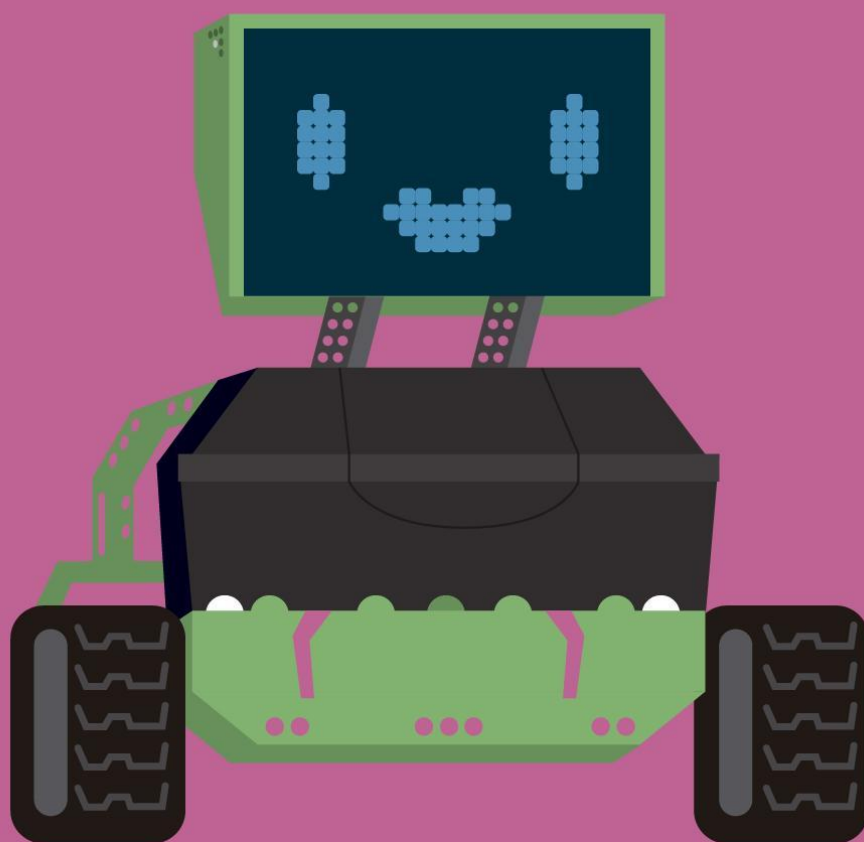


# SABERES DIGITALES

---



ROBOT MÓVIL

---

# AUTORIDADES

## **Presidente de la Nación**

Mauricio Macri

## **Vicepresidenta de la Nación**

Marta Gabriela Michetti

## **Jefe de Gabinete de Ministros**

Marcos Peña

## **Ministro de Educación, Cultura, Ciencia y Tecnología**

Alejandro Finocchiaro

## **Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología**

Manuel Vidal

## **Subsecretario de Coordinación Administrativa**

Javier Mezzamico

## **Director Ejecutivo INET**

Leandro Goroyesky

## **Gerenta General de EDUCAR Sociedad del Estado**

Liliana Casaleggio

## **Directora Nacional de Asuntos Federales**

María José Licio Rinaldi

## **Director Nacional de Educación Técnico - Profesional**

Fabián Prieto

## **Coordinador de Secundaria Técnica**

Alejandro Anchava

## **Responsable de Formación Docente Inicial y Continua INET**

Judit Schneider

## **Coordinador General En FoCo**

Pablo Trangone

AUTORIDADES	¡Error! Marcador no definido.
ROBOT MÓVIL	3
Ficha técnica	3
Presentación	4
Desarrollo	5
<b>Nivel Inicial</b>	5
Paso 1 - Realizar el montaje del robot	5
Paso 2 - Conocer el entorno de programación	6
Paso 3 - Instalar los <i>drivers</i>	7
Paso 4 - Probar los motores	8
Paso 5 - Subir el programa	10
Paso 6 - Bucle infinito	11
Paso 7 - Comenzar el recorrido	11
Paso 8 - Completar el recorrido	13
Paso 9 - Prender los LED RGB	14
Paso 10 - Completar el programa	15
<b>Nivel Intermedio</b>	16
Paso 1 - Conectar el <i>display</i>	16
Paso 2 -Dibujar un mensaje.	17
Paso 3 - Utilizar el botón de la placa	18
Paso 4 - Incorporar el mensaje al recorrido	19
Paso 5 - Continuar el recorrido una vez que se haya leído el mensaje	20
<b>Nivel Avanzado</b>	22
Paso 1 - Optimizar el robot para esquivar objetos	22
Paso 2 - Conectar los sensores	23
Paso 3 - Detectar la distancia	24
Paso 4 - Elegir una dirección	25
Paso 5 - Programar el zumbador	26
Cierre	27
Aplicación móvil y bluetooth	27
Reconocimientos	¡Error