con la plataforma de CODE.

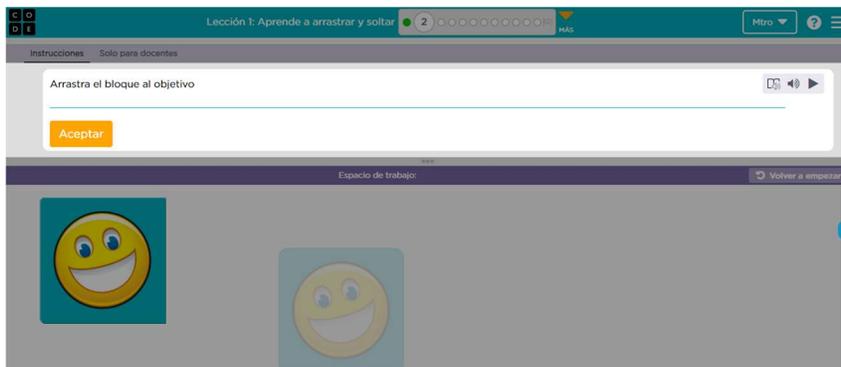
Por ejemplo, en el ejercicio 1, la indicación es que des clic en el bloque (manzana), para realizar el ejercicio debes dar clic en el botón “Aceptar”.



Una vez que resuelves el ejercicio, de acuerdo con lo solicitado, se despliega una pantalla como la siguiente, para cerrarla solo da clic en “Continuar”.



En el ejercicio dos debes arrastrar el bloque al área destino que se te indica.



Tip



Para mover un objeto, coloca el puntero del mouse sobre el objeto a mover, después da clic en el botón izquierdo del mouse y sin dejar de presionar el botón, muévelo

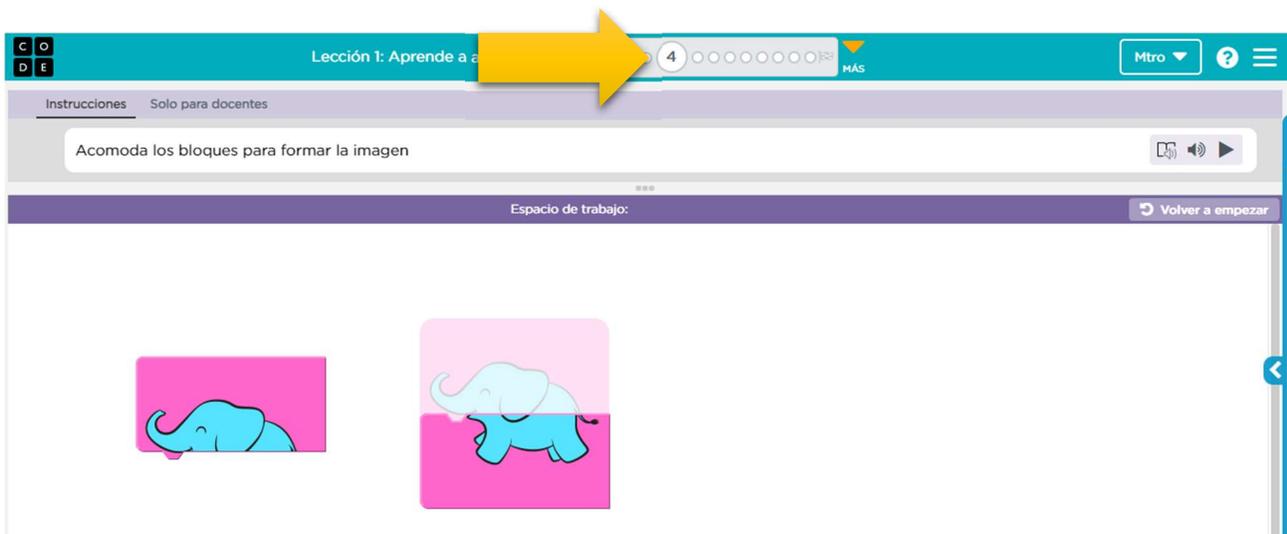
hacia el lugar donde lo quieres colocar, cuando esté sobre la posición indicada, suelta el botón del mouse.

Al igual que la actividad anterior, si la resuelves satisfactoriamente te mostrará una ventana indicándote que lo realizaste correctamente y que presiones “Continuar”.



Conforme avances en los “puzzles”, el grado de complejidad aumentará, pero todos los ejercicios van encaminados a que te familiarices con la interface de Code.org.

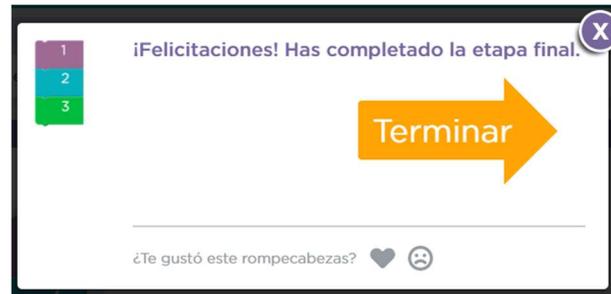
Te puedes apoyar en la barra de progreso, que se encuentra en la parte superior, para identificar el ejercicio que estás realizando y cuántos te faltan para terminar el curso.



Para avanzar a la siguiente etapa resuelve los siguientes ejercicios, debes llegar hasta la actividad 12.



Una vez que hayas terminado los ejercicios de esta lección podrás ver la notificación y felicitación de que has llegado a la etapa final.



Programación con bloques

Descripción de la interface de la aplicación

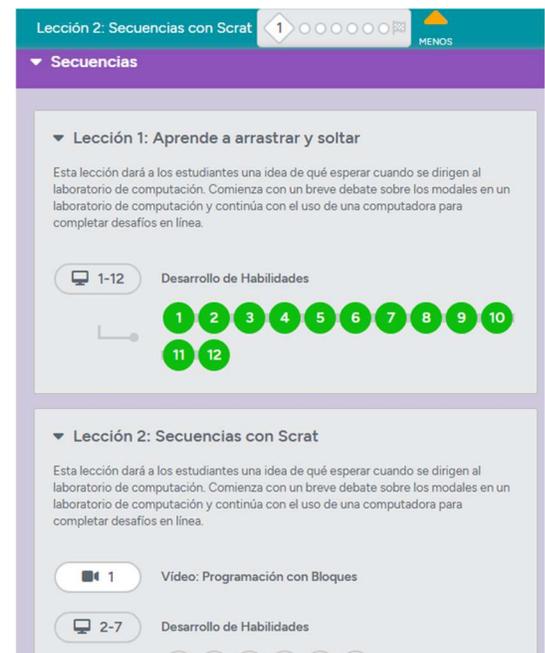
Antes de continuar, describiré brevemente la interface de la aplicación.

En la parte superior, se encuentra la barra de avance, ella te indica el ejercicio que estás realizando.



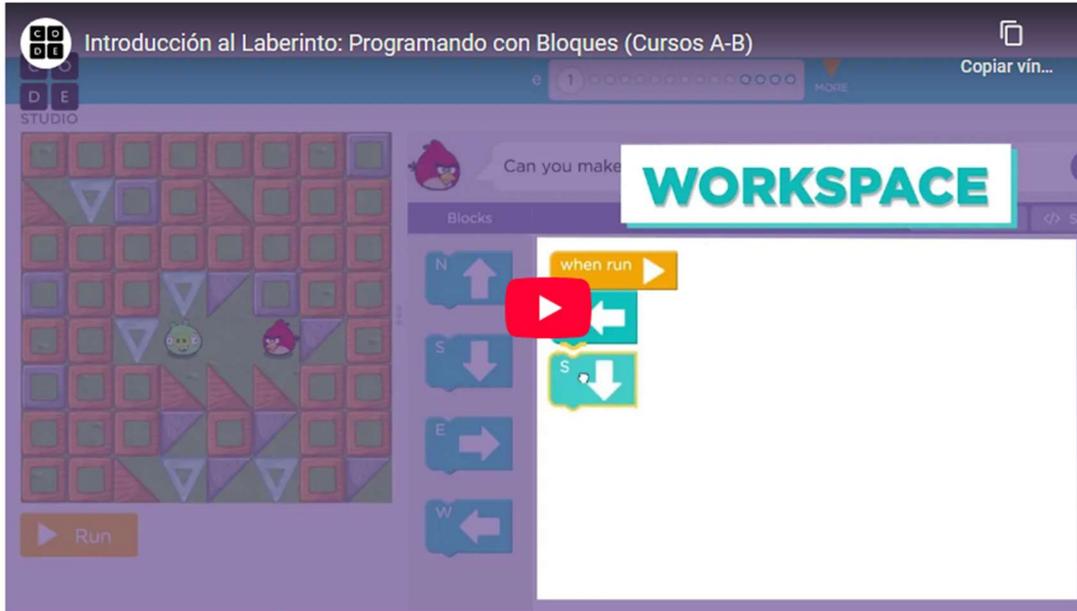
A la derecha de la barra de avance, se encuentra el botón “MÁS” que al  presionarlo te mostrará las opciones para seleccionar otra etapa o curso. Así como los que vayas realizando.

En este caso, te debe mostrar la lección 1 como completada. Cada que completes una lección te enviará automáticamente a la siguiente. También puedes seleccionar una lección fuera de la secuencia si así lo deseas.



La mayoría de las lecciones cuentan con video introductorios, los cuales te describen sobre lo que trata la lección.

Vídeo: Introducción a Code Studio



[Descargar el video](#)

Continuar

Cada lección te va indicando lo que debes realizar para cumplir el reto. Para ti como docente o conductor de una clase, te muestra una opción para resolver el reto, si necesitas de esta ayuda solo da clic en la opción “Solo para docentes”.



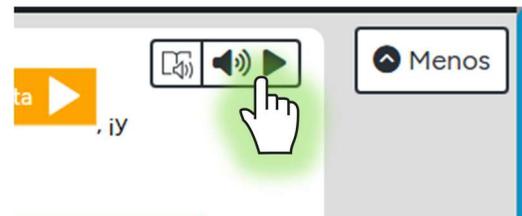
En la parte izquierda de la pantalla, encontrarás el escenario, el cual te plantea el reto que debes resolver. En la parte inferior del escenario, se encuentra el botón “Ejecutar”, que una vez que termines de crear el programa, des clic, se ejecute el programa y verifiques si cumple con lo requerido.



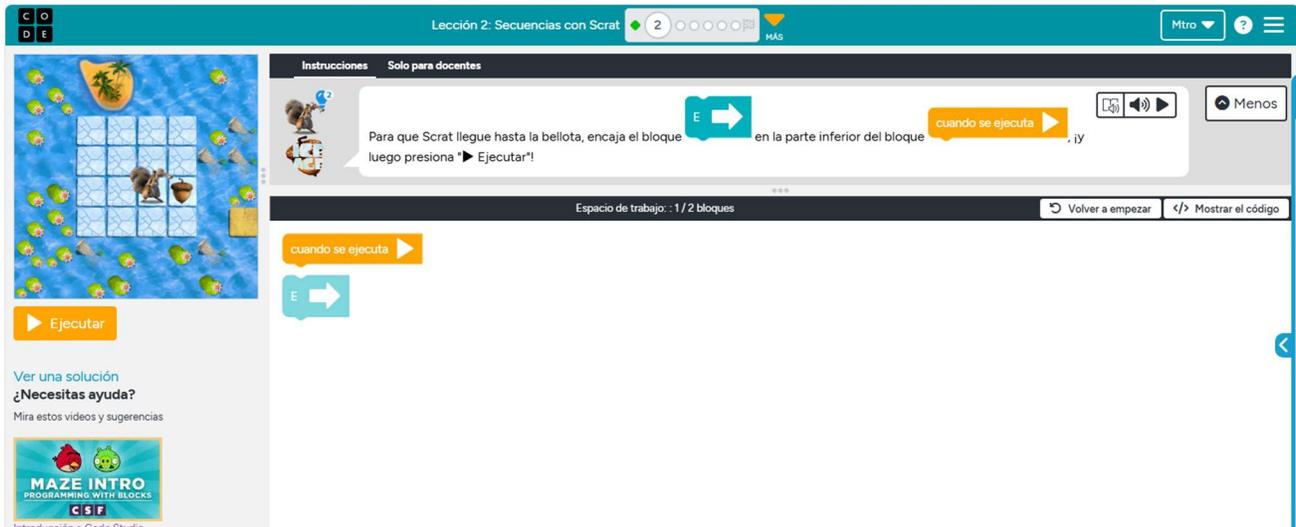
Bajo la barra de progreso se encuentra el área de notificaciones, ahí podrás ubicar las instrucciones o indicaciones para resolver el ejercicio.



Si este ejercicio lo realizan alumnos que no sepan o no puedan leer, pueden utilizar la opción para escuchar las instrucciones.



Bajo el área de notificaciones, se encuentra el “Espacio de trabajo”, que es donde crearás tu programa. Ahí podrás colocar los bloques de instrucciones que consideres necesarios para resolver el reto. Estos bloques deben ir “conectados” al bloque “cuando se ejecute”.



El “Espacio de trabajo” te muestra a manera de pista la cantidad de bloques que debes utilizar para resolver la actividad. En este caso te indica que están colocados un bloque de tres en “Espacio de trabajo: 1/3 bloques”.



Para agregar un bloque al espacio de trabajo solo selecciónalo y colócalo en el espacio. Recuerda que para que estos bloques sean tomados en cuenta al momento de ejecutar el programa, deben estar interconectados.



Para eliminar un bloque del espacio de trabajo, selecciónalo y muévelo hacia la izquierda, al área de bloques, cuando te muestre el bote de basura, solo suéltalo el botón del mouse.

Ahora continuemos resolviendo las lecciones.

La siguiente etapa se denomina “Lección 3. Programación con Angry Birds”

A partir de este ejercicio, empezarás a trabajar con bloques (instrucciones) de programación. Para seleccionar la lección con la que trabajarás a continuación, da clic en el botón “MÁS”.





Podrás ver las lecciones de las que consta el curso.

Busca y selecciona “Lección 3. *Programación con Angry Birds*”. Da clic en el ejercicio o puzzle 1 de esta lección.

Como puedes ver, al ingresar al ejercicio te brinda información de ayuda.



Lo que requieres hacer en el puzzle o ejercicio 1, es colocar los bloque que hagan que el ave se mueva y llegue hasta el cerdo, sin pasarse.



Analizando el escenario del ejercicio, es necesario que el ave se mueva tres espacios hacia la izquierda, para ello puedo colocar tres bloques que indiquen los tres movimientos del ave hacia la izquierda u oeste. Solo debo seleccionar los bloques de la sección “bloques” a la sección “espacio de trabajo” y colocarlos bajo el bloque “cuando se ejecuta”, como se muestra en la imagen, después da clic en el botón “Ejecutar”.



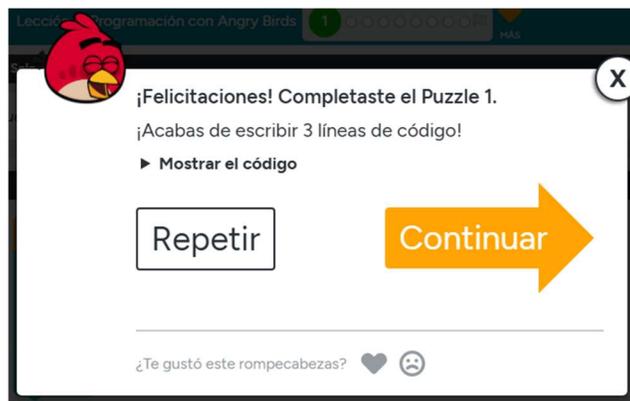
Ejecuta el programa para ver qué es lo que sucede.

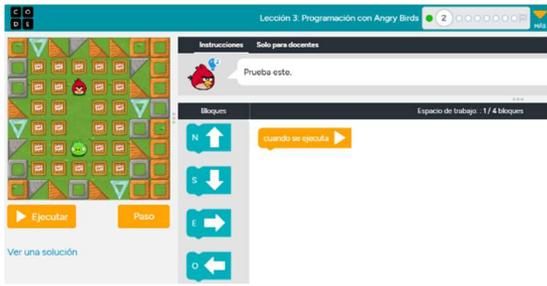
En el caso de que el programa no resuelva el reto, presiona el botón “Reiniciar”, que se encuentra en la parte inferior del escenario.



Posteriormente agrega los bloques que hagan falta para dar solución a la actividad y presiona el botón “Ejecutar” para probar nuevamente el programa.

Cuando hayas resuelto el “puzzle” presiona el botón continuar.





El ejercicio “puzzle” 2, requiere que ahora lleves al ave en dirección “Sur”, hacia el cerdo

De acuerdo con el escenario, el ave debe moverse tres veces hacia el Sur (abajo), agrega los bloques necesarios al programa, después presiona el botón

“Ejecutar”. Si es correcto, te permitirá avanzar al siguiente “puzzle”.

Este nivel es muy sencillo, así que no debes tener problemas para resolverlos.

Actividad 7 Resuelve los puzzles 3 al 9.

Después de practicar con los ejercicios 1 y 2 realiza los restantes para completar esta etapa.

Instrucciones para realizar la actividad

1. Abre un documento de texto (Word) nuevo.
2. Revisa y analiza las instrucciones que se te muestran en el área de notificaciones de cada ejercicio.
3. Observa y analiza el escenario, y tomando como referencia las instrucciones, elabora los bloques necesarios para crear el programa para resolverlo.
4. Después de resolverlo, realiza una impresión de pantalla del ejercicio resuelto y cópiala en el documento de texto que abriste en el punto 1. Debe apreciarse el código que elaboraste, así como el mapa del ejercicio.
5. Repite los pasos 2 al 4, hasta resolver el ejercicio 9.
8. Una vez resuelto el ejercicio 9. Guarda el documento la siguiente manera: actividadnn_tus Iniciales, por ejemplo “actividad07_JGAG”.
6. Envíalo a través de la plataforma para su revisión.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Bucles.

Antes de continuar recordemos que es un bucle en programación: Los bucles son una estructura de control de lenguajes de programación usada para realizar repeticiones. Mediante los bucles podemos realizar la ejecución de una o varias instrucciones de manera repetida.

En programación realizamos continuamente bucles para implementar algoritmos de todo tipo, desde los más básicos hasta los más complejos. De hecho, para la informática, el tratamiento automatizado de la información, los bucles son un elemento esencial, ya que nos permite iterar y realizar tratamientos y operaciones por conjuntos y estructuras de datos.

Continuando con nuestro curso, en este momento, podemos considerar que ya te has familiarizado con la interface de esta aplicación. Como ya se te mencionó, CODE te permite avanzar a otras etapas de un curso, sin que tengas que pasar por las anteriores.

Es momento que trabajes con bucles o ciclos. CODE cuenta en el “Curso D (2021)” con varias etapas en las que puedes practicar y desarrollar programas que involucren el uso de instrucciones de bucles o repeticiones.

Da clic en “Vista rápida”.

Después en “Ver detalles del currículum”.



Ahora ubica la “Lección 10. Dibujar formas con bucles” y da clic en el puzzle o ejercicio 1, que es este caso corresponde a un video informativo.

Lección 10: Dibujar formas con bucles 1

Vídeo: Introducción al Artista

Introducción al Artista con JR Hildebrand

move forward by 100 pixels

Descargar el video | Mostrar notas

Continuar

Después de observar el video, avanza al ejercicio 2, como ya has visto en cada ejercicio te da las instrucciones para resolverlo, en este caso te pide que ayudes al Artista a cubrir sus flores, indicando que debes avanzar 60 pixeles.

Lección 10: Dibujar formas con bucles 2

Instrucciones

¡Que día más hermoso!
Ayuda al Artista a cubrir sus flores antes del invierno avanzando 60 pixeles.

Bloques

Espacio de trabajo : 2 / 3 bloques

mover hacia adelante 60 pixeles

girar a la derecha 90 grados

girar a la izquierda 90 grados

saltar hacia adelante 100 pixeles

cuando se ejecuta

definir ancho 60

Ejecutar

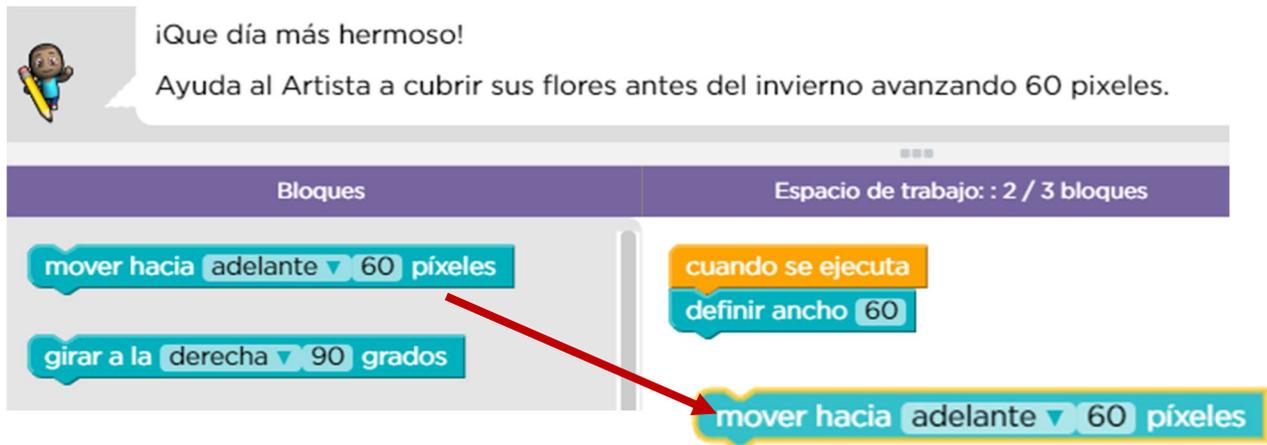
Como puedes ver la interface o pantalla que muestra es muy similar a la que ya trabajaste, sin embargo, ofrece otros elementos que te ayudarán a ejecutar el código de una manera más rápida o lenta.

Esta herramienta es un control deslizante, que se encuentra en la parte inferior del botón “Ejecutar”, y que al deslizarlo te permite ejecutar las



instrucciones de tu programa de una más rápida o lenta, dependiendo si lo mueves hacia la tortuga o hacia el conejo. La velocidad de ejecución dependerá de tus necesidades, si no deseas modificar la velocidad de ejecución del programa lo puedes dejar tal y como está.

Retomando el ejercicio, este requiere que sea resuelto dibujando una línea en utilizando tres bloques. De acuerdo con los bloques que hay en la sección de programación, solo hace falta un bloque, el cual podemos mover de la sección de bloques a la sección de programación, recuerda que debe de estar unido al resto de bloques para que funcione adecuadamente.



Presiona el botón “Ejecutar” para probar el código que creaste.

Los bloques utilizados para resolver este ejercicio deben quedar como en la imagen de la derecha.



Avanza al siguiente puzzle. En este ejercicio el objetivo es que repitas cinco veces determinados movimientos (instrucciones), cambiando de color aleatoriamente en cada movimiento. En este ejercicio utilizarás siete bloques, por lo cual el código puede quedar de la siguiente manera:

Lo primero que haremos será identificar qué bloques te permiten resolver el ejercicio, en este caso es el bloque “definir color | color al azar” y el bloque “mover hacia adelante 60 píxeles”, y como

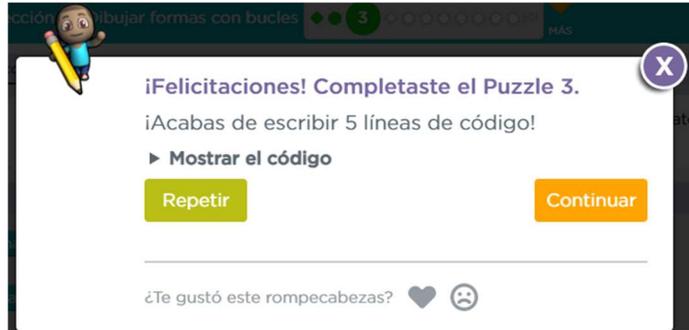
esto se debe repetir cinco veces, el código queda de la siguiente manera, y como puedes observar, se han utilizado los siete bloques que marca como pista.



Nota: en este ejercicio al usar los bloques “definir color” y “mover hacia”, CODE los considera como un solo bloque.



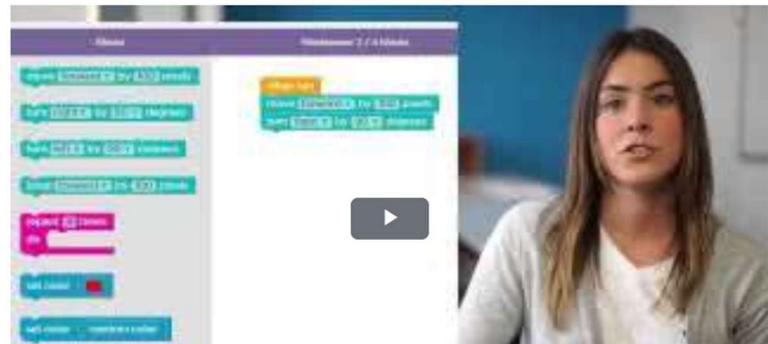
Es hora de verificar que el código que se acaba de crear resuelva el ejercicio, presiona o da clic en el botón “Ejecutar”. Si el código resuelve el ejercicio, podrás ver una imagen como la que se muestra a continuación. En caso contrario habrá que revisar el código y volver a probar los resultados.



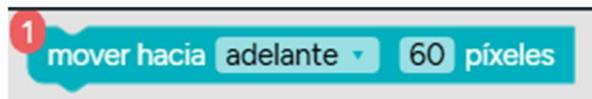
Resuelto el ejercicio, avanza a la siguiente actividad, la número 4 “Video: Bucles del artista”, en el cual se aborda un poco el tema de cómo usar los bucles o loop en programación.



Vídeo: Bucles del artista



Después de observar el video avanza a la lección 5, el ejercicio es muy similar al 3, sin embargo, las instrucciones indican que debes usar el bloque “loop” o “repetir”. Se indica también que se utilizarán cuatro bloques. Observa también que el bloque “mover hacia adelante 60 pixeles” solo se usa una vez.



Considerando lo anterior, el código sería el siguiente:

Usar el bloque repetir, y dentro del bloque repetir “definir color” y “mover hacia”.

Instrucciones

Dibuja la línea de colores 5 veces de nuevo, esta vez usando el bucle (loop) de 'repetir'.

Bloques | Espacio de trabajo: 4 / 4 bloques

0 mover hacia adelante 60 píxeles

girar a la derecha 90 grados

girar a la izquierda 90 grados

cuando se ejecuta

definir ancho 60

repetir 5 veces

haz

- definir color color al azar
- mover hacia adelante 60 píxeles

Después de haber creado el código, es hora de ejecutar el programa. Da clic en el botón “Ejecutar”. Resuelto el reto, podrás continuar con los ejercicios.

¡Felicitaciones! Completaste el Puzzle 5.
¡Acabas de escribir 5 líneas de código!

► **Mostrar el código**

Repetir Continuar

¿Te gustó este rompecabezas? ❤️ 😞

Actividad 8. Resuelve los ejercicios 6 al 11.

El **puzzle 6** requiere que elabores un programa, utilizando un bucle (loop) con una secuencia de bloques avanzar y girar a la derecha 144 grados para crear un patrón de estrella. Considera los bloques que se te indican para resolver la actividad.

Para resolver el **puzzle 7** se requiere escribir el código usando bucles (loops) para dibujar una escalera, como pistas te indica que cada escalón tiene 50 píxeles de largo y 50 píxeles de alto y que debes girar 90 grados en cada dirección. Considera los bloques que se te indican para resolver la actividad.

El **puzzle 8** se puede considerar ya casi resuelto, ya que la lógica utilizada es muy similar a la del ejercicio anterior, solo debes considerar la cantidad de pasos (loops) que se requieren realizar para resolver este puzzle. Considera los bloques que se te indican para resolver la actividad.

Para resolver el puzzle 9 se te pide que utilices loops o bucles (bloque repetir). Probablemente si ejecutes el código que ya está precargado, se resuelva el ejercicio, sin embargo, es necesario utilizar menos bloques. Observa que en el espacio de trabajo indica que se están utilizando 18 en lugar de 7 que pueden resolver el ejercicio.

Puzzle 10, es un desafío de acertijo, sin duda después de realizar los ejercicios anteriores, ya estás listo para resolverlo, así que presiona el botón ¡Estoy listo!



Este ejercicio requiere que utilices bucles, para reducir los bloques de código que se usan para resolverlo.

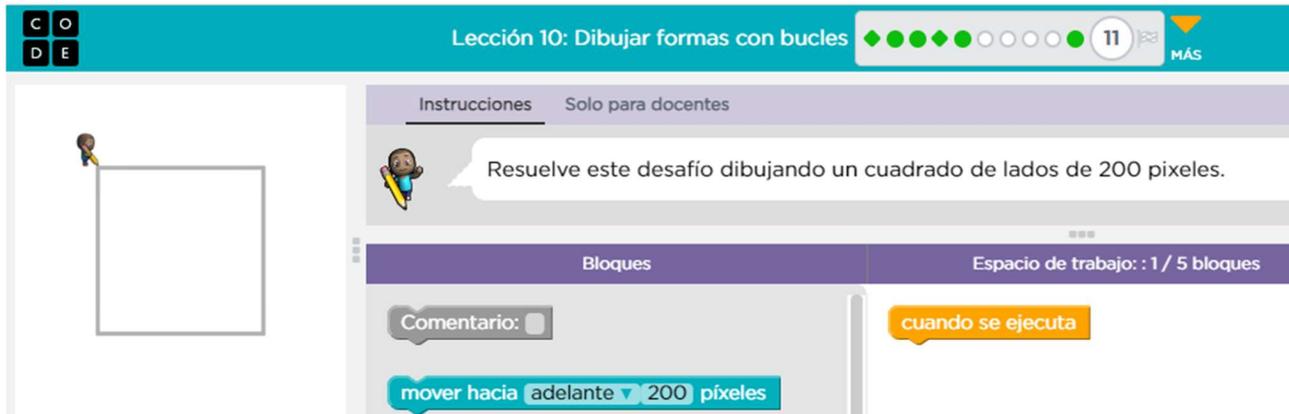
No olvides revisar la cantidad de bloques totales que se indican para resolver el reto. Así que

manos a la obra, modifica el código y verifica que se resuelva el ejercicio, en caso contrario vuelve a revisar y corregir lo que consideres necesario hasta resolverlo.

Resuelto el reto podrás continuar con el siguiente ejercicio. Da clic en el botón “Continuar”



Puzzle 11, último de esta lección requiere que elabores un cuadrado, cuyos lados deben de medir 200 píxeles de largo. Este ejercicio requiere que se usen 5 bloques, entre ellos el bloque repetir. Analiza cuáles serán los bloques a utilizar y elabora tu programa.



Una vez resuelto el ejercicio podrás continuar. Da clic en el botón “Terminar”.



Instrucciones para resolver la actividad

1. Analiza el ejercicio y plantea la solución de acuerdo a lo solicitado.
2. Agrega los bloques que hagan falta
3. Ejecuta el programa, de ser necesario modifica el programa las veces que sea necesario hasta resolver este reto.
4. Avanza al siguiente ejercicio (hasta resolverlos todos), repite los pasos 1 a 3 en cada uno de ellos.
5. Conforme resuelvas cada uno de los ejercicios, en un documento de texto (Word) creado para esta actividad, realiza un impresión de pantalla de cada ejercicio una vez que los vayas resolviendo y pega las imágenes de cada ejercicio resuelto en el documento.
6. Guarda el documento la siguiente manera: actividad08_tus Iniciales, por ejemplo “actividad08_JGAG”.

7. Envíalo a través de la plataforma para su revisión.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Tips para resolver el reto:

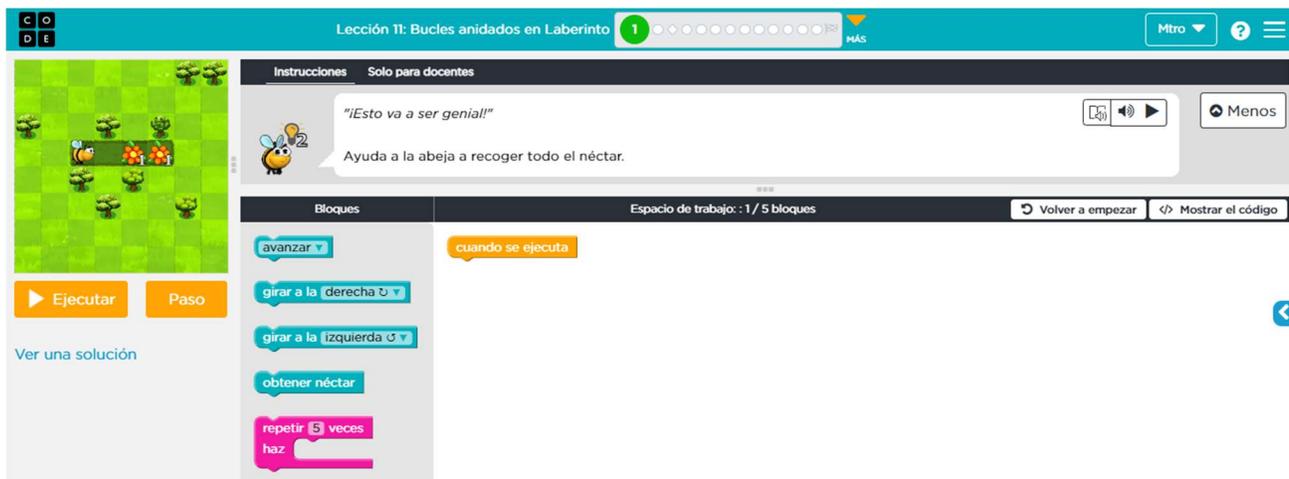


- Recuerda que la elaboración de un algoritmo te auxiliará a resolver el ejercicio.
- Recuerda que en esta lección se requiere el uso de bucles o loops.

Bucles II

Seguimos trabajando con bucles o ciclos de repetición. Para ello trabajaremos con la “Lección 11. Bucles anidados en laberinto”, en esta lección iremos incorporando algún tipo de acción dependiendo de lo que se requiera realizar.

Continuemos. En el ejercicio número 1 se requiere que la abeja recoja el néctar de las flores, observemos el siguiente escenario.



De acuerdo a lo requerido el algoritmo sería el siguiente:

Avanzar	Optimizando el algoritmo quedaría de la siguiente manera:
Avanzar	Avanzar
Recoger néctar	Repetir dos veces
Avanzar	Avanzar
Recoger néctar	Recoger néctar
Fin	Fin

El código sería el siguiente:

Copia el código y da clic en el botón “Paso” para que el programa se ejecute instrucción por instrucción.

Resuelto el ejercicio, continuemos con el ejercicio 3.



El ejercicio 2 se requiere que la abeja recoja todo el néctar.



De acuerdo a lo requerido el algoritmo sería el siguiente:

Avanzar
 Avanzar
 Recoger néctar
 Avanzar
 Recoger néctar
 Girar a la derecha

Avanzar
 Avanzar
 Recoger néctar
 Avanzar
 Recoger néctar
 Girar a la derecha

He separado el algoritmo para indicar las veces que es necesario repetir los mismos pasos o instrucciones, como puedes ver hay tres grupos, por lo tanto, se deben repetir las instrucciones tres veces, quedando de la siguiente manera:

Repetir 3 veces
 Avanzar
 Avanzar
 Recoger néctar
 Avanzar
 Recoger néctar
 Girar a la derecha

Fin

Avanzar
 Avanzar
 Recoger néctar
 Avanzar
 Recoger néctar
 Girar a la derecha

Ahora diseñemos el programa con base a este algoritmo.

Fin

Código del ejercicio 2

Copia el código, verifica los resultados, y en cuanto esté resuelto avanza al ejercicio 3.

Ejercicio 3

El ejercicio 3 corresponde a un video donde se explica el uso de bucles anidados. Obsérvalo y cuando finalice, da clic en continuar.

Continúa al ejercicio 4

Vídeo: Bucles anidados

Ejercicio 4. Para resolver este ejercicio es necesario, o se sugiere realizar una prueba de escritorio, para que, siguiendo los bloques veamos qué es lo que sucederá una vez que presionemos el botón “Ejecutar”.

La respuesta a este ejercicio es la opción “C”, selecciónala y da clic en “Ejecutar”.

Lección 11: Bucles anidados en Laberinto 4

Instrucciones Solo para docentes

Mira los bucles (loops) anidados de abajo. ¿Qué sucederá si haces clic en “Ejecutar”?

- A. La abeja avanzará y recogerá el néctar solo una vez.
- B. La abeja recogerá solo 2 unidades de néctar.
- C. La abeja recogerá todo el néctar.
- D. La abeja girará en la dirección equivocada.

Sólo ver:

```

cuando se ejecuta
  repetir 3 veces
    haz
      avanzar
      repetir 2 veces
        haz
          avanzar
          obtener néctar
      girar a la derecha
  
```

Ejecutar Paso

¿Necesitas ayuda?
Mira estos videos y sugerencias

Ejercicio 5. Nuevamente se requiere que la abeja recoja todo en néctar, analicemos el escenario para resolver el ejercicio.

Lección 11: Bucles anidados en Laberinto 5

Instrucciones Solo para docentes

Esta vez, ayuda a la abeja a recoger todo el néctar usando los menos blo

Bloques Espacio de trabajo: 1 / 6 bloques

```

cuando se ejecuta
  avanzar
  girar a la derecha
  girar a la izquierda
  obtener néctar
  repetir 2 veces
    avanzar
    repetir 5 veces
  
```

Ejecutar Paso

Ver una solución
¿Necesitas ayuda?
Mira estos videos y sugerencias

Algoritmo

Repetir 4 veces
 Repetir 3 veces
 Recoger néctar
 Avanzar

La explicación al código sería:
 Repetir 4 veces, que sería y por cada lado
 En cada lado repite 3 veces
 Recoger néctar
 Avanzar

Girar a la derecha

Termina este bucle y gira a la derecha, vuelve a empezar

Fin

Fin

El código sería el siguiente, cópialo y ejecuta el programa para verificar que este sea correcto. Posteriormente avanza al ejercicio 6.

The screenshot shows a programming interface for a maze game. At the top, it says "Lección 11: Bucles anidados en Laberinto" with a progress indicator showing 5 out of 6 steps completed. Below this, there's a "Solo para docentes" section with instructions: "Esta vez, ayuda a la abeja a recoger todo el néctar usando los menos bloques". The "Bloques" palette on the left contains: "avanzar", "girar a la derecha", "girar a la izquierda", and "obtener néctar". The workspace on the right shows a script starting with "cuando se ejecuta", followed by a "repetir 4 veces" loop. Inside this loop is a "repetir 3 veces" loop containing "obtener néctar" and "avanzar" blocks, and a "girar a la derecha" block.

Actividad 9. Resuelve los ejercicios 6 al 9

Para resolver estos ejercicios usa lo aprendido en estas lecciones, recuerda usar bucles, bucles anidados, todo de lo que puedas hacer uso para resolverlos, considerando además las indicaciones de cada uno de los ejercicios.

Instrucciones para resolver la actividad

1. Analiza el ejercicio y plantea la solución de acuerdo con lo solicitado.
2. Agrega los bloques que hagan falta
3. Ejecuta el programa, de ser necesario modifica el programa las veces que sea necesario hasta resolver este reto.
4. Avanza al siguiente ejercicio (hasta resolverlos todos), repite los pasos 1 a 3 en cada uno de ellos.
5. Conforme resuelvas cada uno de los ejercicios, en un documento de texto (Word) creado para esta actividad, realiza una impresión de pantalla de cada ejercicio una vez que los vayas resolviendo y pega las imágenes de cada ejercicio resuelto en el documento, en cada imagen se debe visualizar el código, así como la vista del escenario resuelto.
6. Guarda el documento la siguiente manera: actividad09_tus Iniciales, por ejemplo “actividad09_JGAG”.
7. Al finalizar los ejercicios, envíalo a través de la plataforma para su revisión.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Tips para resolver el reto:



- Recuerda que la elaboración de un algoritmo te auxiliará a resolver el ejercicio.
- Recuerda que en esta lección se requiere el uso de bucles o loops.

Actividad 10. Resuelve los ejercicios 10 al 13

Para resolver estos ejercicios usa lo aprendido en estas lecciones, recuerda usar bucles, bucles anidados, todo de lo que puedas hacer uso para resolverlos, considerando además las indicaciones de cada uno de los ejercicios.

Instrucciones para resolver la actividad

1. Analiza el ejercicio y plantea la solución de acuerdo con lo solicitado.
2. Agrega los bloques que hagan falta
3. Ejecuta el programa, de ser necesario modifica el programa las veces que sea necesario hasta resolver este reto.
4. Avanza al siguiente ejercicio (hasta resolverlos todos), repite los pasos 1 a 3 en cada uno de ellos.
5. Conforme resuelvas cada uno de los ejercicios, en un documento de texto (Word) creado para esta actividad, realiza una impresión de pantalla de cada ejercicio una vez que los vayas resolviendo y pega las imágenes de cada ejercicio resuelto en el documento, en cada imagen se debe visualizar el código, así como la vista del escenario resuelto.
6. Guarda el documento la siguiente manera: actividad10_tus Iniciales, por ejemplo “actividad10_JGAG”.
7. Al finalizar los ejercicios y con las evidencias solicitadas, envíalo a través de la plataforma para su revisión.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Tips para resolver el reto:



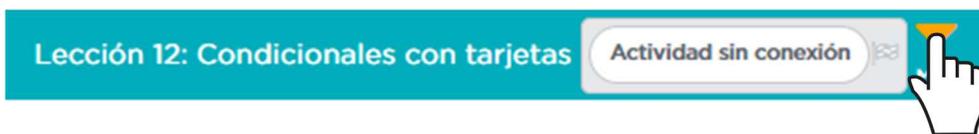
- Recuerda que la elaboración de un algoritmo te auxiliará a resolver el ejercicio.
- Recuerda que en esta lección se requiere el uso de bucles o loops.

Lenguajes orientados a objetos

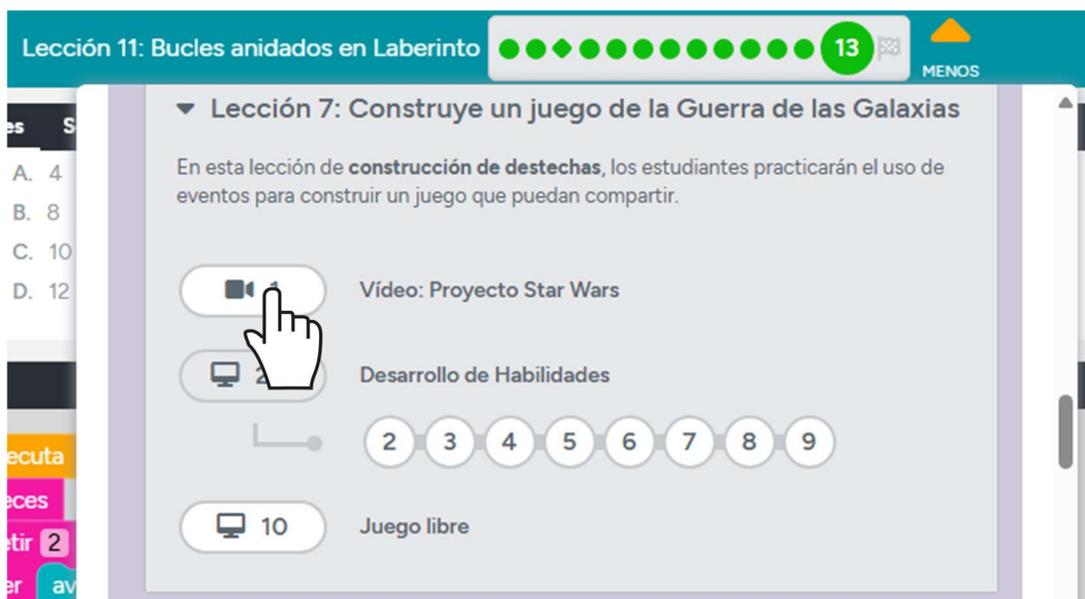
Una vez terminada la “Lección 11: Bucles anidados en Laberinto”, después de presionar el botón “Continuar” te envía a la “Lección 12: Condicionales con tarjetas”, sin embargo, para continuar las actividades sugeridas con nuestro curso es necesario practicar con la “Lección 7: Construye un juego de la Guerra de las Galaxias”

Para ello realiza lo siguiente:

1. Clic en el botón “MÁS”.



2. Ubica la “Lección 7: Construye un juego de la Guerra de las Galaxias” y para iniciar da clic en la actividad 1, que corresponde a un video.



Observa el video, ya que en esta lección trabajarás con programación orientada a objetos, metodología de programación utilizada por aplicaciones como Scratch o mBlock. Pero ¿Qué es la programación orientada a objetos?

Lenguajes orientados a objetos

De manera muy general, se puede mencionar que la Programación Orientada a Objetos (POO) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

Conceptos fundamentales

Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Herencia: es la facilidad mediante la cual la clase "B" hereda en ella cada uno de los atributos y operaciones de "A", como si esos atributos y operaciones hubiesen sido definidos por la misma "B". Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en "A".

Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.

Método: algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento: es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el

mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

Mensaje: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Propiedad o atributo: Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

Como te he mencionado, en esta lección conocerás y trabajarás el principio de “Evento”, entre otros elementos de la programación orientada a objetos.

Inicia con la lección, viendo las veces que sea necesario el video. Después da clic en el botón “Continuar”.

Lección 7: Construye un juego de la Guerra de las Galaxias 1 MAS

Vídeo: Introducción al proyecto de Star Wars



Descargar el video

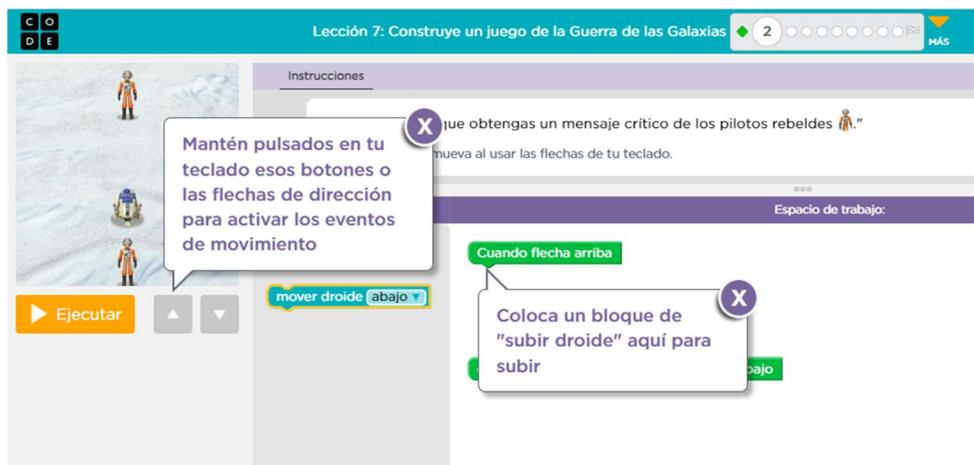
Continuar



Ejercicios.

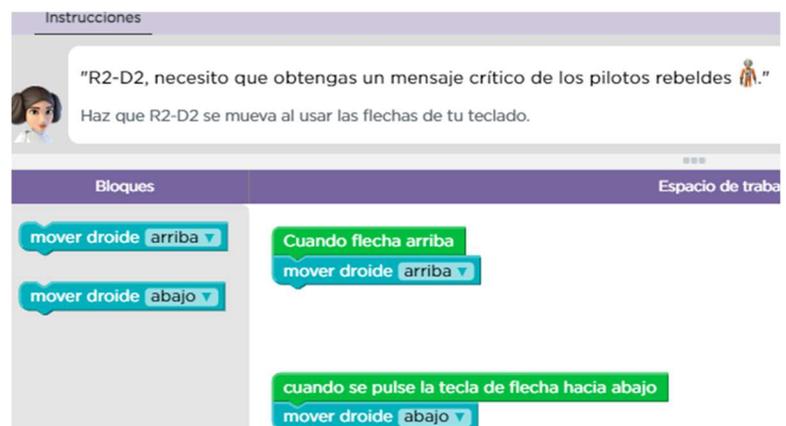
Puzzle 2

El **puzzle 2** te brinda algunos tips, indicando que debes hacer cada vez que se active un evento a través de cierta acción que realices, en este caso presionando las flechas de dirección arriba / abajo; así como la programación del evento o acción, es decir que sucederá si presionas las flechas de dirección. Una vez que leas los mensajes de alerta, ciérralos para continuar.



De acuerdo con las instrucciones de esta lección, al momento de usar las flechas de dirección "R2-D2", debe moverse de acuerdo a la flecha que presiones en el teclado. Los bloques quedarían de la siguiente manera.

Una vez que coloques los bloques como se muestra en la imagen, da clic en el botón "Ejecutar" para ver si funciona de acuerdo con lo requerido.



Solucionado el puzzle da clic en el botón "Continuar"



Puzzle 3

Para resolver el **puzzle 3**, es necesario que “R2-D2” obtenga o tome los “Pilotos Rebeldes”, para ello es necesario que programes los eventos que se ejecutarán cada vez que presiones las flechas de direcciones. Observa que en este ejercicio existen 4 eventos a programar, uno para cada flecha. Por lo tanto, hay que colocar el bloque del acción junto al bloque del evento.

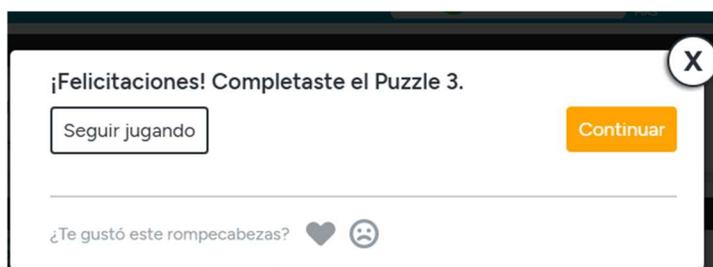


Los bloques de código quedarían de la siguiente manera. Donde al presionar flecha arriba “R2-D2” debe moverse hacia arriba, así con el resto de los eventos y acciones a ejecutarse cada vez que se presionen las flechas.

Copia el código y ejecuta el programa para ver los resultados.



Resuelto el puzzle da clic en el botón “Continuar”.



Puzzle 4

En el **puzzle 4** se incorporan algunos elementos como la reproducción de sonidos y suma de puntos. Dentro de la información del reto se menciona que, aunque no está visible el código para mover a “R2-D2” lo puedes mover; además que puedes utilizar el bloque de sonidos para emitir un sonido cada vez que se toque la meta, en esta caso los “Pilotos Rebeldes”.

Cierra las ventanas de los mensajes para continuar.



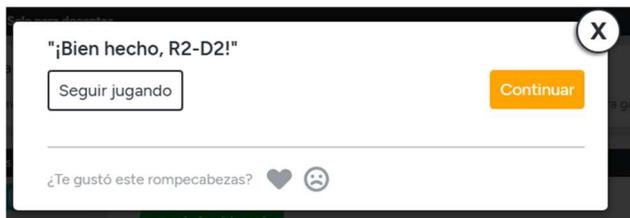
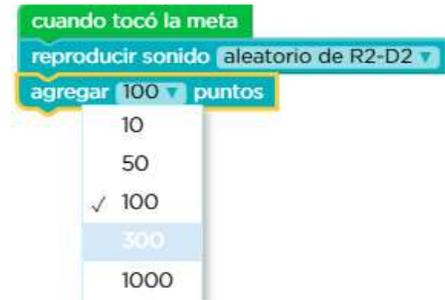
Las instrucciones indican que tendrás que agregar puntos cada vez que “R2-D2” tome un “Piloto Rebelde”, también indica que debes conseguir 900 puntos para ganar; en este sentido puedes ver en el área de bloques que existe el de “agregar 100 puntos”.



Como ya estaba colocado el bloque “reproducir sonido aleatorio de R2-D2”, solo agregamos el bloque “agregar 100 puntos”.



Ahora, para ganar se requieren 900 puntos. Si se ejecuta el programa con la configuración de los bloques actual, no ganaremos ya que solo se estarán generando 300 puntos en total. Entonces, para ganar hay que asignar más puntaje al momento de tomar cada piloto; si se requieren 900 puntos y son 3 pilotos, por lo tanto, se requiere ganar 300 puntos por cada piloto. Para modificar este valor da clic sobre el número “100” del bloque agregar, y selecciona “300”.



Después de esto, es hora de probar el código y verificar si se gana al obtener los 900 puntos.

Resuelto el reto da clic en el botón “Continuar”.

Puzzle 5

Para resolver el **puzzle 5**, es necesario obtener 200 puntos. Los puntos se pueden ganar o perder de la siguiente manera:

- Ganas 100 puntos al tomar Piloto Rebelde. 
- Pierdes 100 puntos al tocar un Soldado de ataque 
- Ganas 100 al tomar un Mynock 

Tomando como referencia lo anterior, los bloques de programación de los eventos quedarían de la siguiente manera:

"Cuidado desde los soldados de asalto 🤖."

Agrega 100 puntos cuando R2-D2 tiene un Piloto Rebelde 🛩️. Elimina 100 cuando tiene a un Soldado de asalto 🤖. Consi rompecabezas. (Puedes usar el Mynock 🐛 para que sea más fácil.)

Bloques	Espacio de trabajo:
reproducir sonido aleatorio	cuando consigue a piloto rebelde
agregar 100 puntos	agregar 100 puntos
quitar 100 puntos	cuando consigue Mynock
	agregar 100 puntos
	cuando consigue Stormtrooper
	quitar 100 puntos

Ejecuta el programa y verifica si puedes ganar este reto.

Una vez solucionado el reto puedes modificar el código de cada evento. Por ejemplo, agregar un sonido a cada evento, lo que se te ocurra, la cuestión es verificar que sucede al modificar o agregar diferentes bloques. Para avanzar da clic en el botón "Continuar".

Puzzle 6

Lección 7: Construye un juego de la Guerra de las Galaxias

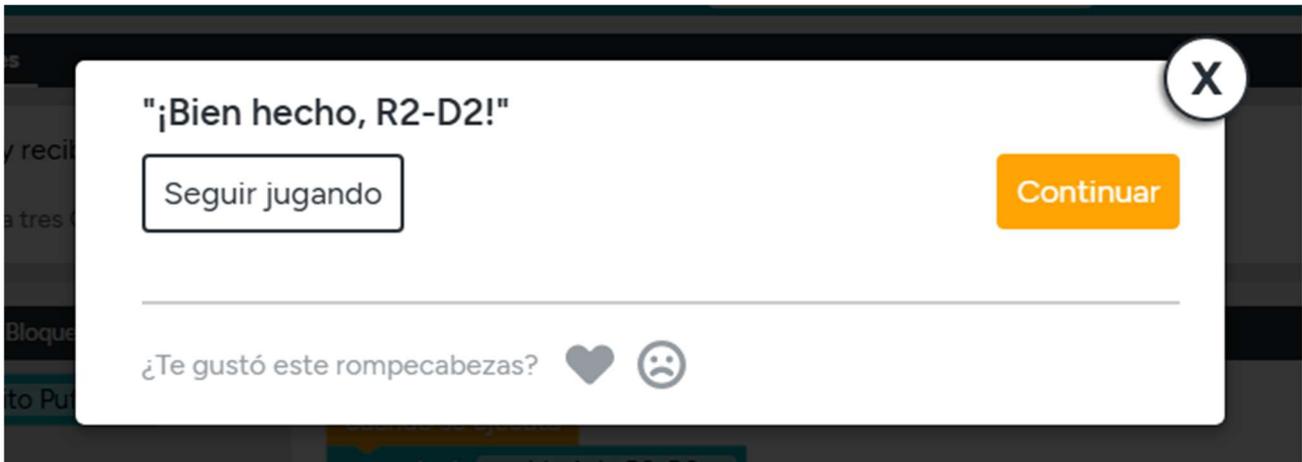
Instrucciones: "Estoy recibiendo señales de actividad en este planeta".
Agrega tres Cerditos Puffer 🐷 al planeta. Luego, obténlos.

Bloques	Espacio de trabajo:
Añadir Cerdito Puffer	cuando se ejecuta
agregar 100 puntos	reproducir sonido 1 de R2-D2
quitar 100 puntos	Añadir Cerdito Puffer
reproducir sonido aleatorio	
cuando consigue globo cerdo	agregar 1000 puntos

Pon aquí los comandos a ejecutar cuando se inicia el programa

Para resolver el **puzzle 6** se requiere primero agregar "Cerditos Puffer", posteriormente debes tomarlos y agregar 1000 puntos por cada cerdito.

Al analizar los bloques precargados en el bloque “cuando se ejecuta” ya se agrega un cerdito, es necesario agregar dos bloques más “Añadir Cerdito Puffer” para que sean los tres. Con esto se debe resolver el ejercicio, ya que no se menciona que se obtenga algún puntaje. Probemos el programa.



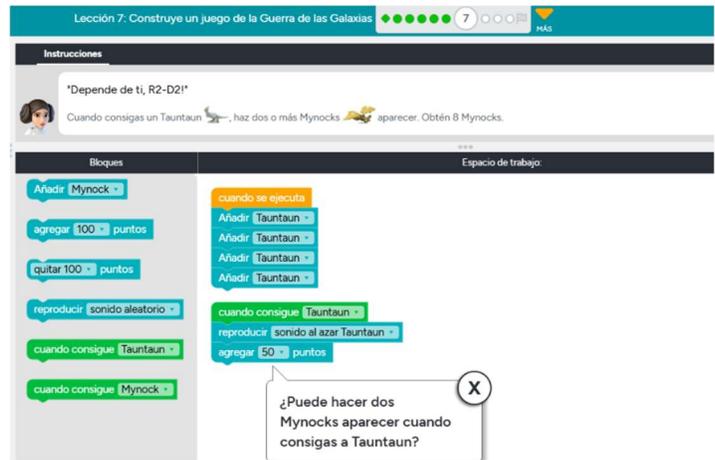
Una vez realizadas las modificaciones adecuadas podrás avanzar al puzzle 7.

Actividad 11 resuelve los puzzles 7, 8 y 9

Indicaciones para resolver el *puzzle 7*.

- Se requiere que por cada “Tauntaun” que tomes, aparezcan dos “Mynock”.
- Se deben recolectar los 8 “Mynocks” para avanzar al siguiente reto.

Analiza los bloques existentes, y de acuerdo con las instrucciones para resolverlo, agrega los bloques que consideres necesarios. Puedes agregar los bloques que desees para mejorar o enriquecer tu programa.



Una vez resuelto avanza al siguiente puzzle.

Aspectos para considerar en la evaluación de la actividad:

1. El código desarrollado debe incluir lo siguiente:
 - Evento donde al tomar un “Tauntaun” aparezcan dos “Mynock”.
 - Evento donde al tomar un “Mynock” puedas acumular puntos.
 - Puedes utilizar a tu consideración diversos eventos para enriquecer tu programa.
2. Ejecuta tu programa las veces que consideres necesario, realiza los ajustes que creas pertinentes.
3. Una vez que CODE te indique que tu puzzle está resuelto, toma una impresión o captura de pantalla del área del código donde se puedan apreciar los bloques utilizados, así como la sección del escenario.
4. Crea un documento nuevo de texto (Word) y pega la imagen de la impresión de pantalla.
5. Guarda el documento con el nombre actividad11_tusiniciales, por ejemplo “actividad11_JGAG”.
6. Puedes cerrar el archivo o mantenerlo abierto para seguir guardando las evidencias de los siguientes puzzles.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Indicaciones para resolver el *puzzle 8*.

- Cada vez que “R2-D2” tome un “Droide ratón”, aparezcan dos más.
- Para resolver el reto, “R2-D2” debe tomar 20 “Droides ratón”.

Analiza los bloques existentes, y de acuerdo con las instrucciones para resolverlo, agrega los bloques que consideres necesarios. Puedes agregar los bloques que desees para mejorar o enriquecer tu programa. Una vez resuelto avanza al siguiente puzzle.

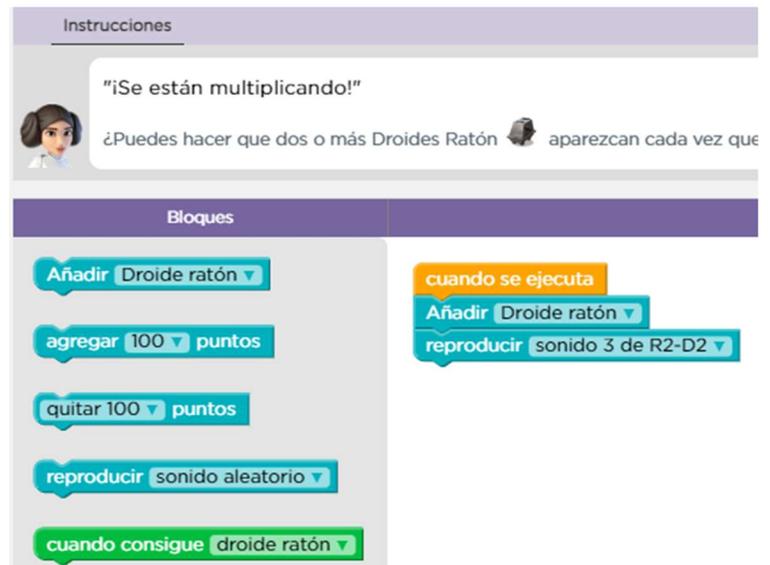
Aspectos para considerar en la evaluación de la actividad:

1. El código desarrollado debe incluir lo siguiente:

- Bloque donde indique que cada vez que “R2-D2” tome un “Droide ratón”, aparezcan dos más.
- Para resolver el reto, “R2-D2” debe tomar 20 “Droides ratón”.
- Puedes utilizar, a tu consideración, diversos eventos para enriquecer tu programa.

2. Ejecuta tu programa las veces que consideres necesario, realiza los ajustes que creas pertinentes.

3. Una vez que CODE te indique que tu puzzle está resuelto, toma una impresión o captura de pantalla del área del código donde se puedan apreciar los bloques utilizados, así como la sección del escenario.



4. Abre el documento de texto (Word) que creaste para resolver el puzzle anterior, ve al final del documento y pega la imagen de la impresión de pantalla.
5. Guarda el documento.
6. Puedes cerrar el archivo o mantenerlo abierto para seguir guardando las evidencias de los siguientes puzzles.

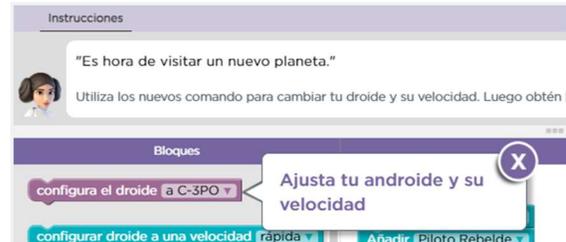
Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Indicaciones para resolver el *puzzle 9*.

Al iniciar el puzzle podrás ver la ventana de mensaje que indica que puedes ajustar la velocidad del androide. Para continuar cierra la ventana.



Para resolver el puzzle es necesario realizar lo siguiente:

- Utilizar un bloque que te permita cambiar de droide.
- Utilizar un bloque que te permita establecer la velocidad del droide.

Analiza los bloques existentes, y de acuerdo a las instrucciones para resolverlo, agrega los elementos que consideres requeridos. Puedes agregar los bloques que desees para mejorar o enriquecer tu programa. Prueba agregando el bloque evento con diversas acciones.



Aspectos para considerar en la evaluación de la actividad:

1. El código desarrollado debe incluir lo siguiente:
 - Utilizar un bloque que te permita cambiar de droide.
 - Utilizar un bloque que te permita establecer la velocidad del droide.
 - Puedes utilizar a tu consideración diversos eventos para enriquecer tu programa.
2. Ejecuta tu programa las veces que consideres necesario, realiza los ajustes que creas necesarios.
3. Una vez que CODE te indique que tu puzzle está resuelto, toma una impresión o captura de pantalla del área del código donde se puedan apreciar los bloques utilizados, así como la sección del escenario.
4. Abre el documento de texto (Word) que creaste para resolver el puzzle anterior, ve al final del documento y pega la imagen de la impresión de pantalla.
5. Guarda el documento.
6. Puedes cerrar el archivo o mantenerlo abierto para seguir guardando las evidencias de los siguientes puzzles.

Recomendaciones:



Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Actividad 12 Resuelve el *puzzle 10*.

Al iniciar este puzzle te recomiendo ver el video de inicio, en él se comentan algunas experiencias de esta lección, observa con atención y al terminar procede a resolver el puzzle.



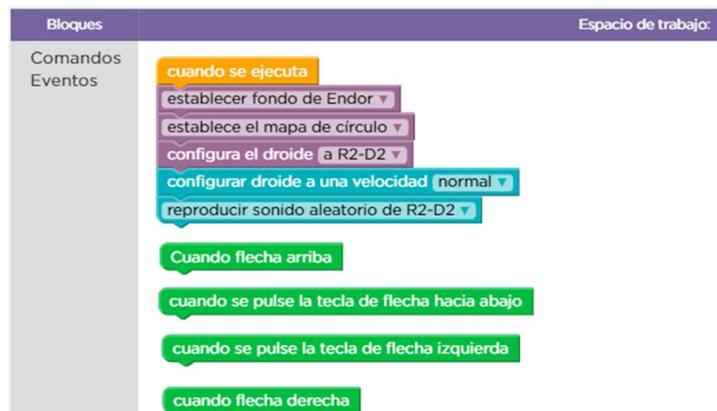
Para resolver el puzzle es necesario realizar lo siguiente:

En las instrucciones podrás leer lo siguiente: "Estás solo en esto, R2-D2", quizá suene un poco drástico, pero realmente no es así, tienes todas las herramientas y recursos para crear lo que consideres de acuerdo con tu aprendizaje adquirido en esta lección. Todo dependerá de tu imaginación.

Utiliza los Eventos y comandos que se han desbloqueado en este ejercicio para crear algo increíble. Cuando hayas terminado, presiona el botón Finalizar para continuar.

Analiza los bloques existentes, y de

acuerdo a las instrucciones para resolverlo, agrega los elementos que consideres necesarios. Puedes agregar los bloques que desees para mejorar o enriquecer tu programa. Prueba agregando diferentes bloques de evento con diversas acciones.



El escenario de este puzzle ya cuenta con varios elementos que puedes usar como punto de partida, así que manos a la obra.

Una vez que termines de programar, verifica los resultados y cuando estés satisfecho de lo que diseñaste, da clic en el botón terminar.



Concluido el reto realiza lo siguiente:

1. Abre un nuevo documento de texto.
2. Toma una impresión o captura de pantalla del área del código donde se puedan apreciar los bloques utilizados, así como la sección del escenario y pégalo en el documento.
3. En la pantalla de CODE presiona el botón

Terminar
4. Copia el enlace de tu juego en el documento.
5. Guarda el documento con el nombre actividad12_tusiniciales, por ejemplo "actividad12_JGAG".
6. Cierra el archivo.
7. Envíalo a través de la plataforma para su revisión.

Aspectos para considerar en la evaluación de la actividad:

1. Considerando que no existe un objetivo específico para resolver este ejercicio, tu juego debe incluir lo siguiente:

- Evento donde aparezca un objeto determinado.
 - Evento donde puedas acumular puntos.
 - Evento donde al capturar un objeto realice alguna acción como aparecer otro objeto o quitar puntos.
 - Utiliza el evento “cuando tocó el obstáculo”
2. Ejecuta tu programa las veces que consideres necesario, realiza los ajustes que creas pertinentes.

Recomendaciones:

Revisa la retroalimentación de la actividad en la plataforma, por si existe alguna observación o sugerencia de tu tutor.

Con este puzzle, se da por concluido este Módulo, continuemos con las actividades de este diplomado.

Módulo 3

Primeros pasos

Open Roberta es una plataforma en línea, cuyo nombre proviene de un proyecto dentro de la iniciativa educativa alemana "Roberta: Aprender con robots", iniciado por Fraunhofer IAIS, que es un instituto perteneciente a la Fraunhofer Society. Con Open Roberta, se busca alentar a los niños a codificar mediante el uso de robots como Lego Mindstorms y otros sistemas de hardware programables como Arduino, BBC micro: bit y Calliope mini.

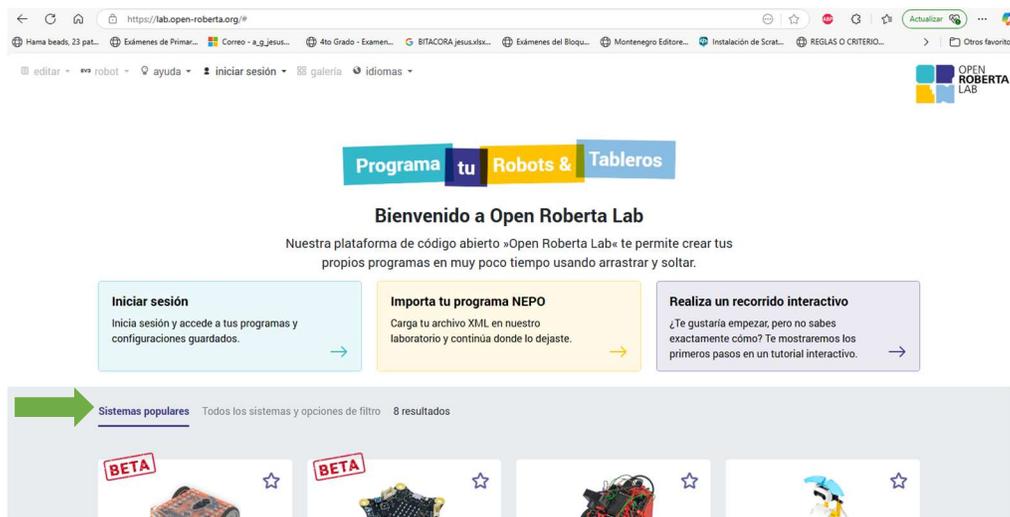
El enfoque en la nube del Open Roberta Lab tiene como objetivo simplificar los conceptos de programación y facilitar que los profesores y las escuelas enseñen a codificar.

En esta guía describiremos los pasos para programar y mostraremos algunos ejemplos de códigos de programación de robots, haciendo semejanza a la programación del robot LEGO MINDSTORMS.

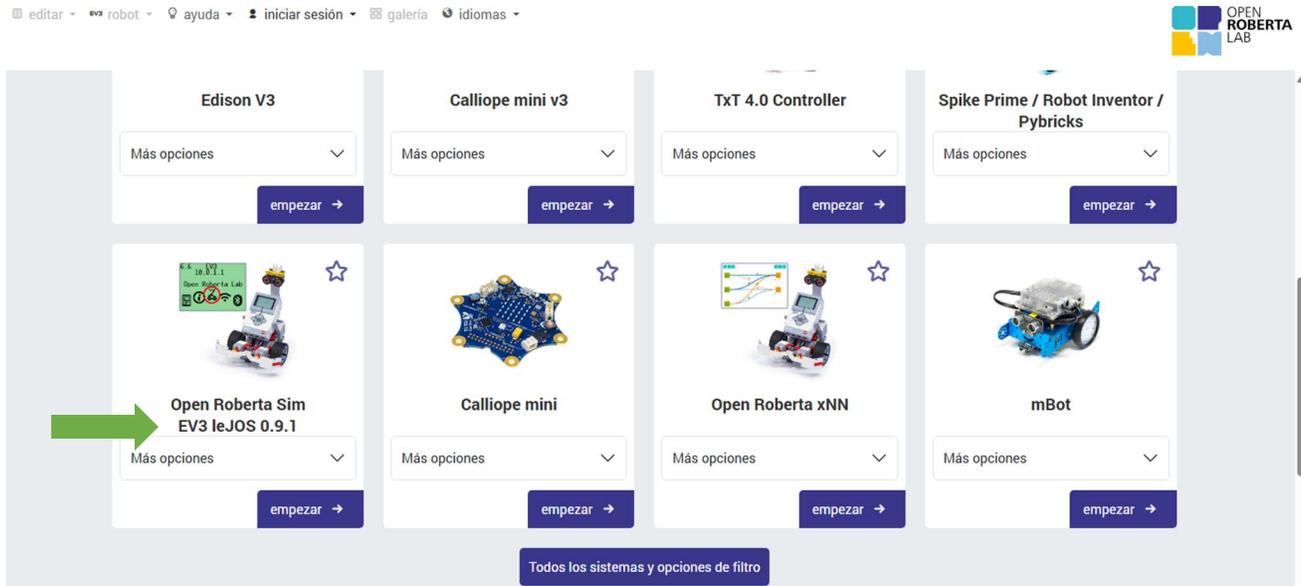
Ingreso a la aplicación

Para iniciar en Open Roberta, debes ingresar a tu buscador en internet (Google, Edge, etc.), y escribir OPEN ROBERTA, o bien colocar la siguiente dirección: <https://lab.open-roberta.org/>

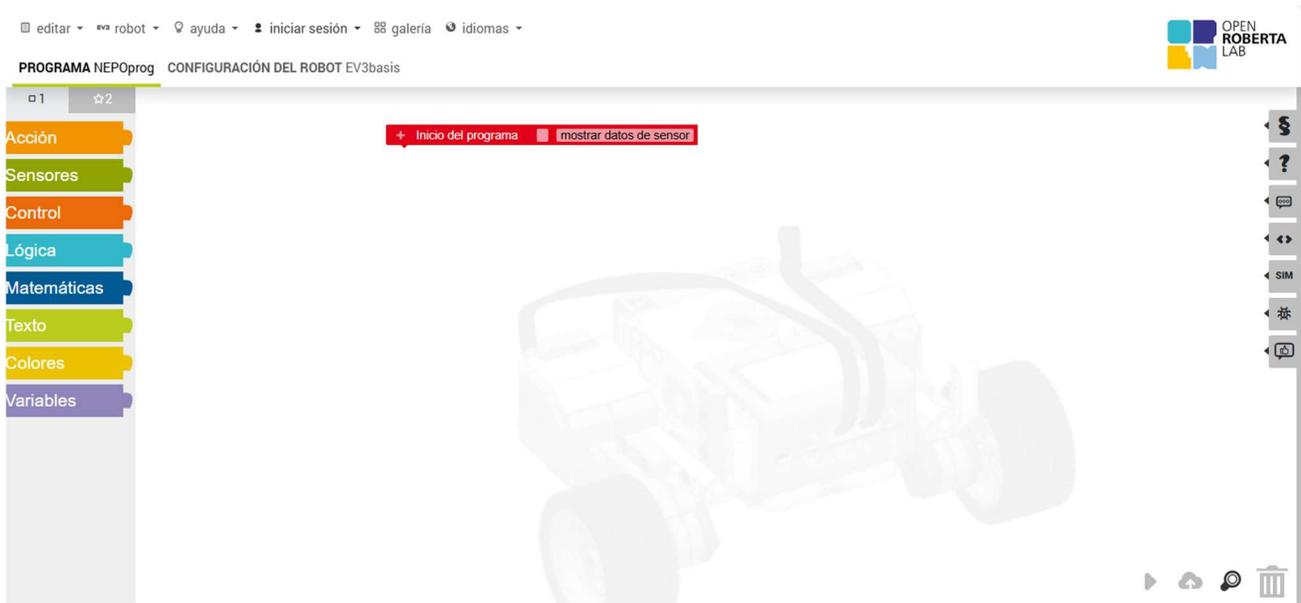
Una vez que ingreses, verás una pantalla como la que se muestra a continuación, en la cual verificamos que esté seleccionada la opción "Sistemas populares".



Desplázate un poco hacia debajo de la pantalla hasta que visualices el modelo “Open Roberta Sim EV3 leJOS 0.9.1”, el cual te muestro en la imagen siguiente:

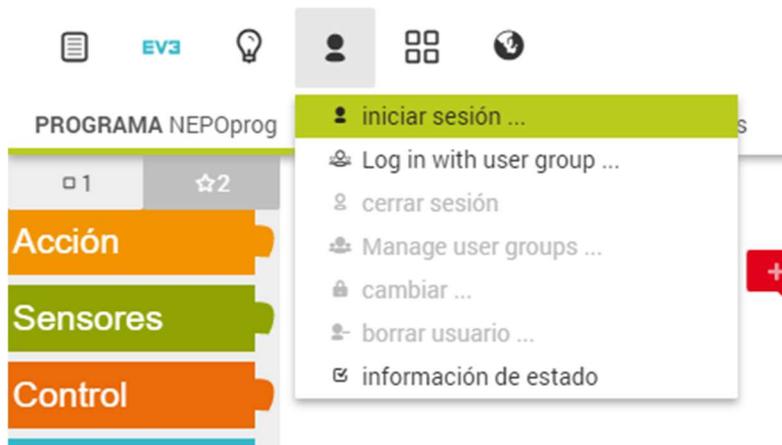


Una vez que lo ubiques, da clic en la imagen para ingresar al simulador, donde podrás comenzar a programar.



Registro en Open Roberta

Para poder hacer uso de todas las herramientas de Open Roberta, es recomendable que crees una cuenta, para ello da clic en la opción  usuario posteriormente en iniciar sesión.



Da clic en “nuevo”.

Nombre de usuario



Contraseña



OK

resetear  nuevo ...

Rellenar información del formulario de registro

nuevo ...

Nombre de usuario

¿»!AmBotman« o »RobellaStracciatella«? No todos necesitan saber tu nombre real. Piensa en un apodo genial que puedas recordar fácilmente.

Contraseña

Repetir contraseña

Nombre

Correo electrónico

[Registrar ahora](#)

[iniciar sesión ...](#)

[resetear password ...](#)

La información requerida en el formulario es muy sencilla, además te proporciona algunos tips que te pueden servir para completar la información.

Nombre de usuario: el nombre que mostrará en la plataforma.

Contraseña que usarás para ingresar a la plataforma. Debes incluir mayúsculas, minúsculas y números.

Repetir contraseña: debe de ser la misma contraseña anterior.

Nombre: Debes escribir tu nombre personal.

Correo electrónico: debes escribir un correo personal al que tengas acceso.

Después de escribir el correo te preguntará la edad. En nuestro caso seleccionaremos la segunda opción.

Correo electrónico

@ eqvd2019.04@gmail.com

Edad

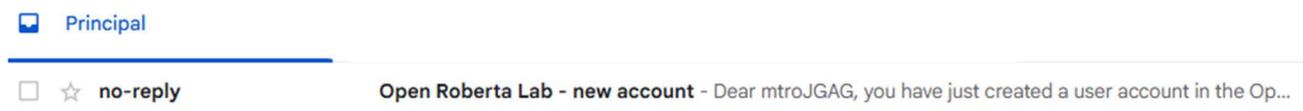
@ Tengo 16 años o más!

¿Tienes 16 años o más?
Entonces, ¿quieres que tus padres que te ayuden. Pueden especificar su dirección de correo electrónico para confirmar tu cuenta.

[Registrar ahora](#)

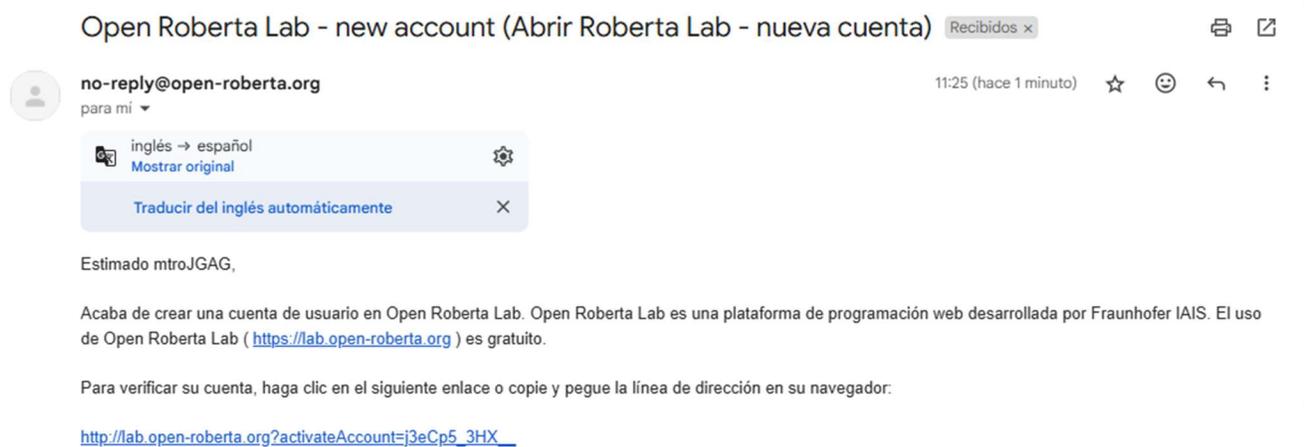
Ahora da clic en el botón “Registrar ahora”.

Una vez que finalices el registro debes revisar tu correo electrónico para activar tu cuenta en Open Roberta Lab, y buscar en tu buzón de entrada (o no deseado) el que contenga el asunto “**Open Roberta Lab - new account**”. El correo que te envía la plataforma debe ser como el que te muestro a continuación:



Ábrelo, es un correo en inglés, si lo quieres traducir utiliza las opciones del navegador para traducción. Fuera de esto no tendrás ningún inconveniente, solo da clic en el enlace o

hipervínculo que dice: “<https://lab.open-roberta.org#activate...>”, el cual te llevará a la pantalla de inicio de sesión en Open Roberta Lab, donde escribirás el nombre de usuario que creaste y la contraseña.



Open Roberta Lab - new account (Abrir Roberta Lab - nueva cuenta) Recibidos x 🖨️ 📧

 **no-reply@open-roberta.org** 11:25 (hace 1 minuto) ☆ 😊 ↩️ ⋮
para mí ▾

 inglés → español
Mostrar original ⚙️
Traducir del inglés automáticamente ✕

Estimado mtroJGAG,

Acaba de crear una cuenta de usuario en Open Roberta Lab. Open Roberta Lab es una plataforma de programación web desarrollada por Fraunhofer IAIS. El uso de Open Roberta Lab (<https://lab.open-roberta.org>) es gratuito.

Para verificar su cuenta, haga clic en el siguiente enlace o copie y pegue la línea de dirección en su navegador:

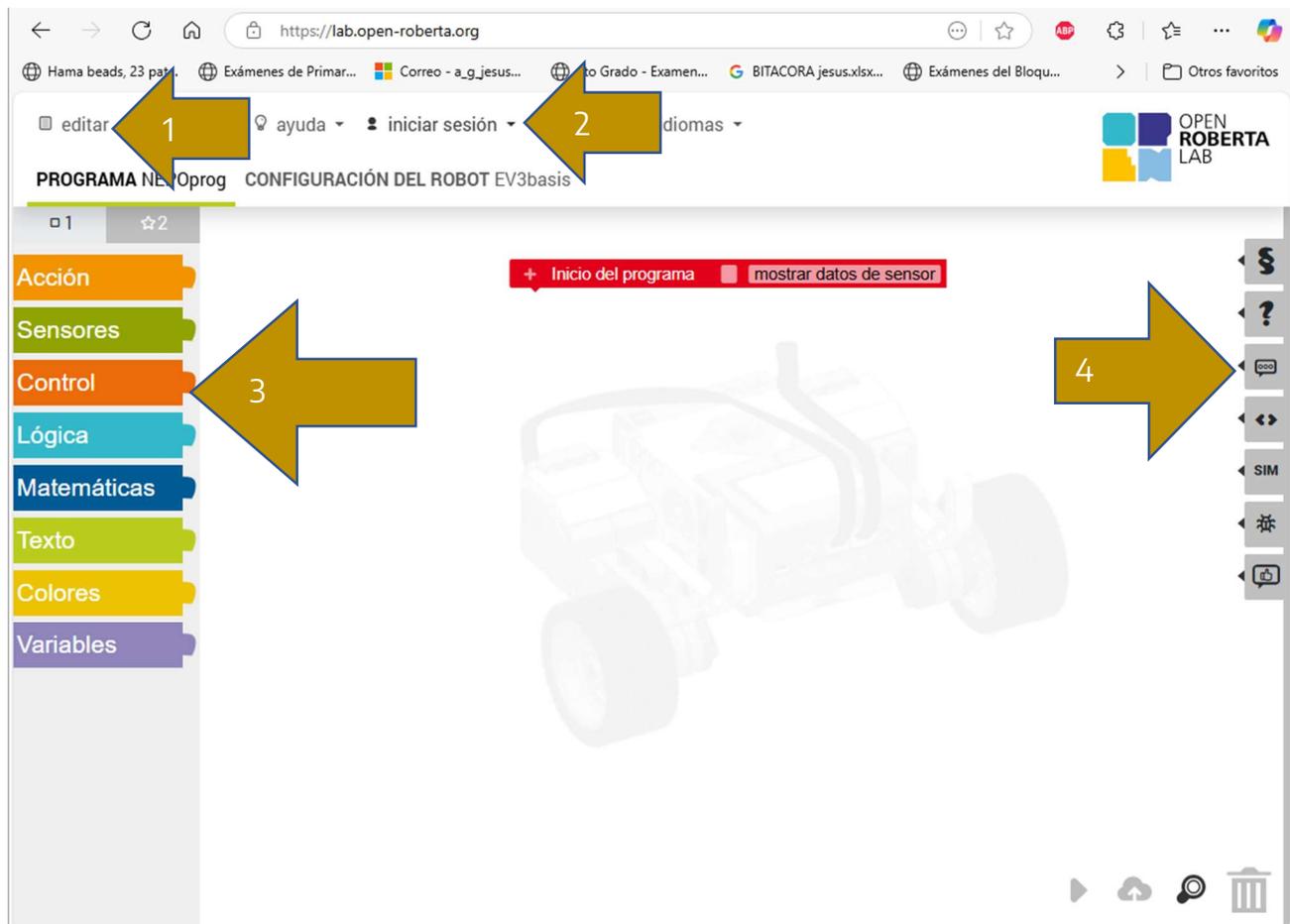
http://lab.open-roberta.org?activateAccount=j3eCp5_3HX

Entorno de la aplicación

Una vez que ingresas tu usuario y contraseña te mostrará la pantalla principal, es similar a la que se muestra sin que inicies sesión, la diferencia es que podrás ver y tendrás disponibles muchas más opciones del simulador.

La pantalla principal de Open Roberta Lab, se encuentra dividida en varias secciones, te mostraré algunas:

1. Editar, menú del programa
2. Iniciar sesión
3. Bloques de programación
4. Menú lateral



Editar, menú del programa

Las opciones de este menú son: editar, robots, ayuda, usuario, galería e idioma.

A continuación, describiré brevemente algunas de las opciones de este menú.

El menú “editar” cuenta con las siguientes opciones:

ejecutar en ladrillo

Con un clic en este elemento del submenú, inicia el programa en el robot. Su programa y la configuración del robot se enviarán al servidor de Open Roberta. Allí se genera el código de la máquina y se envía al robot. El robot inicia el programa tan pronto como recibe el programa completo o cuando lo inicia manualmente.

En este curso solo utilizarás las opciones del simulador para probar los programas que realices.

ejecutar en simulación

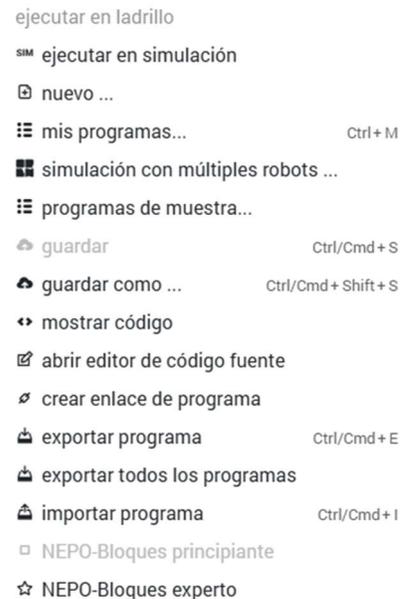
Con un clic en este elemento del submenú, inicia el programa en el entorno de simulación, que es donde realizaremos las actividades de esta parte de nuestro diplomado. Su programa se enviará al Open Roberta Simulator. La configuración del robot del robot simulado se basa en la configuración predeterminada y no se puede cambiar por el momento.

nuevo

Con este botón puede crear un nuevo programa y empezar desde cero.

mis programas

Aquí se abre una nueva sección y se muestran todos sus programas guardados. Para ello, debes iniciar sesión o crear una nueva cuenta de usuario en "usuario".



guardar como

Si has iniciado sesión, puedes guardar tu programa en este menú con un nombre (único), el que tú decidas. Si ese nombre de programa ya lo has usado, la aplicación te mandará una alerta o indicará que ya ha sido utilizado ese nombre para otro programa.

exportar programa

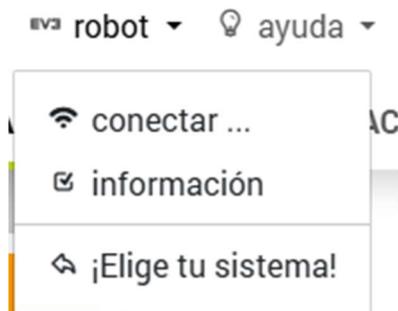
El programa actual se exporta a un archivo XML y se guarda en un medio de almacenamiento, este puede ser en su disco duro o una memoria USB.

importar programa

Un programa se importa desde un archivo XML, es decir, se carga desde un medio de almacenamiento. Solo puedes importar programas para el sistema de robot seleccionado actualmente. Si el programa no coincide con el sistema de robot, Open Roberta Lab te mandará un mensaje de error y le dirá para qué sistema de robot se creó este programa.

Existen otras opciones en este menú, pero con las que se te han mencionado anteriormente podrás empezar a trabajar con Open Roberta Lab.

Menú del Robot



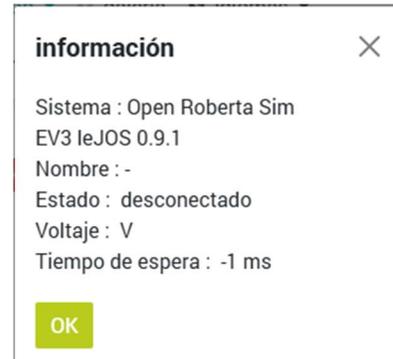
Aquí puedes elegir entre todos los robots disponibles en Open Roberta Lab.

En este menú cuentas con tres opciones que son: conectar, información y ¡Elige tu sistema!

En este curso trabajarás con el

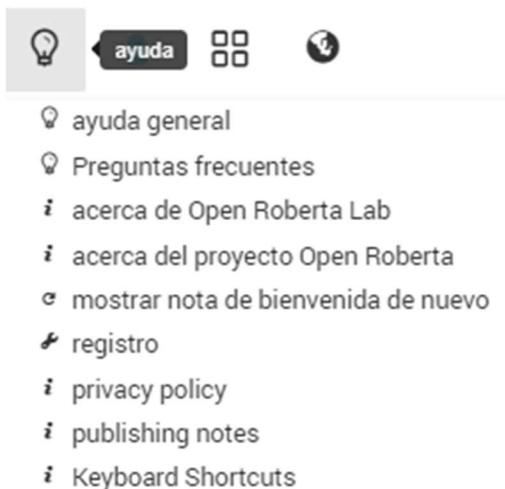
modelo "Open Roberta Sim

EV3 leJOS 0.9.1". En otro momento, si lo deseas, podrás utilizar otros modelos y generar los programas para hacer funcionar el robot seleccionado.



Recuerda que en este curso solo utilizarás el simulador para probar los programas que diseñes.

Ayuda



En el menú de ayuda podrás encontrar algunas categorías que te permitirán ubicar o seleccionar el tópico o tema de ayuda que requieras o que necesites consultar.

Iniciar sesión

Con este menú puede iniciar o cerrar sesión en Open Roberta Lab o crear una nueva cuenta de usuario, eliminar una existente. Los siguientes submenús están disponibles para este propósito:

- 👤 iniciar sesión ...
- 👤 Iniciar sesión con el grupo de usuarios .
- 👤 cerrar sesión
- 👤 Gestionar grupos de usuarios ...
- 🔒 cambiar ...
- 👤 borrar usuario ...
- 📄 información de estado

Iniciar sesión

Inicia sesión con una cuenta de usuario existente. Si aún no tienes una cuenta, puedes crear una nueva. Para iniciar sesión en tu cuenta, debes usar tu nombre de usuario, que elegiste al crear tu cuenta de Open Roberta Lab. Si olvidas tu nombre de usuario, puedes encontrarlo en la dirección del correo electrónico para restablecer su contraseña.

Cerrar sesión

Cerrar la cuenta de usuario que tienes activa. Recuerda que si no inicias sesión puedes crear tus programas y transferirlos al robot seleccionado.

Idioma



Deutsch
English
Svenska
Español
Português

Otra de las opciones que puedes visualizar en el menú del programa, es la opción para seleccionar idioma. Para hacerlo solo es necesario que des clic en el ícono del “mundo” y selecciones el idioma que prefieras, en este caso ya he seleccionado “Español”.

Bloques de programación y configuración del robot



PROGRAMA NEPOprog

CONFIGURACIÓN DEL ROBOT EV3basis

Bajo los íconos del menú de programa descritos anteriormente (editar, menú de robot, ayuda, etc.) se encuentran los menús de “PROGRAMA”, que te permitirán ver los bloques de programación, así como el menú de “CONFIGURACIÓN DEL ROBOT”.

Programa

PROGRAMA NEPOprog Esta opción te permite ver los bloques que te servirán para elaborar tus programas, ya te comentaré como los puedes usar.

Los bloques de programas se encuentran divididos en dos secciones: principiantes y avanzados.



Los bloques para “principiantes” se pueden identificar por el número 1, y los bloques para avanzados por el número 2.

 Bloques principiantes.

 Bloques avanzados.

En el bloque avanzados, se incluyen algunas categorías de programación, que más adelante podrás conocer.

Las categorías de los bloques de programación se describen brevemente a continuación:

Categoría	Notas
Acción	Incluye bloques para que el robot se realice directamente.
Sensores	Contiene bloques para todos los sensores estándar del sistema EV3.
Controlar	Incluye bloques para el control de la secuencia del programa. La categoría incluye los siguientes bloques: If .. do, If .. do .. else, repetir indefinidamente, repetir .. veces, esperar .. ms, esperar hasta ..
Lógica	Con los bloques «lógicos» se pueden crear condiciones. Con esta condición, puede interrelacionar estados, valores y eventos entre sí.
Matemáticas	Operadores matemáticos y bloques de parámetros.
Lista	Incluye bloques para crear una lista y buscar u ordenar elementos de la lista.

Categoría	Notas
Texto	Incluye bloques para escribir cadenas en la pantalla del robot.
Colores	Bloques de colores estándar para comparar las entradas de los sensores.
Variables	Se pueden definir variables locales y globales.
Funciones	Se pueden definir funciones con parámetros de entrada y salida.
Mensajes	Incluye bloques para enviar y recibir mensajes bluetooth.
inicio del programa	Cada programa comienza con este bloque. Este bloque siempre está disponible en el espacio de trabajo.

Un poco más adelante trabajaremos a con estos bloques de programación.

Configuración del robot

En esta sección podrás modificar la configuración “física” del robot, es decir podrás colocar y quitar sensores, tal y como se observan en las imágenes siguientes.



EV3

diámetro de la rueda 5.6 cm
 distancia entre ruedas 18 cm

Sensor 1 sensor de contacto
 Sensor 2 giroscopio
 Sensor 3 sensor de color
 Sensor 4 sensor de ultrasonidos

Motor A
 Motor B grande motor
 regulación sí
 dirección de la rotación hacia delante
 lado derecho

Motor C grande motor
 regulación sí
 dirección de la rotación hacia delante
 lado izquierdo

Motor D

Para colocar o cambiar un sensor, solo activa el bloque de sensores, selecciona el que necesites y conéctalos al puerto o sensor que lo requieras.

EV3

diámetro de la rueda 5.6 cm
 distancia entre ruedas 18 cm

Sensor 1 sensor de contacto
 Sensor 2 giroscopio
 Sensor 3 sensor de color
 Sensor 4 sensor de ultrasonidos

Motor A

Menú lateral



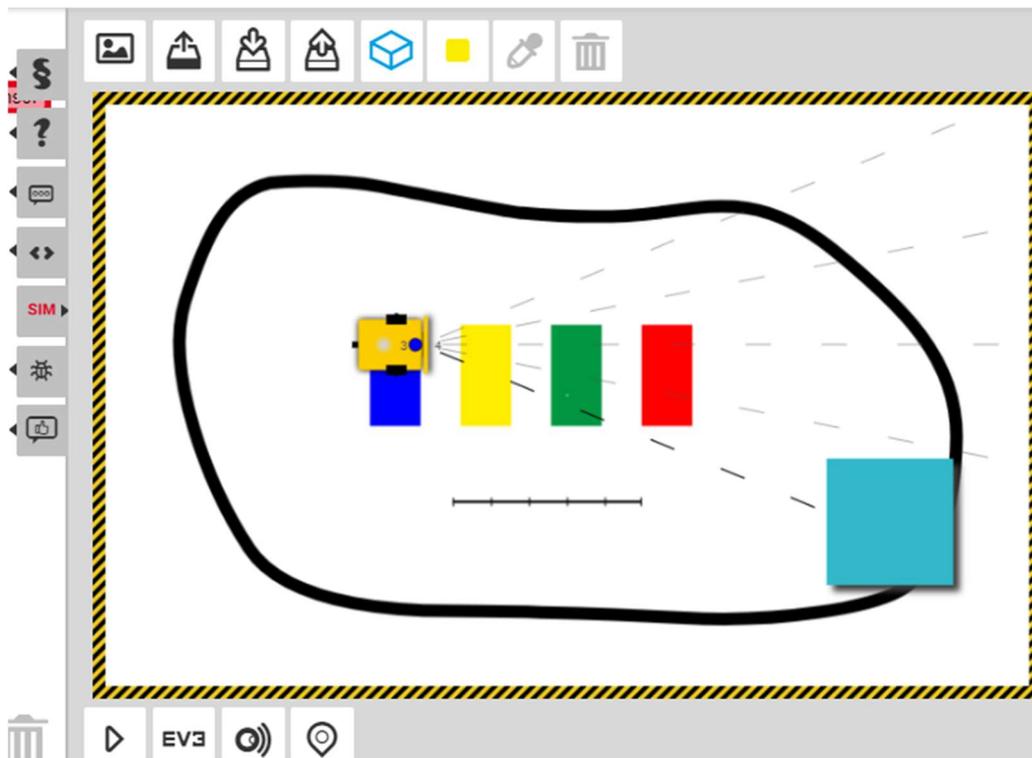
Como en los menús descritos anteriormente, este cuenta con diferentes opciones, como ayuda, código fuente, etc.

La opción que más utilizarás en este curso será la del simulador “SIM”.

Para activarlo solo es necesario que des clic en el ícono



Una vez que activas el simulador, se visualiza la siguiente pantalla, donde la parte principal es el escenario, donde se ejecuta el programa que diseñes para tu robot. Describiré brevemente algunas de las opciones que dispones en el simulador.





Cambiar el escenario, al dar clic en este botón, puedes cambiar dentro de los diferentes escenarios disponibles. Toma en cuenta que el escenario que selecciones debe ser acorde al programa creado.



Cargar un nuevo escenario, puedes crear un escenario, cargarlo y usarlo en el simulador para crear nuevos retos.



Agregar un obstáculo, puedes agregar un obstáculo al escenario que estás utilizando en ese momento.



Eliminar objeto o elemento, con esta herramienta podrás eliminar obstáculos del escenario.

En la parte inferior del escenario, puedes encontrar los siguientes botones:



Ejecutar el programa, da clic en este botón para ejecutar el programa. (Play)



Ver u ocultar el ladrillo del robot activo.



Ver u ocultar la ventana de datos de los sensores del robot.



Al dar clic en este botón, el robot se moverá a la posición de inicio o predeterminada en el escenario.

Mi primer programa. El robot avanza hacia adelante

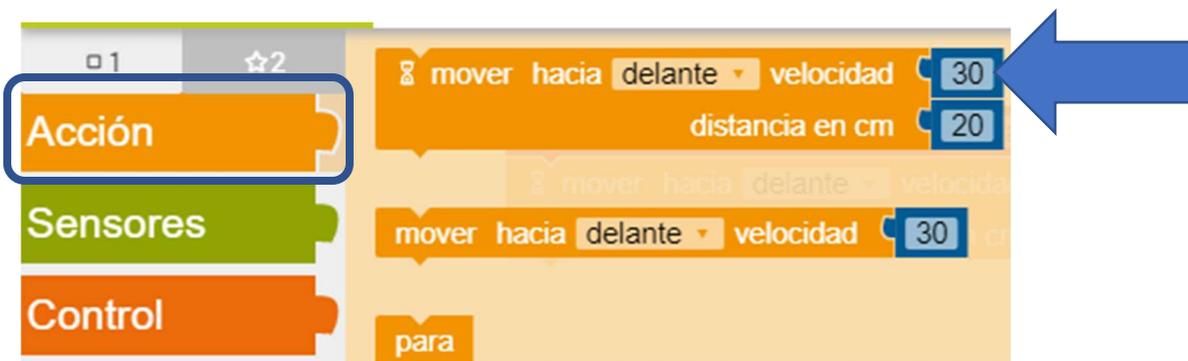
La recomendación para empezar a trabajar con OpenRoberta es que ya hayas creado tu cuenta de usuario e inicies sesión.

En este primer programa el robot debe moverse 40 cm hacia adelante, a una velocidad de 100 m/seg. Para resolver este reto, elaboraremos primero un algoritmo, el cual es el siguiente:

1. Mover el robot hacia adelante, velocidad de 100 metros por segundo, recorriendo una distancia de 40 cm.
2. Esperar 1000 milisegundos (1000 milisegundos = 1 segundo)

Después de revisar que el algoritmo resuelva el reto, es momento de que realices tu primer programa, para ello realiza lo siguiente:

Selecciona la categoría “Acción”, posteriormente el bloque “mover hacia” y distancia en cm.



Muévelo hacia el área de programación. Asegúrate que esté conectado al bloque “Inicio del programa” para que este se pueda ejecutar correctamente.



Modifica los parámetros de “velocidad” a 100 y “distancia en cm” a 40. Para ello de clic dentro del bloque del valor y modifícalo.



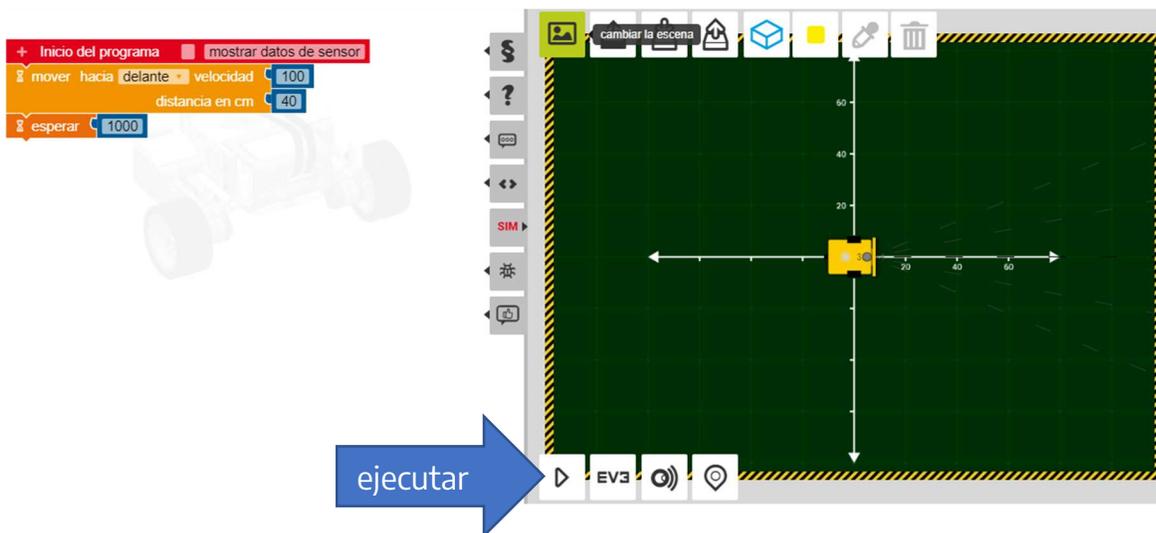
Para agregar la instrucción “esperar” da clic en la categoría “Control”, selecciona el bloque.



Colócalo en el área de programación, asegúrate que esté conectado después del bloque “mover hacia”. Modifica el valor del bloque “esperar” a 1000. Este valor se da en milisegundos, mil milisegundos equivalen a un segundo.



Para probar el programa, activa el simulador y da clic en el botón “cambiar la escena” hasta que se visualice el escenario que se muestra en la imagen siguiente. Da clic en el botón “ejecutar” o play para probar el código del programa.



Mi segundo programa – avanzar y retroceder

Debido a que en Open Roberta se programa de forma secuencial, vamos a ir conectando los bloques de *avanzar*, *esperar*, *retroceder* y nuevamente un bloque de *esperar* al final.

El robot se debe mover a una velocidad de 30 m/s, recorrer una distancia de 20 cm, esperar un segundo, regresar 20 cm a 30 m/s y esperar otro segundo.

El algoritmo para resolver este reto sería el siguiente:

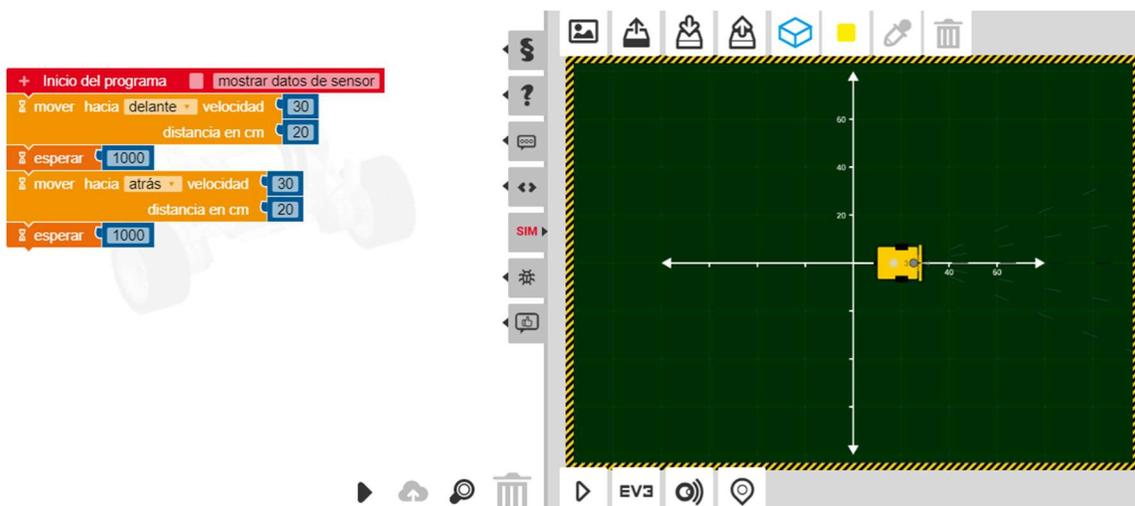
1. Mover el robot hacia adelante, velocidad de 30 y distancia de 20cm
2. Esperar 1000 milisegundos (1000 milisegundos = 1 segundo)
3. Mover el robot hacia atrás, velocidad de 30, distancia de 20cm
4. Esperar 1000 milisegundos (1000 milisegundos = 1 segundo)

El código para resolver el ejercicio, de acuerdo al algoritmo sería el siguiente:

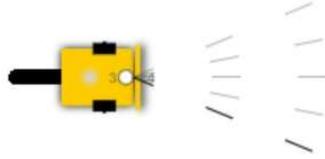


Ahora utilizando los boques, construye el programa y verifica que cumpla con lo requerido.

Como puedes observar, utilizo el mismo escenario para probar el programa que acabo de crear.



Si utilizas otro escenario, es probable que el robot que vaya dejando una línea que muestra su trayectoria.



Realizar giros

Hasta este momento solo has realizado programas que mueven el robot hacia adelante o hacia atrás. Sin embargo, puedes hacer que tu robot realice giros a la derecha o izquierda y en diferentes grados. Para hacer que tu robot gire puedes utilizar el bloque “girar – velocidad – grado”, del grupo de comandos “Acción”.



Para realizar algunos ejercicios con estos comandos, utilizaremos el siguiente escenario:

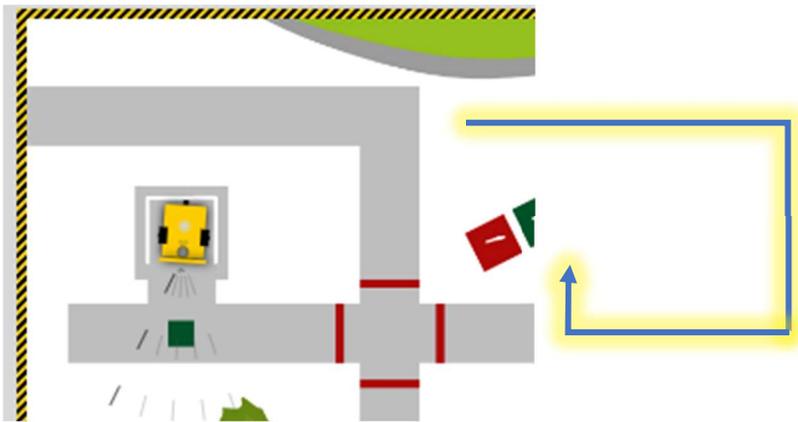


Utiliza el botón para cambiar de escenario para seleccionar el escenario que se muestra en la imagen.

Reto: el objetivo de este reto es que el robot siga el camino gris hasta la zona de parking, debiendo quedar debidamente estacionado en la zona asignada para estacionarse, con la parte frontal del robot hacia el sur.

El algoritmo puede ser el siguiente:

1. Mover hacia adelante el robot "x" distancia
2. Girar a la derecha el robot 90°
3. Mover hacia adelante el robot "x" distancia
4. Girar a la derecha el robot 90°
5. Mover hacia adelante el robot "x" distancia
6. Girar a la derecha el robot 90°
7. Mover hacia adelante el robot "x" distancia
8. Girar el robot "x" grados para que quede con la parte frontal hacia el sur.



Para a resolver el reto crea un programa nuevo



y tomando como referencia el algoritmo, coloca los siguientes bloques:



Después de agregar los bloques, activa el simulador y ejecuta el programa. El robot debe llegar hasta la posición que se muestra en la siguiente imagen:



Una vez que hayas logrado que el robot llegue a la posición, guarda el programa con el nombre “Giros”.

Hasta aquí se ha resuelto una parte del reto, es tu turno de resolver el ejercicio, toma como referencia lo realizado hasta el momento y el algoritmo que se realizó previamente. Esto lo realizarás en la sesión del día de mañana.

Actividad 3. Giros

Como ya te lo he mencionado, es turno que completes el programa, toma como referencia el algoritmo y el código creaste anteriormente (Giros).

Con el código que se generó hasta el momento, el robot debe de estar en la posición que se muestra en la imagen siguiente:



Debes llevarlo a la posición de Parking marcado con la “P” en el escenario.

Como sugerencia, puedes resolver paso por paso los puntos que faltan utilizando el algoritmo, aquí te lo vuelvo a mostrar. Identifica en que parte del algoritmo para que puedas continuar elaborando el programa.

1. Mover hacia adelante el robot “x” distancia
2. Girar a la derecha el robot 90°
3. Mover hacia adelante el robot “x” distancia
4. Girar a la derecha el robot 90°
5. Mover hacia adelante el robot “x” distancia
6. Girar a la derecha el robot 90°
7. Mover hacia adelante el robot “x” distancia

8. Girar el robot “x” grados para que quede con la parte frontal hacia el sur. Recuerda que en los mapas el Norte queda en la parte superior.

Indicaciones:

1. Abre tu navegador y ve a la página de la aplicación Open Roberta, inicia sesión.
2. Abre el programa “Giros” que creaste anteriormente.
3. Completa el programa tomando como referencia el algoritmo.
4. Una vez que esté resuelto todo el programa, verifica que cumpla con lo requerido, guárdalo como “Actividad4Giros”.
5. Realiza una impresión de pantalla del reto finalizado, cópialo en un documento nuevo de Word.
6. Guarda el documento de Word con el nombre “Actividad4Giros”_nombre, por ejemplo “Actividad4Giros_JesusAlvarado”. Guárdalo sin las comillas
7. Cierra el documento de Word.
8. Envía el archivo a través de la plataforma (SET-Capacitacion) para su revisión.
9. Después de esto continúa con las actividades del curso.

Recomendación: ve resolviendo el ejercicio un paso a la vez, guarda el programa cada vez que resuelvas un punto.

Los bloques mostrar texto y girar

Vamos ahora a incrementar los movimientos de nuestro robot con dos bloques nuevos de programación. El bloque mostrar texto se utiliza para que el robot “escriba” en la pantalla del ladrillo un texto breve, esta información puede ser relacionada con lo que hace el robot en ese momento.

El bloque GIRAR, como ya lo has visto, se utiliza para poder girar el robot un determinado número de grados. En el siguiente programa harás que el robot se mueva hacia adelante, gire 90°, muestre el texto: “hola, he girado 90 grados a la derecha y que al final espere por 1 segundo.

El programa lo debes elaborar tomando como base el siguiente algoritmo:

1. Mover hacia adelante 20 cm a una velocidad de 30 m/s
2. Girar a la derecha 90 grados
3. Mostrar texto “hola, he girado 90 grados a la derecha”.
4. Esperar dos minutos
5. Avanzar

Lo primero que debes hacer es crear un nuevo programa



Agrega los siguientes bloques:



Tanto el bloque “gira” y “mostrar texto” se encuentran en la categoría “Acción”.

El bloque “gira” cuenta con tres parámetros:

- El primero indica hacia donde realizará el giro el robot, derecha o izquierda.



- El segundo parámetro indica la velocidad a la que se desplazará el robot, por default es 30.



- El tercer parámetro tiene que ver con el giro, para ello debes escribir un número que represente los grados, en el ejercicio tomaremos 90.



En este programa se usa también la instrucción “mostrar texto”, que también cuenta con tres parámetros, los cuales te describo brevemente a continuación.

- Mostrar texto, en este parámetro colocarás el texto que quieras se despliegue en el ladrillo o robot.

mostrar texto “ hola, he girado 90 ”

- En columna, se usa un valor numérico para indicar en que columna del display empezará a mostrar el texto.

en columna 0

- En fila, a partir de que fila se despliega el texto.

en fila 0

Para este ejemplo se usan dos bloques “mostrar texto”, ya que la capacidad del display o pantalla del ladrillo es limitada.

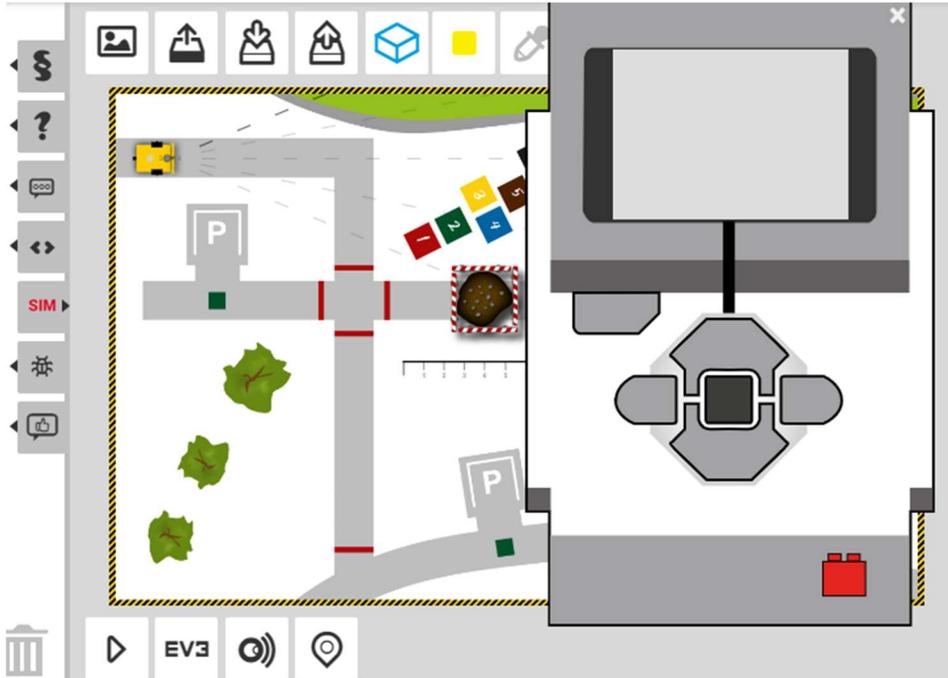
Una vez que hayas agregado los bloques que se mencionan anteriormente, es hora de probar el código para ver cómo funciona. Para ello, utilizando la opción del simulador “cambiar la escena” activa la que se usó en el programa “giros”, se muestra en la siguiente imagen una parte del escenario.



Ahora, da clic en el botón “abrir/cerrar la vista del robot”.

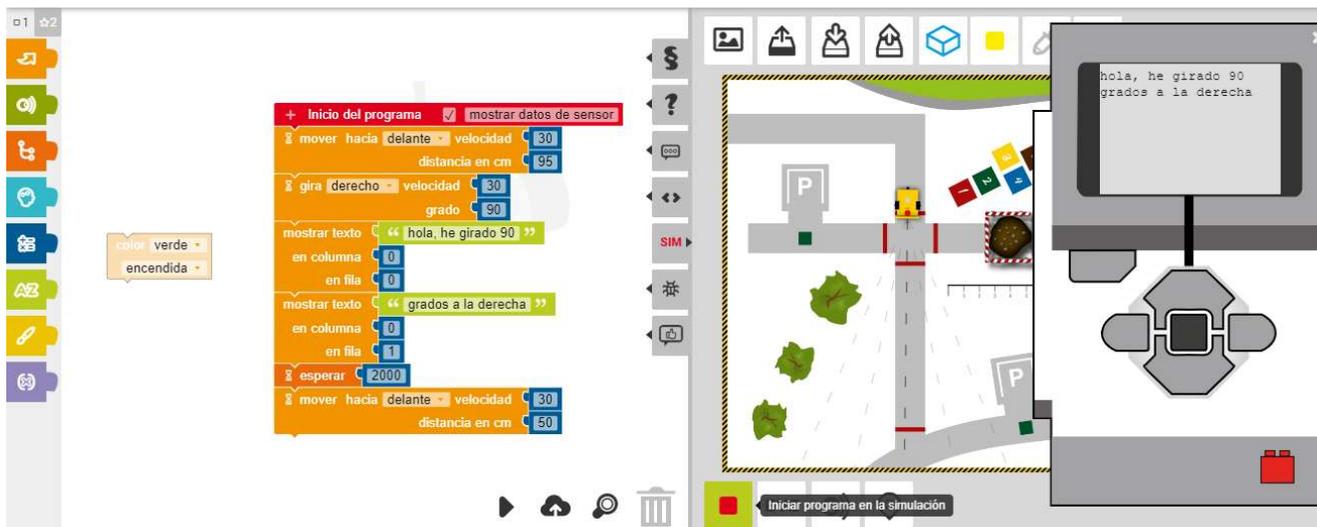


Al activar esta opción, puedes visualizar el ladrillo o cerebro del robot, esto servirá para verificar que se despliegue el texto que se indica en el programa, así como el resto de las instrucciones del programa.



Da clic en el botón “reproducir” para ejecutar el programa.

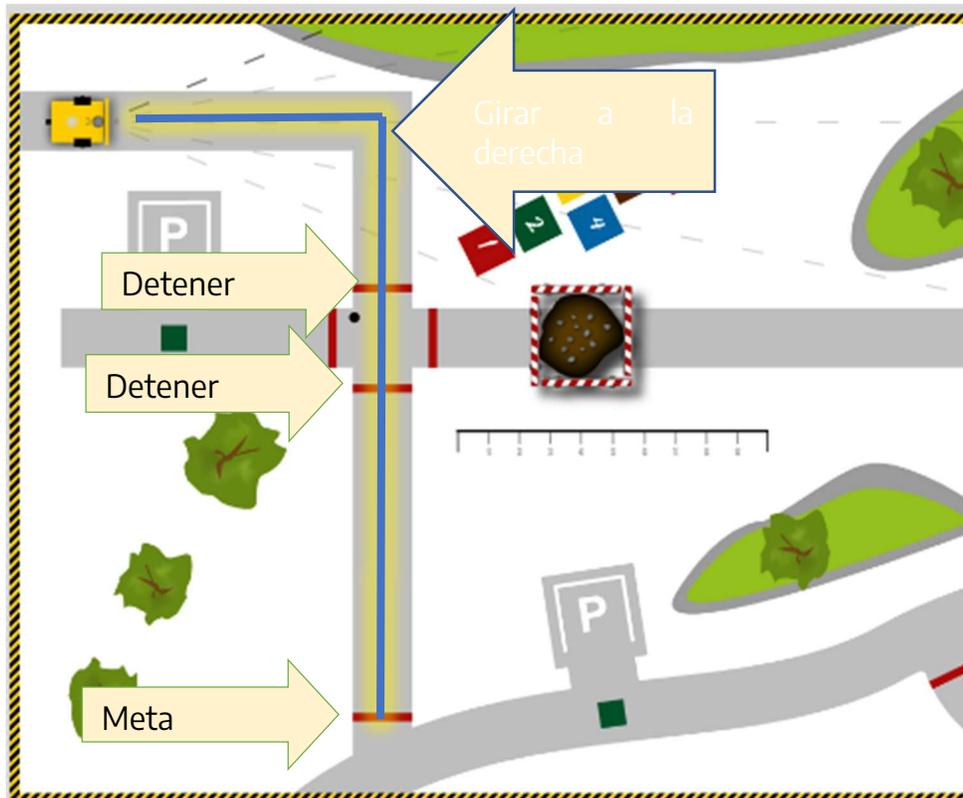
En la imagen siguiente puedes ver el resultado del programa, así como el texto se despliega en el display del ladrillo.



Verifica que tu programa funcione adecuadamente, si no es así verifica el código y de ser necesario, realiza las correcciones que se requieran. Una vez que esté funcionando correctamente, guárdalo con el nombre “TextoGiros”.

Actividad 4. Mostrar texto y girar.

Utiliza el escenario del programa anterior. En el escenario se muestra la ruta que debe seguir el robot, así como lo que debe de hacer en algunas estaciones. Agrega bloques “mostrar texto” en cada una de las “estaciones” que se indican, estos bloques deben mostrar información referente a la acción que realiza el robot, tales como girar, detener, o fin del programa.



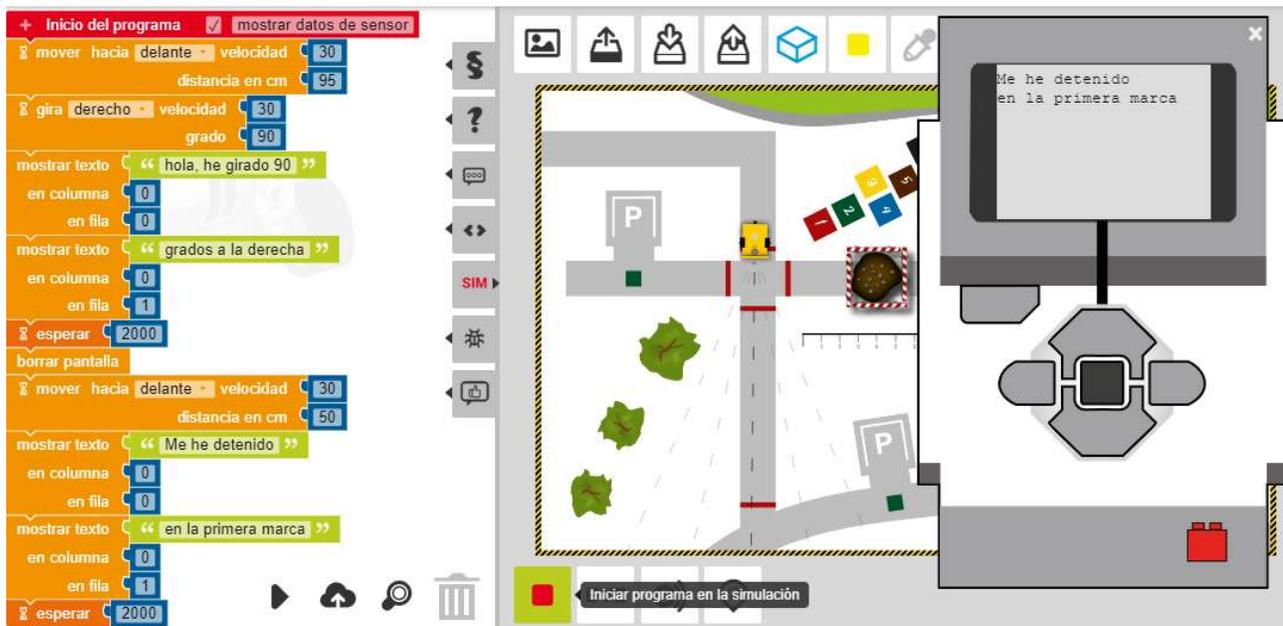
Tomando como referencia lo aprendido en los ejercicios anteriores realiza el siguiente ejercicio

Indicaciones:

- Debes crear un programa que utilice la instrucción BUCLE para hacer que el robot dibuje un cuadrado.
- Elabora el algoritmo que te permita resolver el ejercicio.
- La velocidad a la que se mueva el robot debe ser 30 m/seg, distancia recorrida 30 cm, la figura que debe formar, como ya se mencionó, es un cuadrado. Con base a esta sugerencia selecciona la instrucción a utilizar.
- Utiliza el siguiente escenario para probar el programa que crees.

- Una vez que hayas resuelto el ejercicio guárdalo con el nombre “ActividadGirarTextos”. (sin comillas)
- Realiza una impresión o captura de pantalla donde se puedan observar los bloques utilizados, el escenario y el "ladrillo" o robot; posteriormente abre un nuevo documento de Word y pega la imagen.
- Guarda el documento con el nombre “Actividad4_GirarTextos”_tunombre, por ejemplo: “Actividad4_GirarTextos_JesusAlvarado”. (sin comillas)
- Envíalo a través de esta plataforma.

En la siguiente imagen te dejo como pista una parte del programa que puedes usar como base para crear el tuyo.



El bucle

Un bucle o ciclo, es una secuencia de instrucciones que se ejecuta repetidas veces, hasta que la condición asignada a dicho bucle deja de cumplirse. En open Roberta, puedes utilizar el bucle de forma INFINITA o puedes indicar que se repite un determinado número de veces. Los tipos de bucles que puedes utilizar en esta aplicación son los siguientes bloques o instrucciones:



- repetir indefinidamente
- repetir “n” veces
- repetir “hasta”
- contar con i desde 1 hasta 10 de a 1

entre otras instrucciones o bloques de programación, en este curso se usarán principalmente las dos primeras opciones.

A continuación, veremos ejemplo donde el robot debe repetir cinco veces una serie de instrucciones, que son: moverse hacia adelante y decir uno, esperar un segundo, moverse hacia atrás y decir cero, esperar un segundo.

El algoritmo que se puede usar para resolver el ejemplo y crear el código del programa que use un bucle, repetido cinco veces, será el siguiente:

1. Repetir cinco veces
 - 1.1. El robot se moverá hacia enfrente
 - 1.2. Escribe “Adelante”
 - 1.3. Esperará 1 segundo
 - 1.4. Se moverá hacia atrás
 - 1.5. Escribe “Atrás”
 - 1.6. Esperará 1 segundo

Nota: Observa que se usa en el algoritmo una numeración 1.1, 1.2... esto te puede servir para indicar el uso de bucles.

Los bloques de programación quedarían de la siguiente forma:

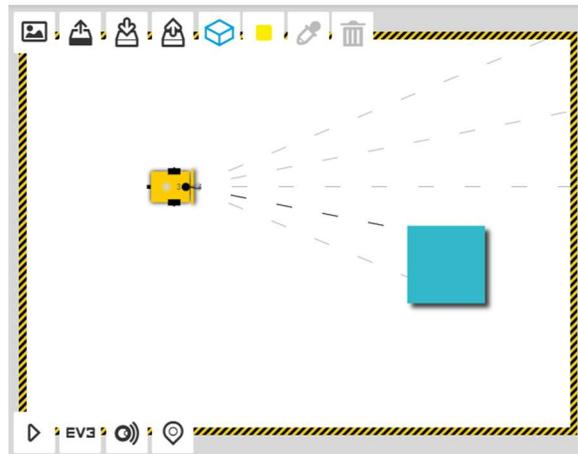


Nota: se agrega el bloque de instrucción “borrar pantalla” para eliminar el texto que se imprime con la instrucción “mostrar texto”.

Actividad 5. Bucles

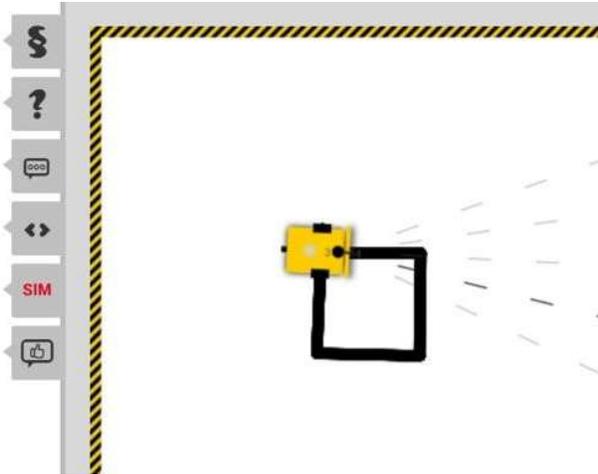
Tomando como referencia lo aprendido las sesiones anteriores, realiza el siguiente ejercicio

- Indicaciones:
- Debes crear un programa que utilice la instrucción BUCLE para hacer que el robot dibuje un cuadrado.
- Elabora el algoritmo que te permita resolver el ejercicio.
- La velocidad a la que se mueva el robot debe ser 30 m/seg, distancia recorrida 30 cm, la figura que debe formar, como ya se mencionó, es un cuadrado. Con base a esta sugerencia selecciona la instrucción a utilizar.
- Utiliza el siguiente escenario para probar el programa que crees.



- Una vez que hayas resuelto el ejercicio guárdalo con el nombre “ActividadBucles”. (sin comillas)
- Realiza una impresión o captura de pantalla donde se pueda observar el escenario y el código utilizado para resolver la actividad, abre un nuevo documento de Word y pega la imagen.
- Guarda el documento con el nombre “Actividad5_Bucles”_tunombre, por ejemplo: “Actividad5_Bucles_JesusAlvarado”. (sin comillas)
- Envíalo a través de esta plataforma.

Nota: recuerda que el robot pintará automáticamente la ruta que siga.

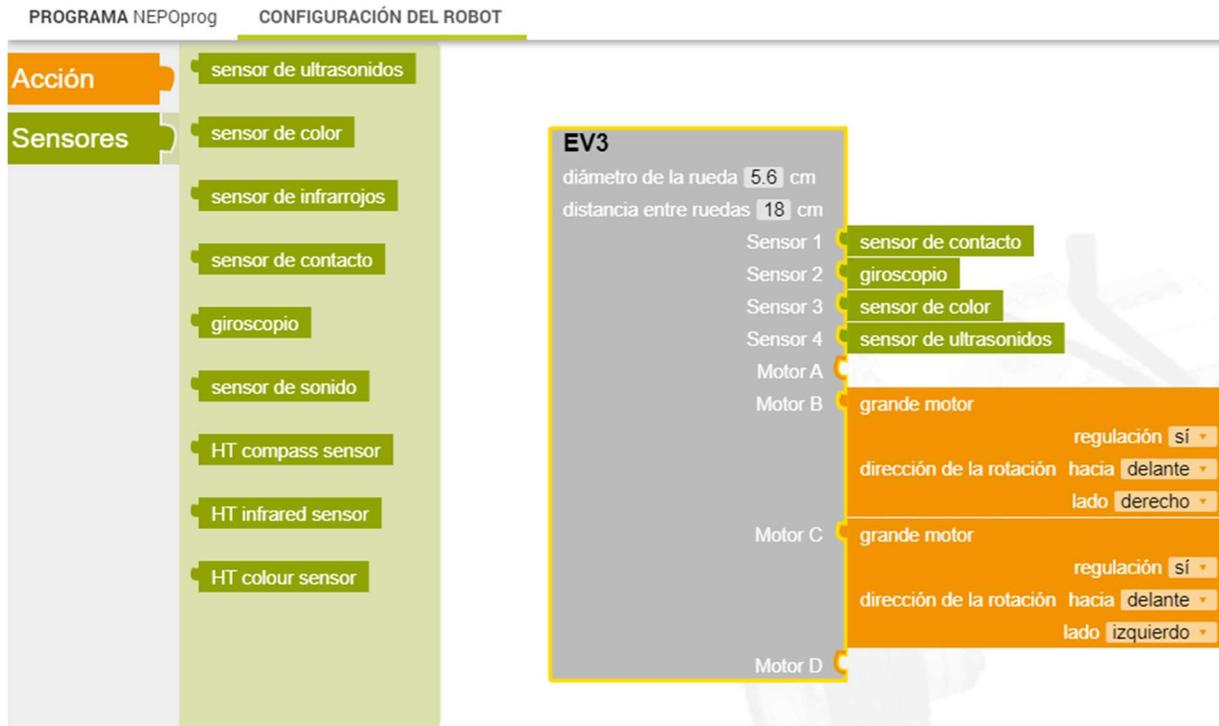


El resultado de tu programa debe de ser similar al de la imagen de la izquierda.

Una vez que tu programa cumpla con lo requerido, guárdalo con el nombre “Actividad5_Bucles”_tunombre, por ejemplo: “Actividad5_Bucles_JesusAlvarado”.

Sensores

El modelo de robot utilizado en este curso (EV3) cuenta con sensores que le permiten determinar cuando hay un objeto cerca, o pasó o está sobre un color, entre otro tipo de alertas, estas alertas le permiten tomar una decisión. Estos sensores, así como los motores que usa el robot se configuran desde la opción CONFIGURACION DEL ROBOT.



Para utilizar los sensores al programar, se utiliza la instrucción “esperar hasta”, y posteriormente se elige el sensor requerido, esto dependerá del propósito del desafío o la tarea a realizar.



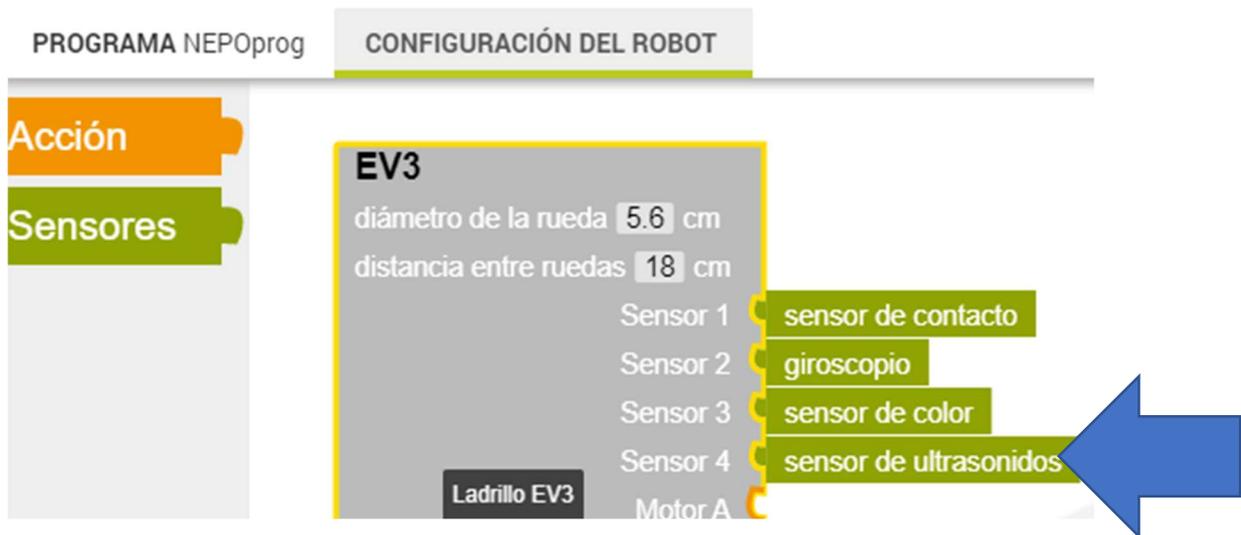
El robot EV3 tiene capacidad para conectar simultáneamente cuatro sensores, si requieres de uno diferente a los que se encuentran conectados, debes desconectar uno de los que se encuentran conectados en el ladrillo del robot y después seleccionar uno de los sensores disponibles para el robot.

El sensor ultrasónico (detecta objetos)

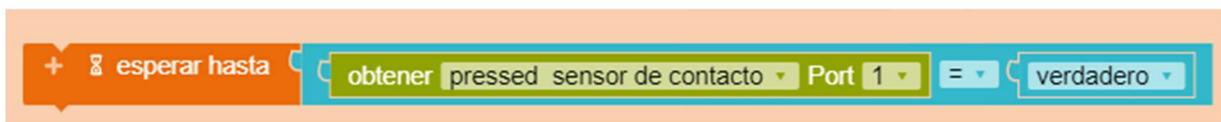
Si deseas crear un programa para tu robot en el que este detecte objetos, puedes crearlo utilizando el sensor ultrasónico. Este sensor funciona de la siguiente manera: una vez que el robot va avanzando, puede detectar un objeto a cierta distancia, cuando esto suceda le decimos que se detenga.

Para realizar un ejemplo, utilizando el sensor ultrasónico, es necesario que consideres lo siguiente:

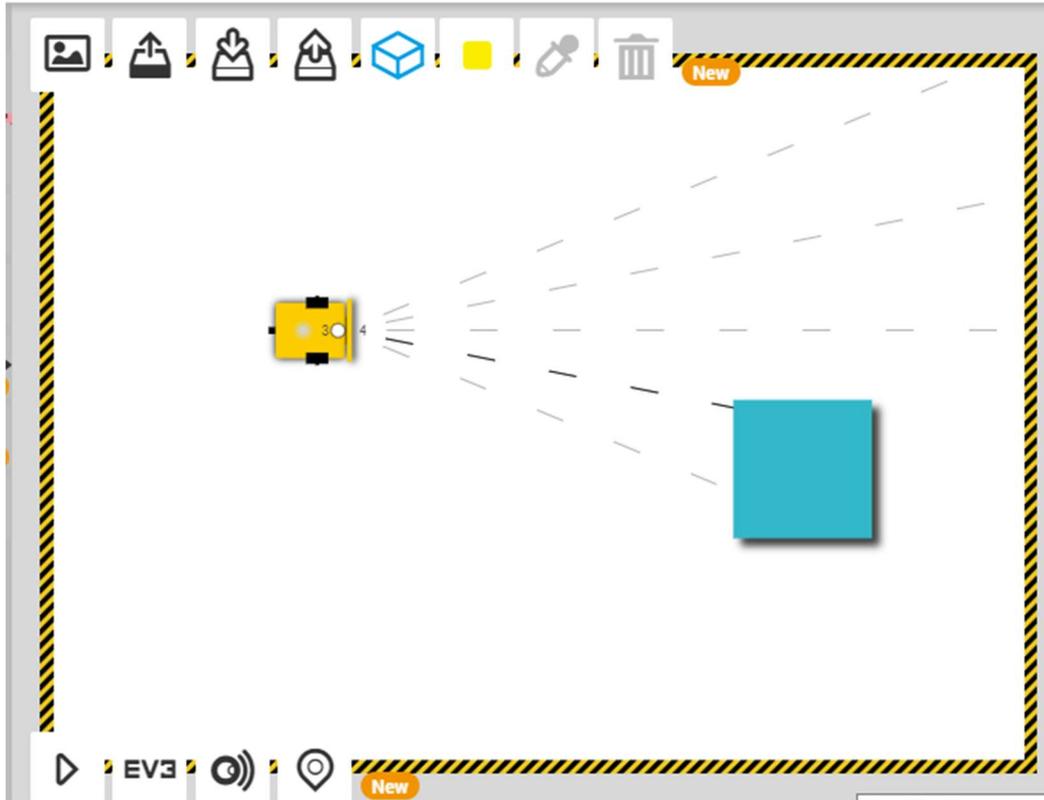
- Revisa en la configuración del robot que el “Sensor 4” tenga seleccionado al “sensor de ultrasonidos”, recuerda que esto se hace en la opción de “CONFIGURACIÓN DEL ROBOT”.



- Debes utilizar la instrucción “esperar hasta”, que se encuentra en la categoría “control”.



- Seleccionar la escena que tiene un cuadro, en la siguiente imagen puedes visualizarlo.

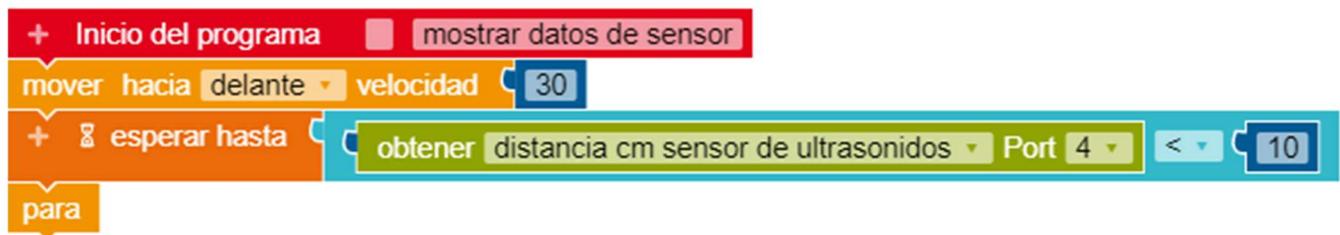


El algoritmo que puedes utilizar para crear el programa sería el

siguiente:

1. El robot se mueve hacia adelante
2. Al detectar un objeto a menos de 10 cm se detiene
3. Si no hay un objeto a menos de esa distancia se sigue moviendo.

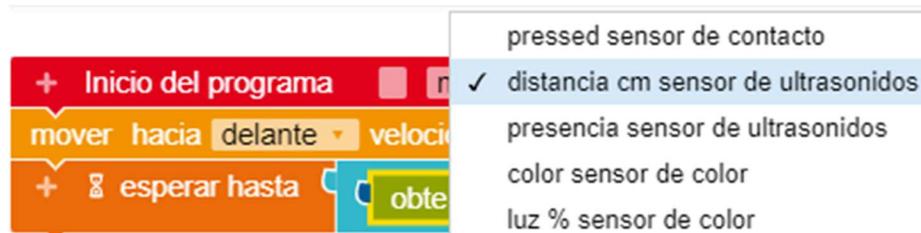
El programa generado de acuerdo con el algoritmo sería el siguiente:



Para indicar que sensor utilizar, en la instrucción “esperar hasta” da clic en “obtener”

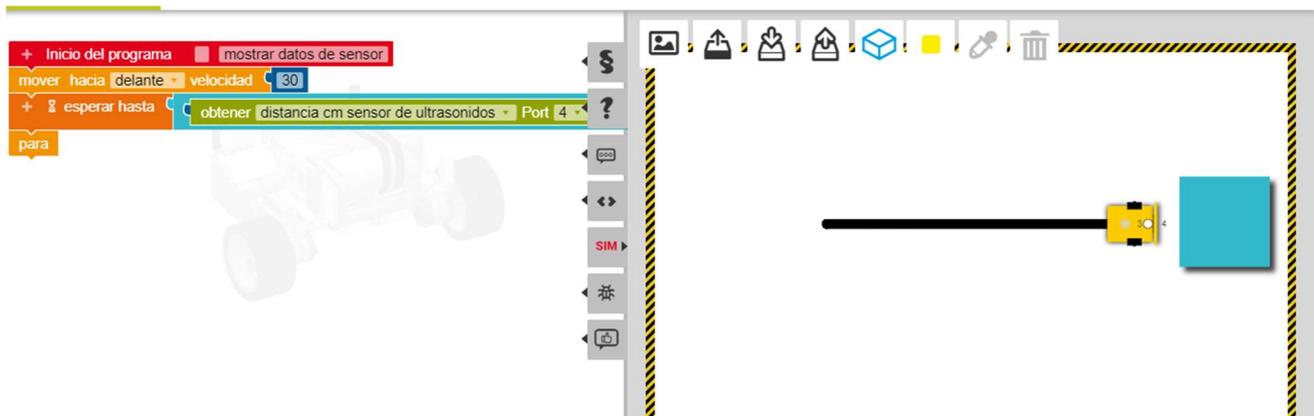


y selecciona la opción “distancia cm sensor de ultrasonidos”.



Copia el programa, tomando como referencia el código anterior, para probarlo en el simulador no olvides seleccionar el escenario recomendado. Una vez que actives el escenario, coloca el cuadro frente el robot y ejecuta el programa, observa los resultados.

El resultado debe ser similar al presentado en la imagen siguiente:

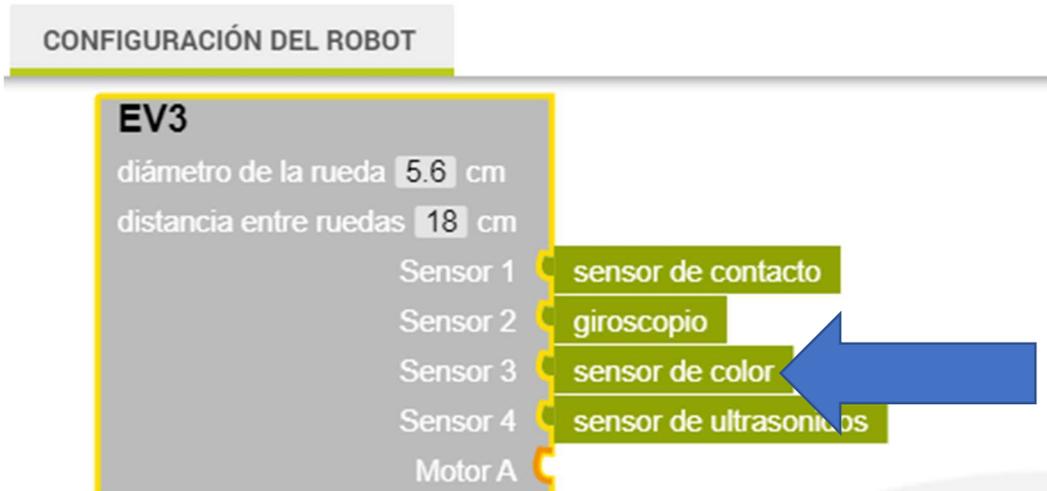


Al detectar un objeto a menos de 10 cm el robot debe detenerse. En caso de que no se detenga, revisa el código de tu programa y vuelve a intentarlo.

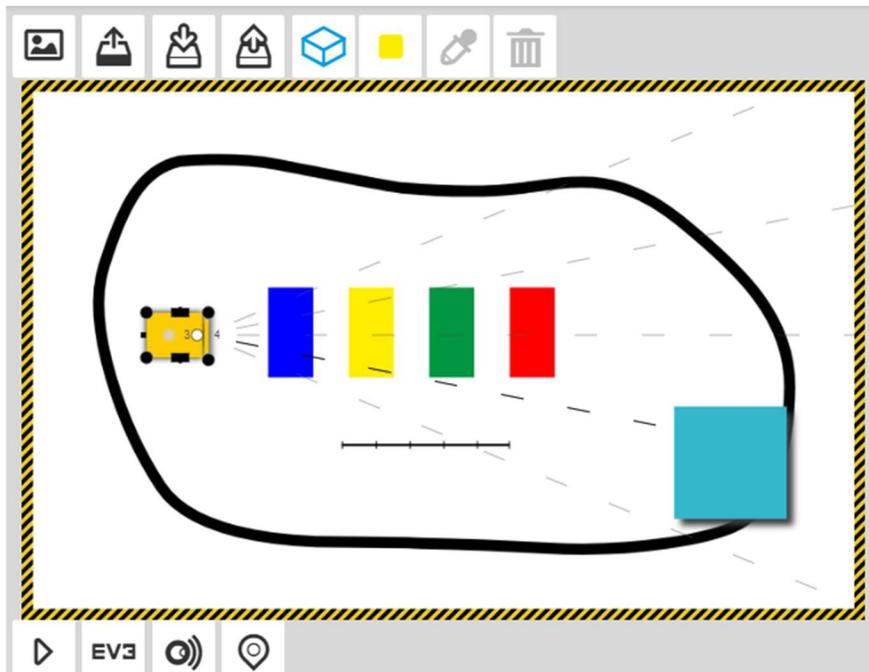
Una vez que tu programa cumpla con lo requerido, guárdalo con el nombre “ActSensorUltrasonico”

El sensor de color

El sensor de color detecta 7 diferentes colores: rojo, azul, verde, amarillo, café, negro y blanco. Para usar el sensor de color, verifica que esté conectado al puerto 3.



Ahora utilizando el sensor de color realizarás un programa en el que harás que el robot avance, y cuando detecte el color seleccionado, se detenga. Para realizar este ejercicio selecciona el escenario que se muestra en la imagen inferior. Por default, el robot está colocado sobre el color azul, muévelo a la posición que se muestra.



Para crear el programa se utilizan las instrucciones “mover hacia adelante”, “mostrar texto”, “esperar” y “esperar hasta”.

El algoritmo que puedes utilizar para resolver este ejercicio es:

1. Avanzar
2. Buscar color azul
3. Si detecta color azul muestra el texto “color azul detectado” y esperar un segundo.

El código para resolver el ejercicio es el siguiente:



Explicación del programa:

mover hacia adelante velocidad 30 Mover hacia “adelante” esta instrucción se encuentra dentro del bloque “Acción”, le indica al robot que debe moverse hacia adelante a una velocidad de 30 m/seg.

La instrucción “esperar hasta”, como su nombre lo indica, esperará hasta obtener o detectar un valor **esperar hasta** obtener color sensor de color Port 3 = color, dependiendo del sensor que se tenga configurado, en este caso se utiliza el sensor de color.

Para seleccionar al sensor de color da clic en “obtener” y selecciona el tipo de sensor que necesites, en este caso es “color sensor de Verifica que esté habilitado el puerto “3” tres, condición sea igual “=” y color “azul”.

distancia cm sensor de ultrasonidos
 presencia sensor de ultrasonidos
color sensor de color
 luz % sensor de color
 luz ambiente % sensor de color
 distancia cm sensor de infrarrojos

color”.
 que la



La siguiente instrucción que se utiliza en este programa es “mostrar texto”, la información o texto se mostrará en el display o pantalla del

ladrillo. Para ingresar la información que deseas mostrar da clic en el recuadro verde e ingresa el texto, se sugiere que sean frases cortas de 4 o 5 palabras máximo por línea. Puedes indicar la ubicación del texto con base a la columna y línea en la que deseas que se muestre el texto.

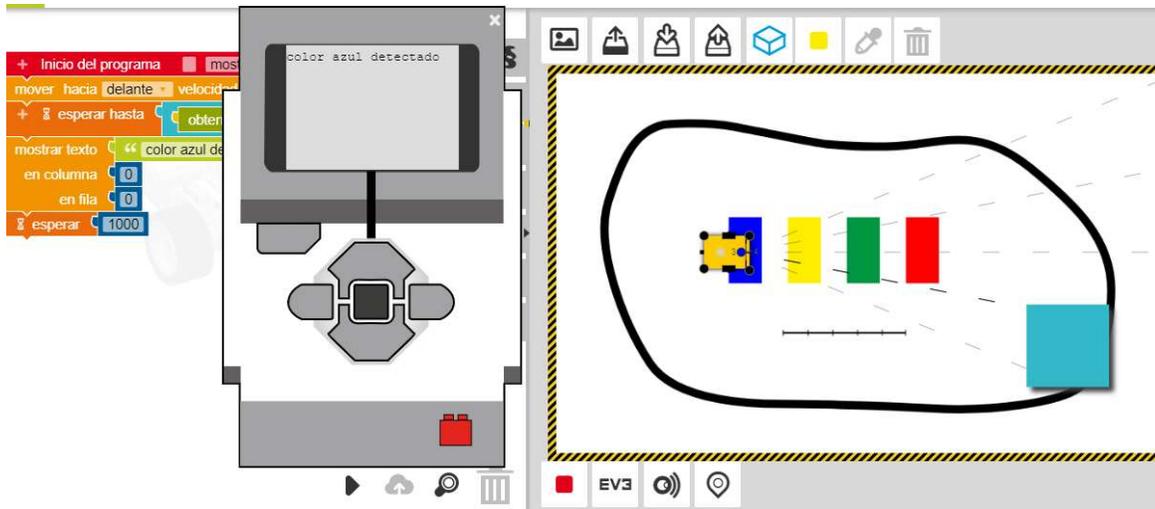


La última instrucción que se utiliza en este programa es “esperar”, la cual indica una pausa en el desarrollo de un programa, el valor que se ingresa es en milisegundos, mil milisegundos equivalen a un segundo.

Ahora ejecuta el programa y verifica los resultados, de ser necesario realiza los ajustes correspondientes.

No te olvides de habilitar la opción para visualizar el ladrillo (EV3) que se encuentra en la parte inferior del simulador, para poder ver el texto en el display del ladrillo.

Guarda el archivo con el nombre “ActividadSensorColorAzul”.



Actividad 6. Detectar los cuatro colores usando bucle y condiciones.

Para resolver el siguiente ejercicio debes considerar lo siguiente:

- Que el programa que elabores incluya un bucle.
- Utiliza el escenario del ejercicio anterior.
- Que el robot detecte los cuatro colores.
- Cada que detecte un color despliegue en la pantalla del ladrillo el color detectado, haciendo una pausa de dos segundos.
- Al detectar el último color del escenario, debe detener el programa, después de indicar en la pantalla del ladrillo qué color detectó.

Nota: el último color en detectar será el rojo, es ahí donde se debe “romper” el bucle, para ello agrega la instrucción “romper el bucle”, la cual encontrarás en:

Bloques avanzados

Control

Bucles

romper el bucle



- Una vez que hayas resuelto el ejercicio, guárdalo con el nombre “BucleySensoresColores”.
- A continuación, te proporciono una parte del código para resolver el ejercicio, analízalo y agrega el código faltante para resolver la actividad.



Explicación del programa.

Después del bloque de “Inicio del programa” se conecta el bucle o ciclo “repetir indefinidamente”. Esta instrucción lo que hace es que todo lo que se conecte dentro de él se repita indefinidamente.



El siguiente bloque de instrucción que se utiliza es “mover hacia”, esto hará que el robot se mueva hacia adelante a una velocidad de 30 m/seg.



La siguiente instrucción es una condición “si – hacer”. En este bloque debes de conectar la instrucción que determine la condición, en este caso es obtener el color.

Para agregar la condición al bloque “si – hacer” realiza lo siguiente:

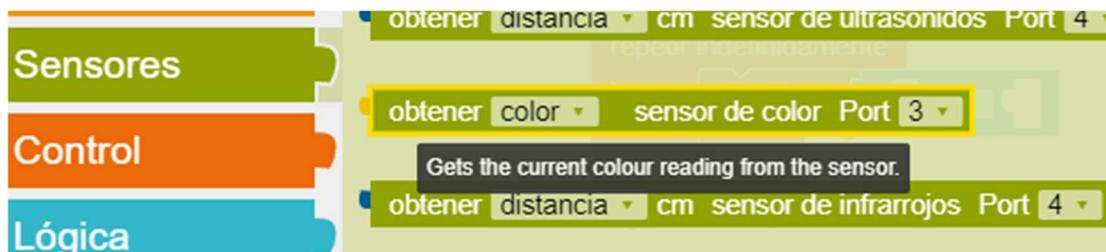
Selecciona la categoría “Lógica” y selecciona el bloque de instrucción para devolver si dos condiciones son verdaderas.



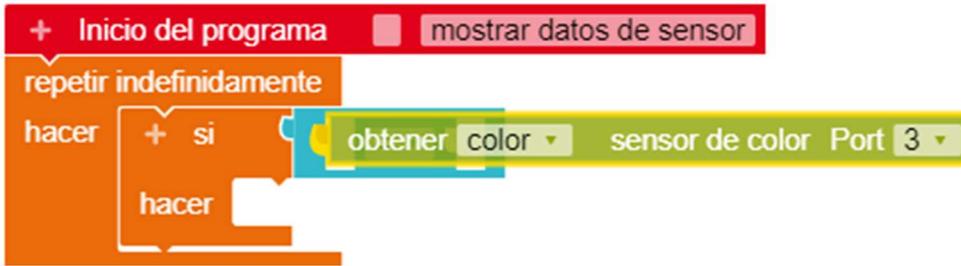
Colócalo en la sección “si”.



Ahora es necesario agregar el primer parámetro para comparar las condiciones, da clic en la categoría “Sensores”, selecciona el bloque de instrucción “obtener color”.



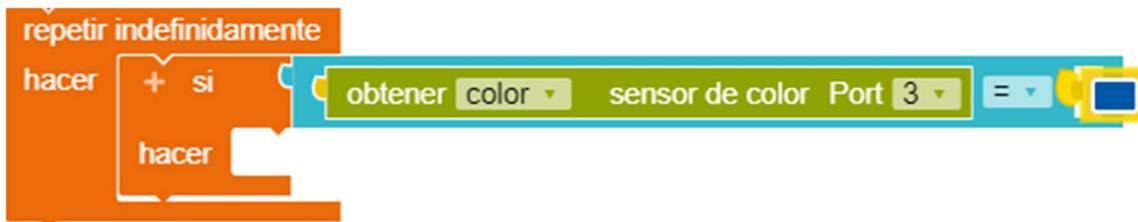
Colócalo en el primer parámetro de la condición



Ahora da clic en la categoría de “Color” y selecciona el color con el que deseas comparar la primera condición, en este caso selecciono el azul.



Colócalo en el segundo parámetro para realizar la comparación.



La condición “si – entonces” debe quedar de la siguiente manera:

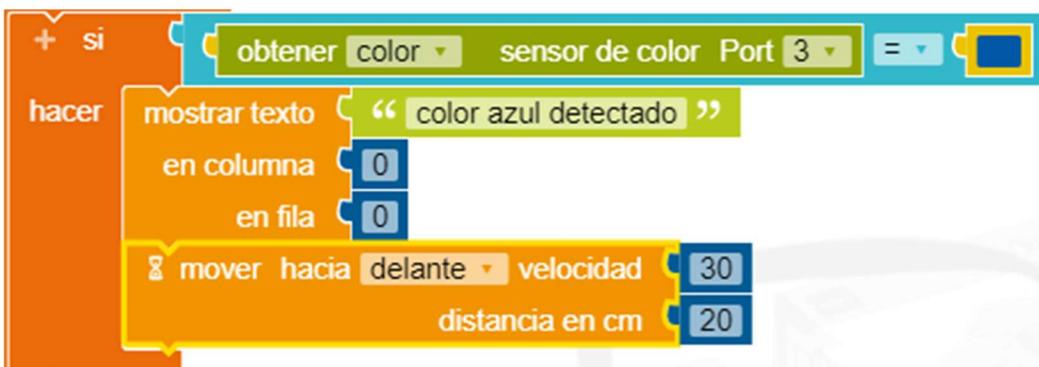


Donde se pregunta si al obtener o detectar un color mediante el puerto 3 y este es “azul” realiza lo siguiente. Es hora de colocar las instrucciones que se realizarán cuando la pregunta sea verdad.

De la categoría “Acción” selecciona la instrucción “mostrar texto” y en el parámetro del texto escribe “color azul detectado”, los parámetros en columna y en fila quedan igual.



De la misma categoría selecciona ahora la instrucción “mover hacia delante”. Escribe en el valor de velocidad “30” y distancia en cm “20”. Esta instrucción se agrega para cruzar el ancho de la banda de color detectado.



Ahora de la misma categoría “Acción” selecciona la instrucción “para”, usada para detener el movimiento del robot.



De la categoría “Control” selecciona la instrucción “esperar” y cambia el valor a “2000”.



Finalmente, de la categoría “Acción” selecciona la instrucción “borrar pantalla”, esto borrará el texto que se escribió anteriormente al detectar el color.



Con esta serie de instrucciones se cumple con lo requerido en la elaboración del programa.

Recuerda: Para salir o romper un bucle se utiliza la instrucción “romper el bucle”, esta se usa generalmente cuando se cumple una condición o se realiza alguna acción determinada.



Es hora de que termines la actividad, toma como base esta parte del programa y complétalo. Al finalizar el programa y que verifiques que funciona correctamente, guárdalo con el nombre “BucleySensoresColores”.

Cómo subir un escenario para un nuevo reto

Para resolver la siguiente actividad, te mostraré primero como subir un nuevo escenario.

Lo primero que debes hacer es generar o descargar un nuevo escenario, para posteriormente cargarlo al simulador de Open Roberta.

Para este ejercicio descarga el escenario del siguiente enlace [Reto_01](#).

También lo puedes descargar de esta sección del curso:

Módulo 3 Uso de Sensores y Construcción de escenarios

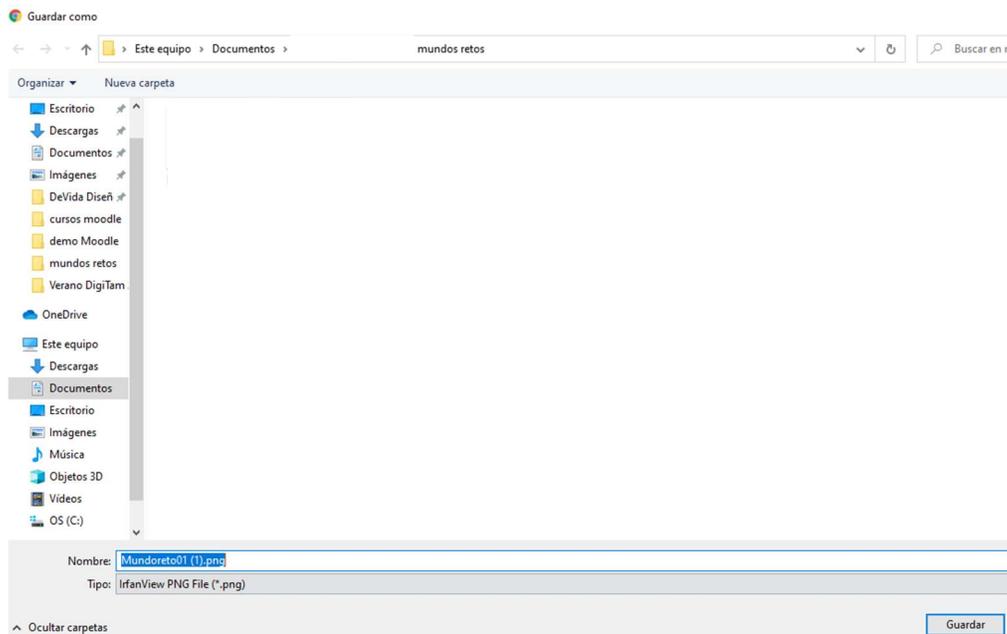
> **Cómo subir un escenario para un nuevo reto** Clic para Mostrar/Ocultar

 Descarga Escenario 01 ←

 **Actividad**

 Actividad: Misión Especial.

Si tu navegador tiene la opción para seleccionar la carpeta donde deseas guardar el archivo, selecciona la carpeta deseada y da clic en el botón guardar.



En caso contrario el archivo descargado se guardará en la carpeta “Descargas”.

★ Acceso rápido

Escritorio

↓ Descargas

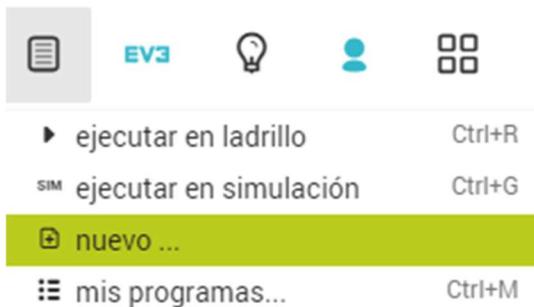
Documentos

Imágenes

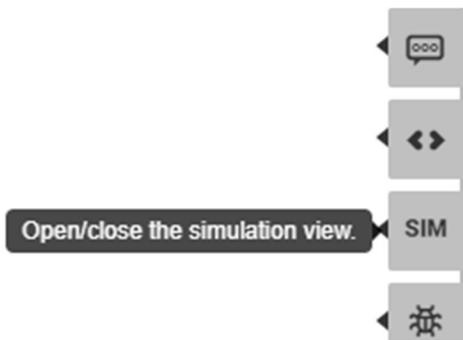
Estas son las dos ubicaciones principales donde podrás ubicar el archivo del escenario para subirlo o cargarlo en Open Roberta.

Para realizar la carga del nuevo escenario realiza lo siguiente:

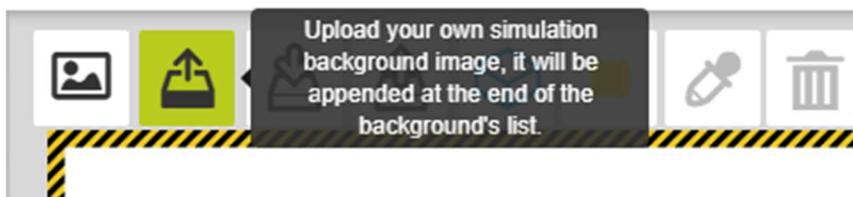
1. abre un archivo nuevo.



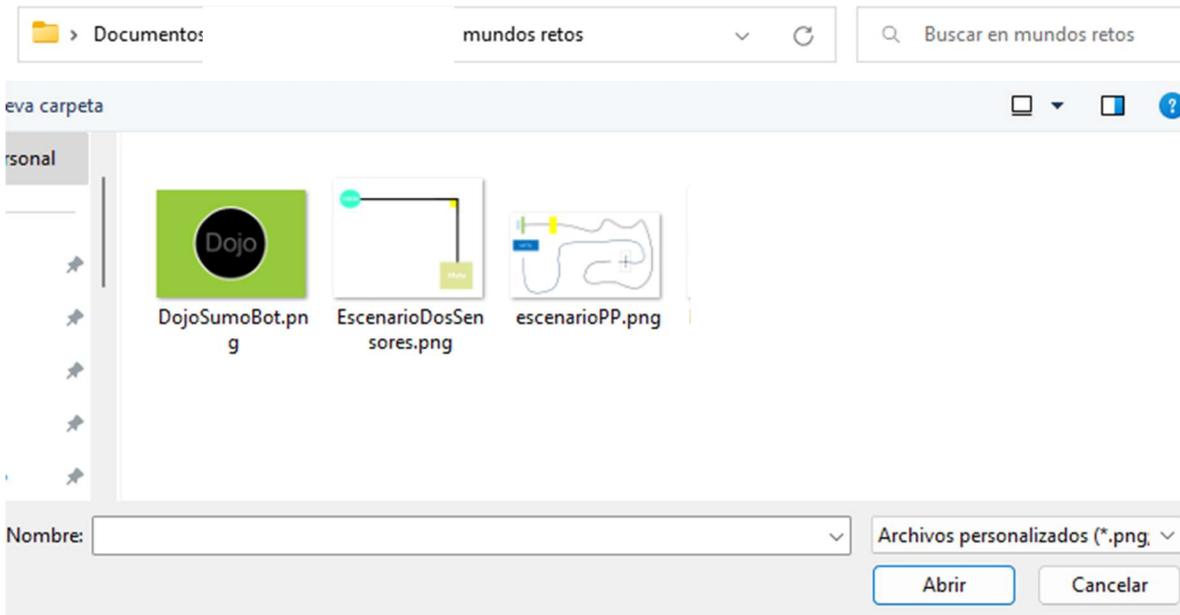
2. Activa el simulador



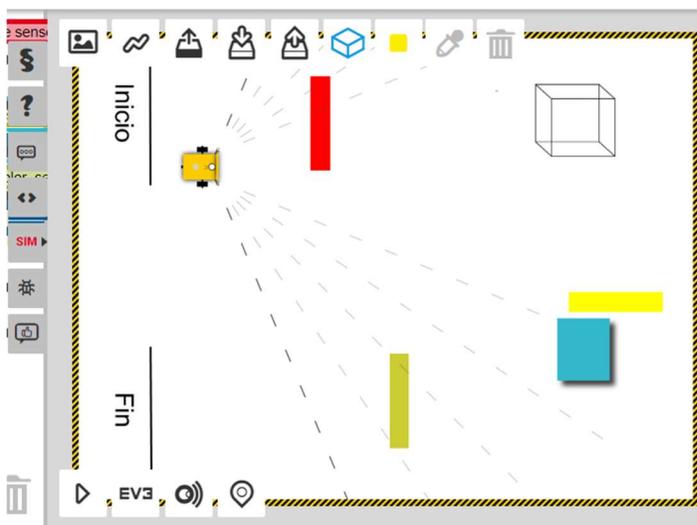
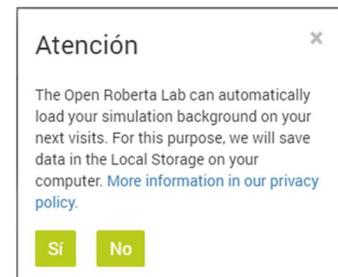
3. Da clic en el botón para cargar el escenario



Te abrirá una ventana para cargar el escenario, ubica la carpeta donde descargaste o guardaste previamente el archivo, selecciónalo y da clic en el botón “Abrir”.



El archivo con el escenario se ha cargado. Te muestra la siguiente ventana (derecha), donde te indica que, si deseas guardar los datos de almacenamiento de tu computadora para posteriormente, cargar automáticamente el escenario que acabas de subir. Da clic en el botón “Si”.



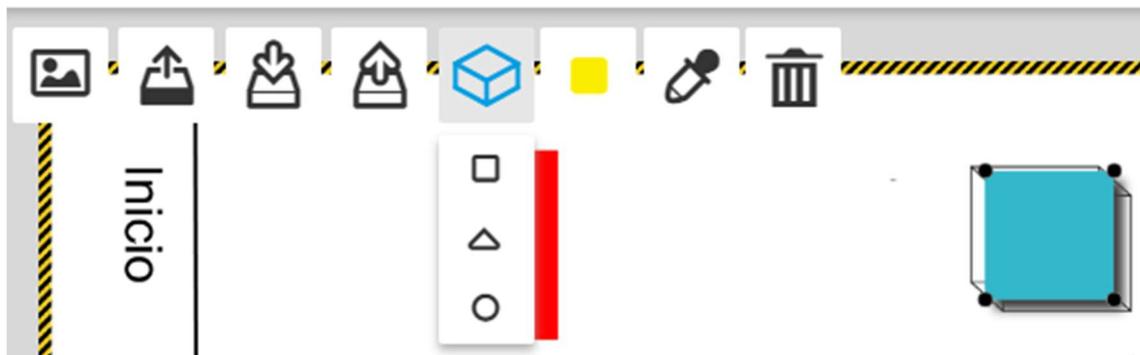
Como podrás observar, se ha cargado el escenario, ahora estás listo para resolver la siguiente actividad.

Actividad: Misión Especial.

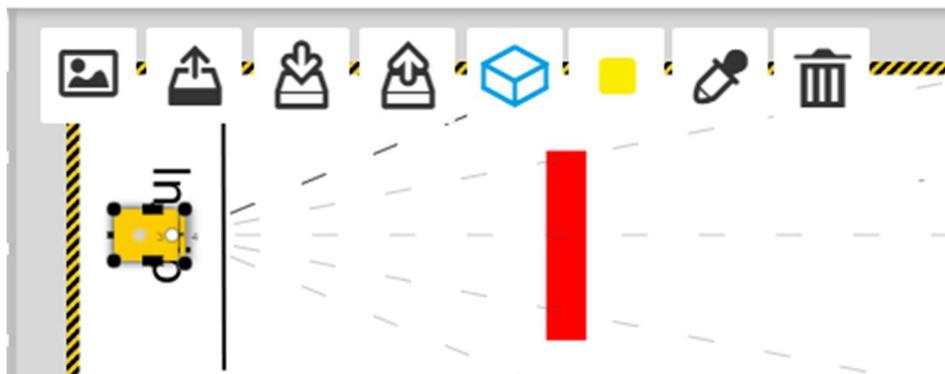
Programa tu robot para completar una misión especial, la cual se desarrolla en un circuito con retos a superar, entre ellos tendrás que buscar patrones de color, detectar obstáculos, esquivarlos para llegar a la meta. En otras palabras, debes utilizar lo aprendido hasta este momento.

Antes de iniciar:

1. Verifica que el escenario que acabas de cargar esté habilitado, de lo contrario selecciónalo para que esté disponible para esta actividad.
2. Para agregar el obstáculo, da clic en el botón para agregar el obstáculo, selecciona la opción del cuadrado, arrastra el cuadro (objeto) que aparece en el escenario hasta el espacio donde se encuentra dibujado en el escenario el cubo. Este objeto lo tendrá que esquivar el robot.



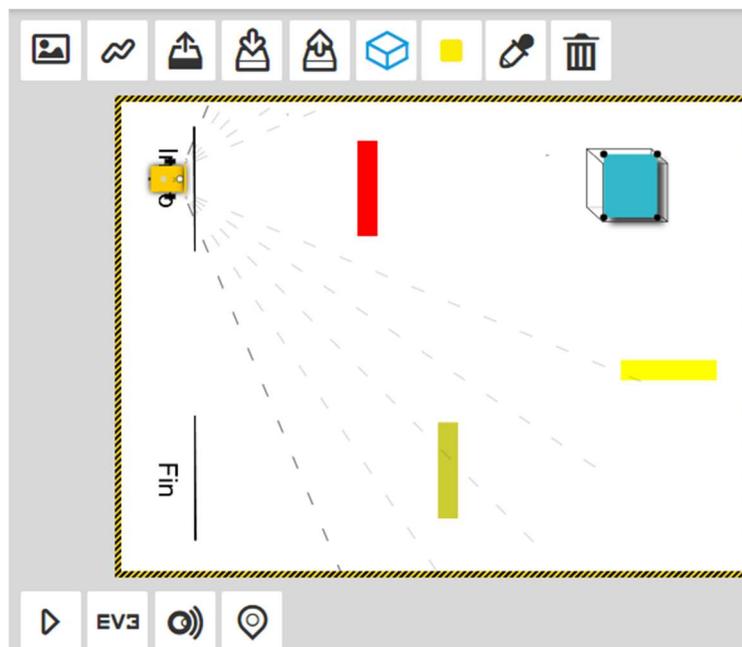
3. Debes colocar a tu robot sobre la marca de INICIO que se encuentra en la parte superior izquierda del escenario. Para ello da clic sobre el robot y muévelo a la posición deseada.



4. Confirma en Configuración del Robot que el sensor 3 = “sensor de color” y el sensor 4 = “sensor de ultrasonidos”.



El escenario que debes tener para resolver esta actividad debe ser como el de la imagen siguiente:



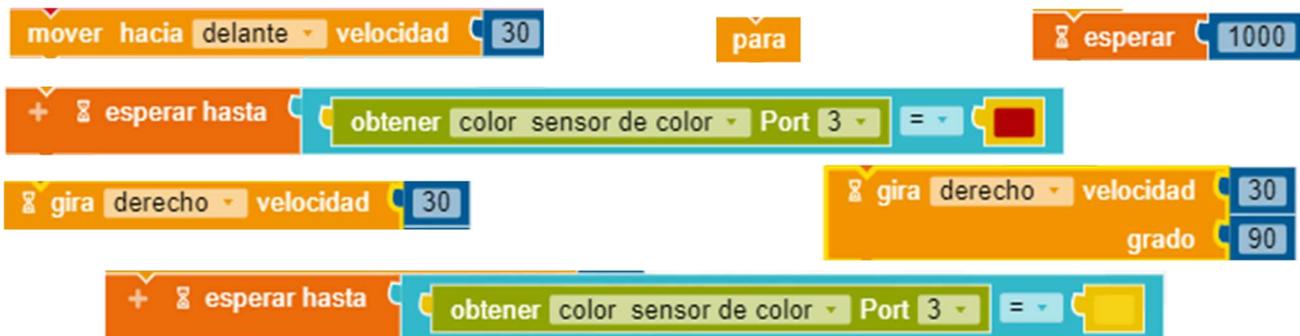
El algoritmo para resolver la actividad puede ser el siguiente:

1. El robot se coloca manualmente en la palabra “Inicio”.
2. El robot deberá avanzar de forma ilimitada, hasta que detecte el color rojo donde deberá detenerse.
3. Espera 1 segundo.

4. Ahora el robot avanza hasta detectar con su sensor ultrasónico un objeto frente a él. Deberá detenerse 10cm antes del objeto.
5. Debe esquivar el objeto, utilizando únicamente los bloques girar y avanzar.
6. Después de haber esquivado el objeto, el robot sigue avanzando y antes de topar la pared del escenario, debe girar 90 grados para colocarse en dirección de la línea amarilla.
7. Ahora el robot avanzará hasta detectar la línea amarilla y deberá frenarse.
8. Esperar 1 segundo.
9. El robot avanza de nuevo y antes de topar con la pared gira 90 grados en dirección hacia la línea azul.
10. Ahora el robot avanza de forma ilimitada y cuando detecta la línea azul, se detiene.
11. Espera 1 segundo.
12. Avanza hasta llegar a la palabra FIN donde terminará su recorrido

Tips

- Recuerda que debes usar instrucciones tales como:



- En este ejercicio no se usan bucles o ciclos.
- Los límites del escenario funcionan como límites u obstáculos.
- Puedes resolver el reto realizando un programa de manera secuencial, es decir, usando instrucciones que permitan resolver una parte del reto.

Te proporciono una parte del programa para que empieces a elaborar el programa que se necesita. Ejecútalo y analiza su funcionamiento. Después de realizar el paso anterior, complementa el programa, no olvides apoyarte en el algoritmo para resolver esta actividad.



Una vez que termines el programa que resuelve el reto, guárdalo con el nombre “MisionEspecial”.

Realiza una captura de pantalla de la pantalla de Open Roberta, donde se pueda ver el programa.

Abre un nuevo documento de Word y pega la imagen de la captura de pantalla.

Guarda el documento de Word con el nombre “MisionEspecial”_tu nombre, por ejemplo “MisionEspecial_JesusAlvarado”.

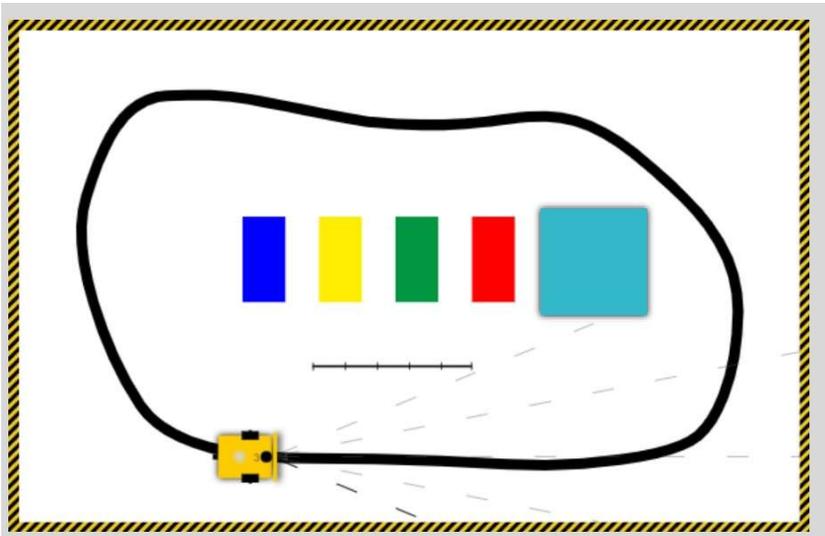
Envía el archivo de Word a través de la plataforma.

Seguidor de línea

Un robot seguidor de línea es aquel que utiliza sensores de color para hacer un recorrido sobre la línea de color negro. Para programar tu robot, puedes utilizar uno o dos sensores de color. En este curso podrás realizarlo con ambas opciones, empezaremos con la primera opción, que es utilizar un sensor de color para programar un seguidor de línea.

Seguidor de línea con 1 sensor de color

A través de una toma de decisiones, el robot va diferenciando entre el color negro (línea) y el color blanco (fondo del escenario), para así hacer un movimiento de zig-zag e ir avanzando.



El seguidor de línea con 1 sensor de color, consiste conectar al robot un sensor de color en el puerto 3, para que cumpla con el algoritmo de seguir la línea negra en el menor tiempo posible.

Para este ejercicio utilizarás los bloques de programación:



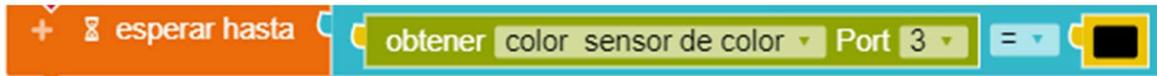
Esta instrucción ya se ha utilizado previamente, cuando lo utilizas indica que todo lo que se coloque dentro de este bucle o ciclo se va a repetir indefinidamente.



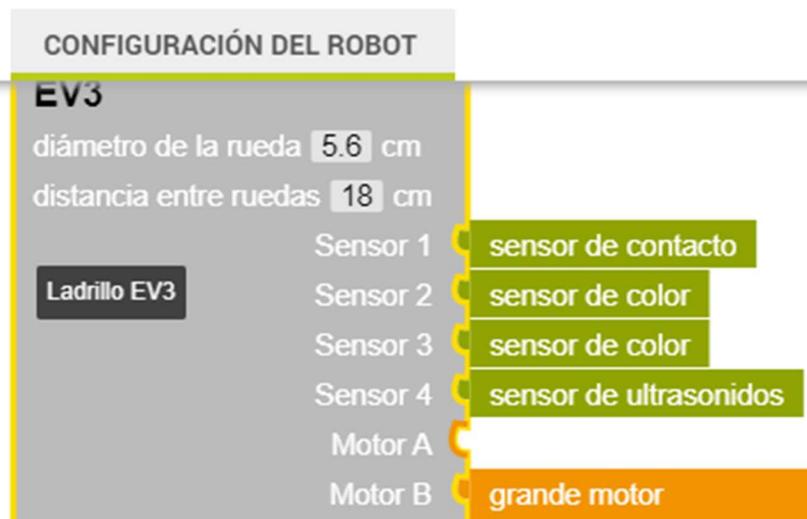
Steer (dirigir) se utiliza para mover los motores de las ruedas a diferente

velocidad, lo que permite que realice un movimiento tipo zig-zag para ir “buscando” o siguiendo la línea negra.

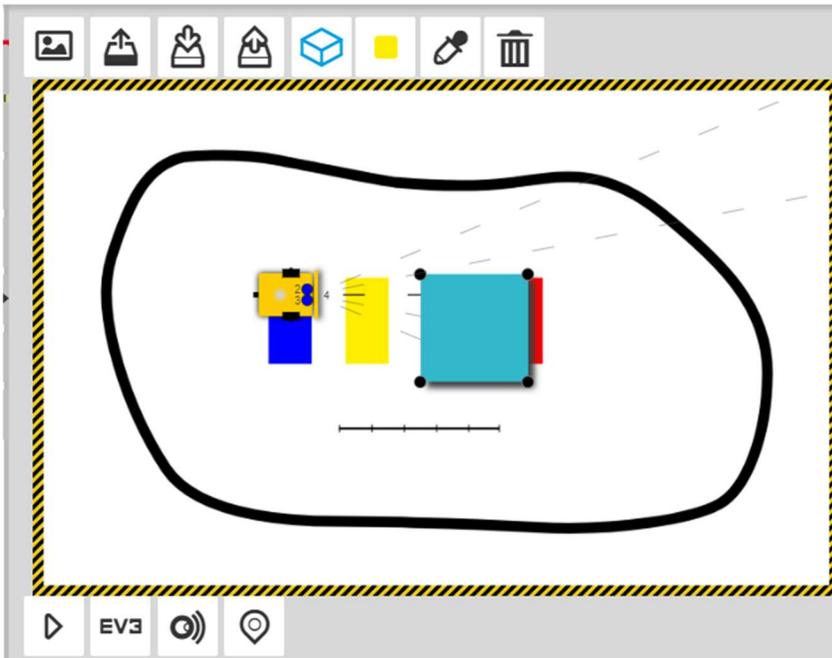
En el caso de la instrucción “esperar hasta” servirá para detectar el color mediante el sensor de color conectado, en este ejemplo, al puerto 3. El color que buscará será el negro.



También debes verificar que en el puerto número tres esté conectado el sensor de color.



Es hora de empezar a programar, antes de iniciar selecciona el escenario que se muestra a en la imagen siguiente. Una vez que finalices el programa o vayas a realizar pruebas, mueve el obstáculo a un lugar donde no interfiera el camino del robot.



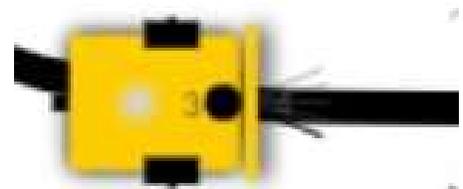
Tomando en cuenta lo anterior, el código del programa puede ser el siguiente:

```

+ Inicio del programa
  mostrar datos de sensor
  repetir indefinidamente
    hacer
      + esperar hasta
        obtener color sensor de color Port 3 =
      steer hacia delante velocidad izquierdo 100
        velocidad derecho 60
      + esperar hasta
        obtener color sensor de color Port 3 =
      steer hacia delante velocidad izquierdo 60
        velocidad derecho 100
  
```

Una vez que termines de copiar el código, guárdalo con el nombre "SensorDeColorUnSensor".

Para probar el código coloca el robot de tal manera que el puerto 3, donde se encuentra conectado el sensor de color, quede sobre la línea negra.



Puedes colocar el robot en cualquier parte de la línea, siempre y cuando el puerto 3 detecte la línea negra.

Ejecuta el programa y observa los resultados. En caso de ser necesario, realiza los ajustes correspondientes para que el programa realice lo que se requiere.

Analizando el código del programa lo que el robot realiza es lo siguiente: al detectar mediante el sensor, el color negro, con la instrucción “steer” hace avance y gire ligeramente a la derecha al tener más potencia en el motor de la rueda izquierda.

Cuando detecta el color blanco, hace que avance y gire un poco a la derecha, ya que la instrucción “steer” le da más potencia al motor de la rueda derecha.

Esto ocasiona que, como ya se mencionó anteriormente, el robot se mueva en zigzag, lo que le permite seguir la línea negra.

Seguidor de línea con 2 sensores de color

El seguidor de línea con 2 sensores utiliza un color para detectar la línea negra y otro para detectar el fondo blanco, haciendo una comparación de estos y así ir avanzando. En este ejemplo, conectaremos un sensor de color en el puerto 1 y otro sensor en el puerto 3.

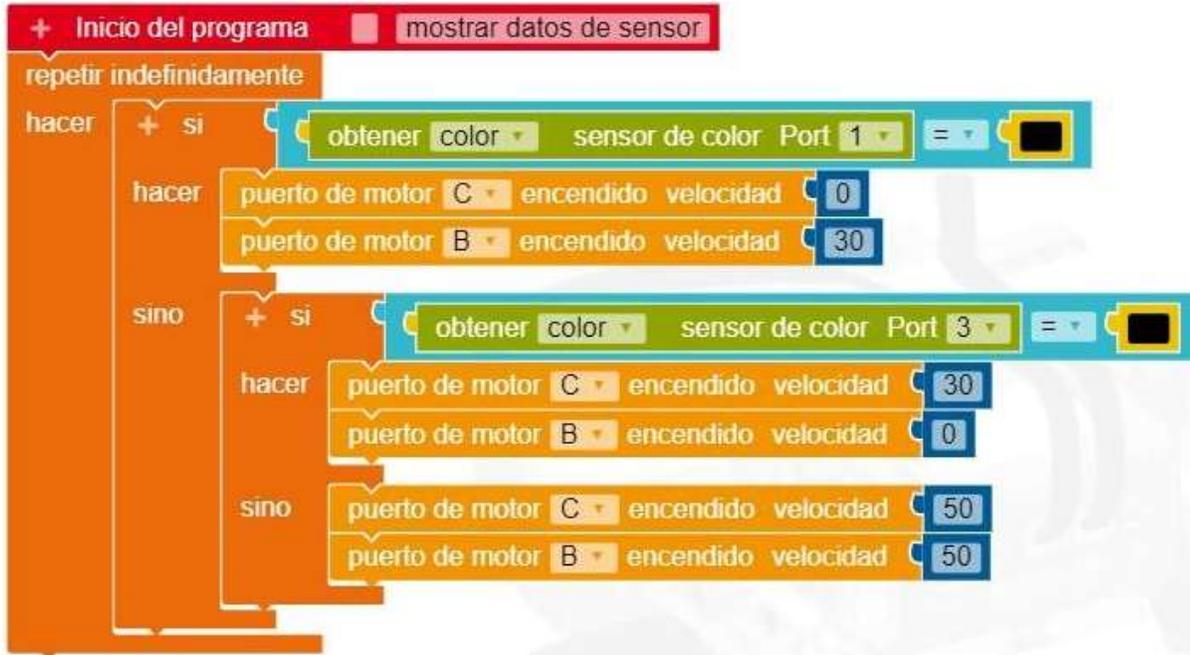
Dependiendo de la potencia utilizada será la velocidad del robot.

Para colocar más se un sensor de color, debes ir a CONFIGURACIÓN DEL ROBOT, verifica que los sensores de color estén conectados en los puertos uno y tres. En caso contrario realiza las modificaciones correspondientes.



Para este ejercicio utilizaremos nuevas instrucciones, que son “puerto de motor” y la condicional “Si – hacer – Sino”. La primera instrucción sirve para hacer mover los motores de las ruedas de manera individual. La segunda es una condicional, que funciona a manera de pregunta, es decir, si se cumple la condición indicada, realiza lo siguiente, si no se cumple, realiza esto otro.

El código propuesto para utilizar dos sensores de color es el siguiente:



Explicación

Inicia el bucle o ciclo. Se pregunta si el sensor de color en el puerto 1 detecta el color negro, detiene el motor izquierdo y hace que el motor derecho se mueva a una velocidad de 30 m/seg.

En caso contrario, se vuelve a preguntar si el sensor de color en el puerto 3 detecta el color negro, mueve el motor izquierdo a una velocidad de 30 m/seg y detén el motor derecho.

En caso contrario continúa moviendo los motores derecho e izquierdo a una velocidad de 50 m/seg.

Se vuelve a repetir el bucle.

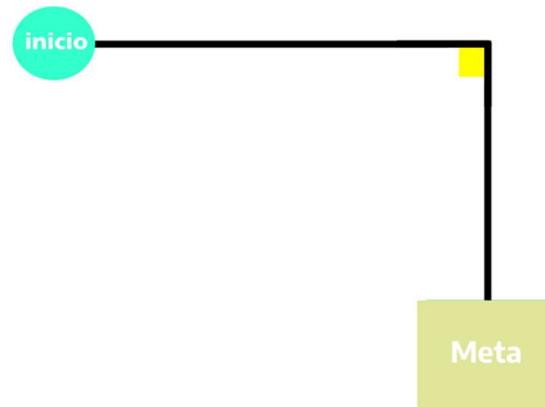
Ahora ve a Open Roberta, crea un nuevo programa, copia el código anterior. Verifica que el programa funcione correctamente. Utiliza el escenario que se usó en el seguidor de línea de un sensor.

Una vez que verifiques que funciona correctamente, guárdalo con el nombre "SeguidorLineaDosSensores".

Uso de dos sensores de color para detectar múltiples colores

Para resolver este reto con múltiples colores, utilizarás un escenario donde es necesario utilizar dos sensores: uno para seguidor de línea (color negro) y otro sensor para detectar color (amarillo) y tomar decisiones.

El escenario lo puedes observar en la imagen de la derecha.



Algoritmo

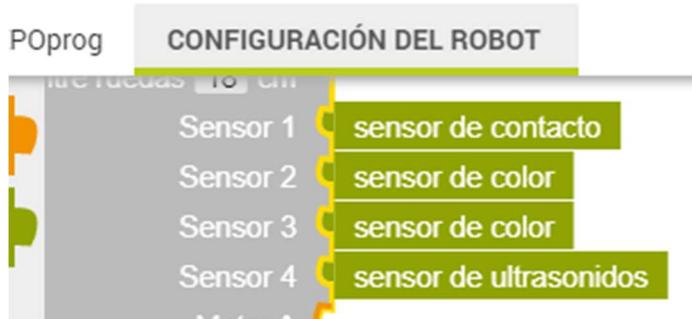
1. El robot debe comenzar con el sensor 2 sobre la línea negra.



2. Seguir la línea negra con el sensor 2, hasta que detecte el color amarillo con el sensor 3.
3. Cuando detecta el color amarillo, el robot se detiene, avanza un poco y después gira hacia la derecha hasta encontrar nuevamente la línea negra con el sensor 2.
4. Espera 1 segundo
5. Ejecuta nuevamente las instrucciones para seguir la línea (negra) con el sensor 2 hasta que detecte el rectángulo del color azul (Meta) con el sensor 3, ahí habrá completado su recorrido.

Antes de iniciar, prepara el escenario para resolver la actividad realizando lo siguiente:

1. Descarga el escenario de la siguiente dirección <https://n9.cl/dossensores> a tu computadora y después cárgalo en Open Roberta con la opción  , tal y como lo hiciste en la actividad anterior.
2. Coloca a tu robot con el sensor 2 sobre la línea negra.
3. Confirma en Configuración del Robot que el sensor 2 y sensor 3 tengan activados el “sensor de color”.



4. En este ejercicio se utilizan las siguientes instrucciones para resolver el ejercicio.



El final del ejercicio se explicarán brevemente el uso de estas instrucciones.

El código que se puede utilizar para resolver esta actividad es el siguiente:

```

+ Inicio del programa
mostrar datos de sensor
repetir indefinidamente
hacer
+ si
  obtener color sensor de color Port 2 = [negro]
  hacer
    steer hacia delante velocidad izquierdo 40
    velocidad derecho 10
+ si
  obtener color sensor de color Port 2 = [blanco]
  hacer
    steer hacia delante velocidad izquierdo 10
    velocidad derecho 40
+ si
  obtener color sensor de color Port 3 = [rojo]
  hacer
    para
      mover hacia delante velocidad 30
      distancia en cm 15
    gira derecho velocidad 20
    + esperar hasta obtener color sensor de color Port 2 = [negro]
    para
      esperar 500
+ si
  obtener color sensor de color Port 3 = [azul]
  hacer
    mover hacia delante velocidad 30
    distancia en cm 20
    para
      gira derecho velocidad 100
      grado 360
      gira derecho velocidad 100
      grado -360
    romper el bucle
  
```

Analizando un poco el código del programa anterior, la explicación sería la siguiente:

El primer bloque que se utiliza es “repetir indefinidamente”, esto quiere decir que las instrucciones que se coloquen dentro del bucle se repetirán indefinidamente.

```

repetir indefinidamente
hacer
+ si
  obtener color sensor de color Port 2 = [negro]
  hacer
    steer hacia delante velocidad izquierdo 40
    velocidad derecho 10
  
```

El siguiente bloque utilizado es “si – hacer”, esto se puede interpretar como una pregunta, si el sensor del puerto dos detecta el color negro, realiza lo siguiente; en caso, de que la respuesta sea positiva el robot se “dirigirá” y avanzará levemente hacia la izquierda, dada la potencia o fuerza que se indica para el motor izquierdo y derecho.



La siguiente condición es para cuando el sensor conectado al puerto dos, detecta el color blanco, cuando la respuesta sea positiva el robot se moverá ligeramente hacia la derecha.



La siguiente condición es para determinar si el sensor de color conectado en el puerto tres detecta el color amarillo. Cuando esto suceda el robot realiza lo siguiente:

1. Se detiene
2. Se mueve hacia adelante 15 cm a una velocidad de 30 m/seg
3. Gira hacia la derecha a una velocidad de 20 m/seg
4. Espera hasta detectar con el sensor colocado en el puerto dos el color negro
5. Se detiene
6. Espera medio segundo para continuar con el ciclo



El siguiente paso o ciclo es determinar si el sensor de color conectado en el puerto 3 reconoce o detecta el color azul, en caso de que se cumpla esta condición, se realiza lo siguiente:

1. Mover hacia adelante 20 cm a una velocidad de 30 m/seg
2. Detener
3. Girar a la derecha 360° a una velocidad de 100 m/seg
4. Girar a la izquierda 360° a una velocidad de 100 m/seg
5. Después de esto, se rompe el bucle o se termina el programa. Ya que al detectar el color azul indica que ha llegado a la meta, para ello se utiliza el bloque “romper el bucle”.



Con base a la explicación anterior, analiza detenidamente el código. Observa que las condiciones pueden no ejecutarse secuencialmente, una después de otra. Al utilizar “si – hacer”, permite determinar que condición se cumple y ejecutarla, esto sucederá hasta que se rompa el bucle.

Ahora, crea un nuevo programa en Open Roberta y copia el código del programa. Una vez que termine, usando el escenario indicado y siguiendo las indicaciones, verifica que funcione adecuadamente el programa.

Posteriormente, guárdalo con el nombre “SensorDeColorVariosColores”.

Sumobot

Un sumobot es un tipo especial de robot que simula los combates japoneses de luchadores de sumo. Consiste en dos robots dentro de un dojo, los cuales hacen uso de sensores, motores y una adecuada programación, para sacar al oponente fuera del dojo y obtener una puntuación.

Para esta lección, utilizaremos el siguiente escenario (dojo) que consiste en un círculo de color negro con una línea blanca en la orilla.



Un algoritmo básico para hacer funcionar a un sumobot, es el siguiente:

1. El robot se prepara para iniciar el combate.
2. El robot debe localizar al oponente, haciendo uso de sus motores ya sea para girar o avanzar. Una vez que el sensor de presencia (puede ser un ultrasónico, infrarrojo, etc.) detecte al oponente, el robot activa sus motores y avanza hasta empujar al oponente.
3. En caso de que el robot avance y se vaya derecho sin encontrar al oponente, el sensor de color detectará la línea blanca, la cual indicará el límite del robot y deberá regresar para no salir del dojo.

Repetir las acciones anteriores hasta sacar al oponente del dojo.

Programando el sumobot

Ahora programaremos un sumobot (luchador de sumo) con un código sencillo, el cual una vez que analices y pruebes, podrás modificar para que sea más efectivo. Los componentes que requiere el robot para programar el sumobot, son los siguientes:

- 1 sensor ultrasonido
- 2 motores
- 1 sensor de color

El programa propuesto para el sumobot es el siguiente:



Las instrucciones ya las has utilizado durante el curso, se utiliza un bucle, giros, mover y esperar hasta.

Una vez que termines de crear el programa, guárdalo, para ello da clic en el menú “editar”, da

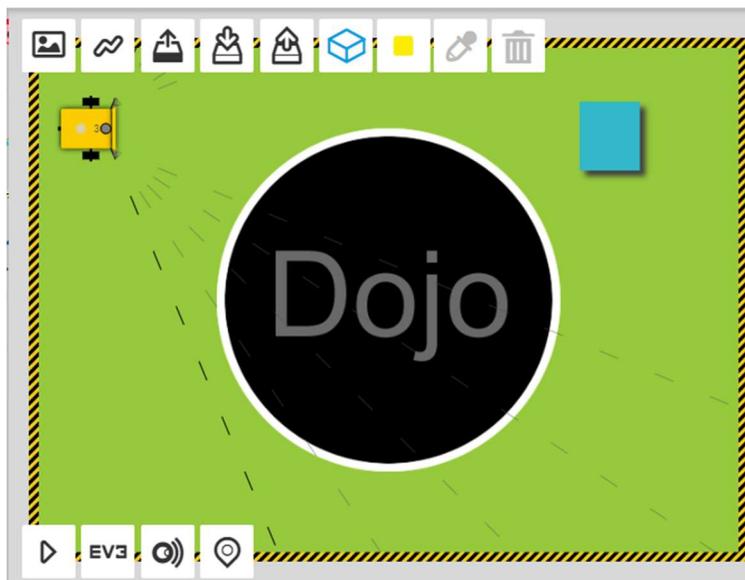


clic en guardar como, el nombre que le asignarás será “sumobot_01”, da clic en el botón “OK”.

Tu programa sumobot ha sido guardado.

Es hora de probar el programa, para ello es necesario cargar el escenario a la plataforma, para ello descárgalo de esta liga: [sumobot, \(https://setcapacitacion.tamaulipas.gob.mx/mod/resource/view.php?id=14064\)](https://setcapacitacion.tamaulipas.gob.mx/mod/resource/view.php?id=14064)

Una vez que los descargues, es necesario cargarlo en el simulador. Después de cargar el escenario activa el simulador y ejecuta el programa. Debes tener una escena como la de la imagen siguiente:



¿Qué pasa con el robot? El robot solo se la pasa girando en el centro del Dojo, ya que de acuerdo con la programación está girando, tratando de “localizar” al oponente, pero no hay oponente.

Para resolver esta situación o colocar un oponente en el mismo escenario realiza lo siguiente:

1. Asegúrate de haber grabado el programa. Ahora graba el programa con otro nombre, esto para mantener una copia del programa que se acaba de crear, para ello da clic en “guardar como”, asígnale el nombre “sumobot_02”. Esto creará un programa similar, con la misma programación, este será el oponente.
2. Para mostrar los dos robots en el Dojo da clic en el menú “editar” y da clic en la opción “simulación con múltiples robots ...”
3. Te debe mostrar una pantalla como la de la imagen de abajo, ubica el programa “sumobot_02”, ahora da clic sobre el botón “+” para indicar la cantidad de robots que necesitas, en este caso serán 2, ahora presiona el botón “OK”.

También puedes usar la opción “Search” para realizar la búsqueda del programa

NOMBRE DEL PROGRAMA	CREADOR	FECHA DE CREACION	
sumobot_02			- 2 +
BaseDetectarCuatroColores	jesusag	15.06.2021, 09:29	- 0 +
demo01	jesusag	17.03.2023, 22:45	- 0 +
SensorDeColorMultiplesColores	jesusag	19.08.2021, 15:11	- 0 +
DetectarLosCuatroColores	jesusag	14.06.2021, 14:54	- 0 +
girar	jesusag	01.06.2021, 11:01	- 0 +
avanzaretrocede	jesusag	01.06.2021, 10:12	- 0 +
textosenpantalla	jesusag	07.06.2021, 09:19	- 0 +

Showing 1 to 10 of 20 rows 10 records per page

Ya tengo los dos oponentes en el Dojo, es hora de ejecutar el programa. Da clic en el botón ejecutar del simulador y observa los resultados.



El código anterior cumple con el algoritmo para que tu sumobot haga la tarea sencilla de localizar al oponente y sacarlo del área de combate, para que tu sumobot sea el mejor dependerá de los algoritmos y modificación a la programación que logres implementar para que tenga un mejor rendimiento en el combate.

Para enviar esta actividad por la plataforma realiza lo siguiente:

1. Activa el escenario de tal manera que se pueda realizar el combate
2. Ejecuta el programa.
3. Cuando estén en combate los robots toma una captura de pantalla.
4. Abre un documento nuevo de Word y pega la captura de pantalla.
5. Guarda el documento de Word con el nombre "Sumobot"_tunombre, por ejemplo "Sumbobot_JesusAlvarado".
6. Cierra el documento de Word.

7. Ahora envía el documento a través de la plataforma del curso “setcapacitación”.

Sugerencia. Lo puedes hacer en otro momento.

Mejora tu luchador

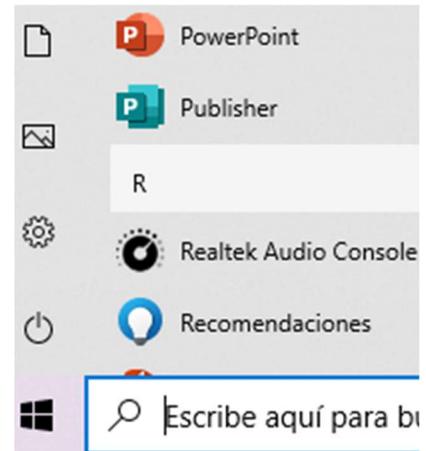
Es tu turno de hacer que elijas tu luchador (un robot), y lo programes de tal manera que mejore lo que pueda hacer, es decir que sea más fuerte que el oponente.

Para ello aplica lo que aprendiste en este curso. Algo de lo que puedes hacer poner más potencia a los motores, que realice un giro, etc., lo que se te ocurra, esto con el fin de que tu luchador sea más fuerte.

Generando escenarios

Open Roberta te permite cargar tus propios escenarios, los cuales pueden ser creados en algún programa de diseño o descargar escenarios de internet e importarlos en la plataforma. Para hacer lo anterior, se necesitan los siguientes pasos.

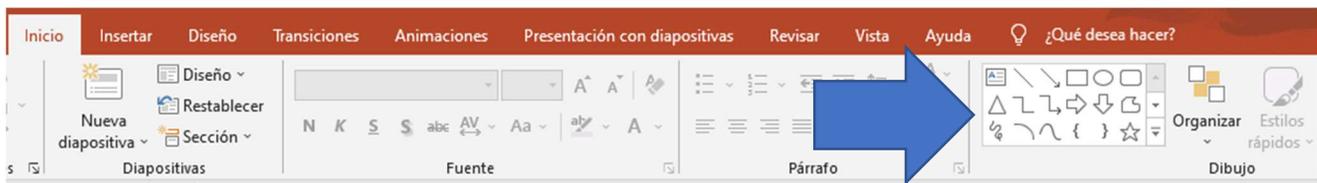
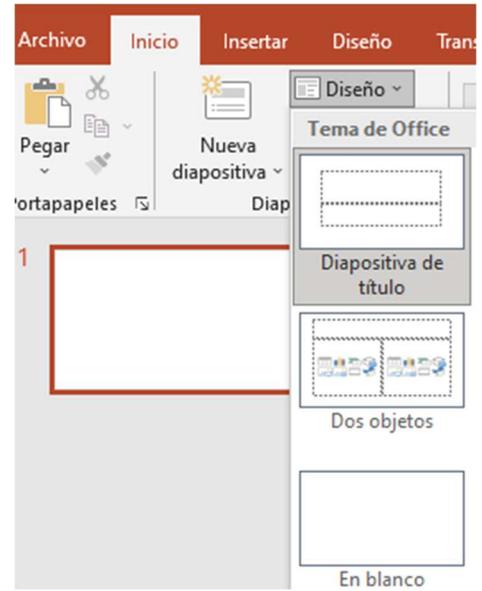
1. Tener el diseño a importar. Primero debemos crear un escenario sobre el cual programaremos nuestro robot. Para ello puedes utilizar una hoja reciclada y dibujar el escenario, este dependerá de lo que quieras que tu robot resuelva.
2. Para crear el escenario, en este ejemplo utilizaré un programa que quizá ya conozcas o hayas usado, PowerPoint (debe de estar instalado en tu equipo). Para empezar, abre PowerPoint, para ello da clic en el botón de inicio y localiza el programa. Una vez que lo tengas habilitado, da clic en él.
3. Una vez que haya abierto el programa da clic en “Presentación en blanco”



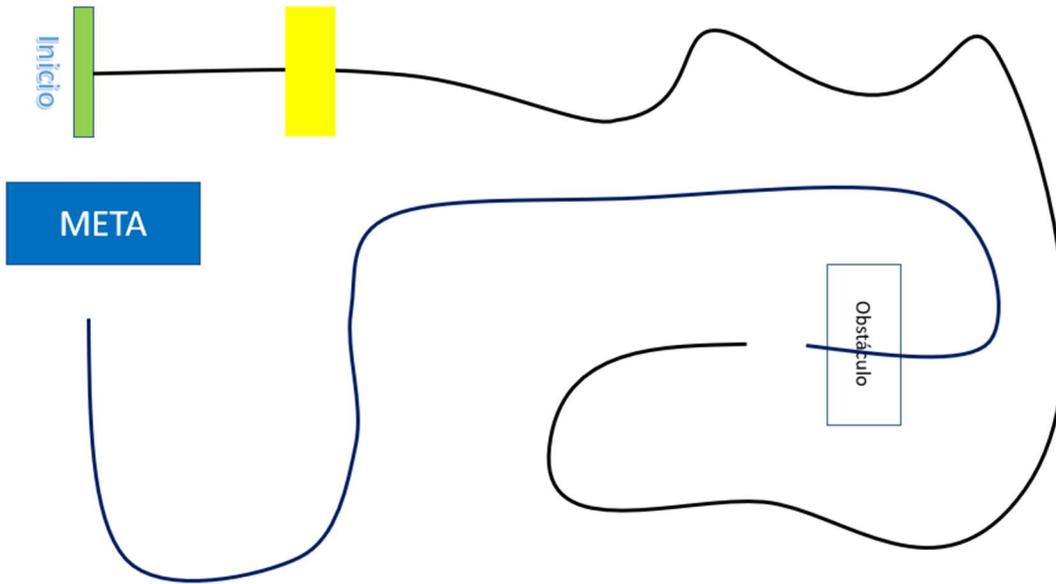


4. Cuando abra el archivo da clic en el comando “Diseño” de la ficha de “Inicio” y selecciona “En blanco”. Esto quitará todas las opciones para agregar elementos predefinidos como cuadros de texto o imágenes.

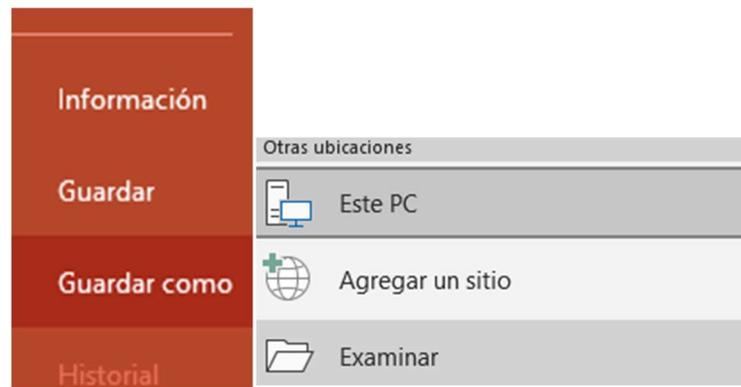
5. Con la ayuda de las herramientas de dibujo puedes empezar a crear tus escenarios. Utiliza tu imaginación y creatividad para crear el escenario que desees.



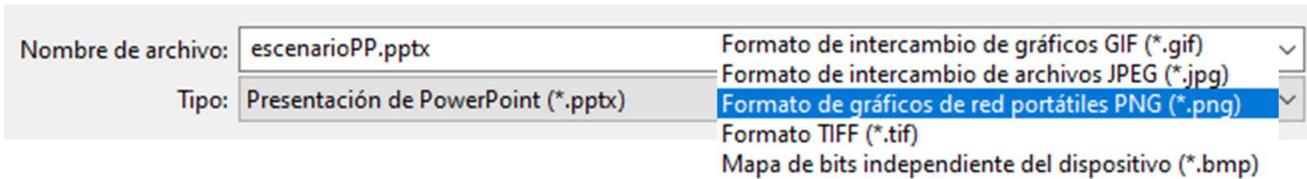
6. A continuación, te muestro el escenario que diseñé con PowerPoint.



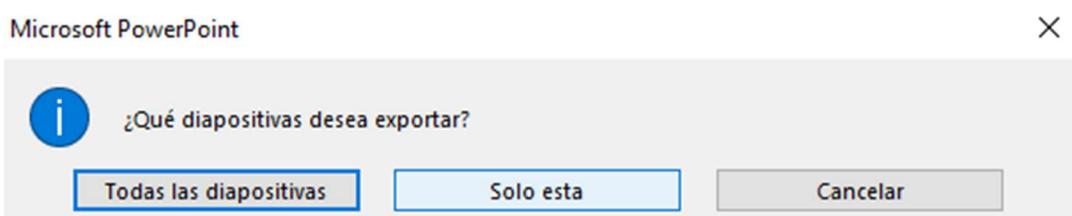
7. Es hora de grabarlo para poderlo cargar en el simulador, da clic en la ficha o menú “Archivo”, selecciona “Guardar como”, después da clic en “Examinar”.



8. Después de seleccionar la ubicación donde guardarás el archivo, escribe el nombre que deseas para tu escenario. Posteriormente da clic en las opciones de “Tipo” y selecciona “Formato de gráficos de red portátiles PNG (*.png)”



9. Da clic en el botón “Guardar”. En el caso de que de pregunte “¿Qué diapositivas desea exportar?” da clic en el botón que dice “Solo esta”.



10. De esta manera has creado tu escenario, solo te falta cargarlo a la plataforma, este procedimiento ya lo realizaste anteriormente, así que manos a la obra.
11. Una vez cargado, es momento de crear el programa que resuelva los retos del escenario.

Recuerda que puedes crear infinidad de escenarios utilizando diferentes programas o aplicaciones, por ejemplo en tu computadora tengas instalados programas como “Paint” o “Paint 3D”. Puedes intentar crear con esos diferentes escenarios.

Es momento de empezar a explorar todas las opciones que te ofrece Open Roberta, intenta crear nuevos programas utilizando todas las opciones de código que esta plataforma te ofrece, poco a poco dominarás esta herramienta, lo que te permitirá crear programas cada más complejos, solo no desanimas y se constante.

Duración del curso

El diplomado tiene una duración de 120 horas, distribuidas en 25 sesiones de 3 horas durante 5 semanas, a través de la Plataforma en línea SET-Capacitación y 45 horas de trabajo independiente para elaborar las actividades de aprendizaje.

3. Modalidad (presencial o mixta) y recomendaciones para su implementación.

El Diplomado está diseñado para impartirse en modalidad en línea, por lo que el participante tendrá total flexibilidad en los aspectos de horario y lugar para trabajar, a través de la plataforma educativa “SET-Capacitación” (<https://setcapacitacion.tamaulipas.gob.mx/>) del Departamento de Capacitación del Centro Estatal de Tecnología Educativa en Tamaulipas.

El trabajo de este asesor, en lo relacionado al acompañamiento al participante, se llevará a cabo exclusivamente por medio de tres canales de comunicación de la plataforma educativa “SET-Capacitación”, como son los mensajes escritos internos a la plataforma, los foros de dudas o novedades y por medio del correo electrónico oficial proporcionado por el asesor.

4. Criterios de evaluación formativa y requisitos de acreditación.

Módulo 1

Presentación.

Actividad 2 ¿Qué es un Robot?	4 puntos.
Actividad 3. Antecedentes	8 puntos.
Actividad 4. Ensayo “Impacto de la Robótica en Educación”	10 puntos.
Actividad 5. Mapa Mental.	8 puntos.
Actividad 6. Descarga, Instalación y Exploración de RoboMind.	10 puntos.
Actividad 7. Trabajando en RoboMind Academy	10 puntos.
Actividad 8. Resolver reto lección seis	10 puntos.
Actividad 9. Resolver las lecciones ocho, nueve y once.	10 puntos.
Actividad 10. Resuelve las lecciones cuatro, cinco y seis.	15 puntos.
Actividad 11. Resuelve las lecciones siete y ocho	15 puntos.
Total	100 puntos

Módulo 2

Actividad 00	3 puntos.
Actividad 01. Elaboración de un algoritmo cotidiano	6 puntos.

Actividad 02. Resolver reto tres	6 puntos.
Actividad 03. Resolver retos cinco, seis y siete	8 puntos.
Actividad 04. Resolver reto cinco “Get the bacon back”	6 puntos.
Actividad 05. Reto seis “Get the bacon back (2)”	6 puntos.
Actividad 06. Exploración de CODE	6 puntos.
Actividad 07. Resuelve los puzzles 3 al 15	10 puntos.
Actividad 08. Resuelve los ejercicios 4 y 5	8 puntos.
Actividad 09. Resolver los ejercicios 7, 8, 9 y 10	10 puntos.
Actividad 10. Resuelve los ejercicios 3, 4, 5 y 6	10 puntos.
Actividad 11. Ejercicio 10-14	6 puntos.
Actividad 12. Crea tu propio juego	15 puntos.
Total	100 puntos

Módulo 3

Actividad Inicial	2 puntos.
Bloques de programación	3 puntos.
Actividad Avanzar – Retroceder	10 puntos.
Actividad Mostrar Texto – Girar	10 puntos.
Actividad Bucles	10 puntos.
Detectar los cuatro colores usando bucle y condiciones	15 puntos.
Actividad: Misión Especial	20 puntos.
Actividad. Sumobot	15 puntos.
Actividad. Creando mi escenario	7 puntos.
Actividad Final	8 puntos.
Total	100 puntos

Una vez concluidas la totalidad de las actividades en tiempo y forma, se promediarán los tres módulos, donde el participante deberá obtener un puntaje mínimo de 80 puntos para acreditar el curso y ser acreedor a la constancia del diplomado, otorgada por el Centro Estatal de Tecnología Educativa, así mismo, una vez que el participante concluya todas las actividades, se le pide responder un cuestionario de satisfacción con la finalidad de evaluar los contenidos del diplomado, la atención del asesor y coordinador respectivamente, en el sentido de la atención y acompañamiento ofrecido durante el desarrollo del mismo, esto a través de las herramientas de comunicación que ofrece la plataforma educativa “SET- Capacitación”.

5. Referencias

- Angulo, C. (15 de diciembre de 2016).** *Educaweb*. Recuperado el diciembre de 2017, de Usos y beneficios de la robótica en las aulas: <http://www.educaweb.com/noticia/2016/12/15/usos-beneficios-robotica-aulas-10717/>
- Barrios, D. (s/f).** *La Robótica y su relación con la Computación*. Recuperado el 2017, de Colegio de Matemáticos del Perú: http://www.comaperu.org/eventos/publicaciones/la_robotica_y_su_relacion_con_la_computacion.pdf
- BeJob. (14 de marzo de 2017).** Recuperado el diciembre de 2017, de 10 aplicaciones para enseñar programación robótica en colegios: <https://www.bejob.com/10-aplicaciones-para-ensenar-programacion-robotica-en-colegios/>
- Code.org.** (noviembre de 2018). *Code.Org* Obtenido de <https://code.org/>
- Figueroa Segura, M., García Cotero, M., Quiroga Plaza, M., Reaño Gutiérrez, M., & Sainz González, M. (mayo de 2011).
- Cualli,** (2017) *Software de simulación en la enseñanza*: <http://www.cualli.org/2011/05/software-de-simulacion-en-la-ensenanza/>
- Gabinete de Tele-Educación.** (1 de febrero de 2017). *Arduino en la programación y robótica educativa*. Recuperado el diciembre de 2017, de <http://blogs.upm.es/observatoriogate/2017/02/01/arduino-en-la-programacion-y-robotica-educativa/>
- González, V.** (2003). *Fundamentos de Robótica*. Recuperado el noviembre de 2017, de http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/ft/index.htm
- Jiménez, F.** (s/f). *¿Qué es La Educación STEM?* Recuperado el 2017, de <http://makermex.com/blog/manufactura-digital-1/post/que-es-la-educacion-stem-269>
- Joyanes Aguilar, L.** (2008). *Fundamentos de programación*. madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U. Recuperado el 15 de febrero de 2018
- RoboMind.** (2016). *Bienvenido a RoboMind.net una nueva introducción a la programación*. Recuperado el 5 de abril de 2016, de <http://www.robomind.net/es/index.html>
- Universidad de Cauca.** (octubre de 2016). *La Robótica Pedagógica como Herramienta para la Construcción de Aprendizajes Significativos en el Aula*. Recuperado el noviembre de 2017, de Universidad de Cauca: <http://www.unicauca.edu.co/eventos/index.php/educoloquio/2016/paper/viewFile/210/103>
- Universidad de Colorado.** (2002). *PhET Interactive Simulations*. Obtenido de <https://phet.colorado.edu/es/>

Universidad Politécnica de Madrid. (1991). Arduino en la programación y robótica educativa. Recuperado el 2017, de Gabinete de Tele-Educación (GATE): <http://blogs.upm.es/observatoriogate/2017/02/01/arduino-en-la-programacion-y-robotica-educativa/>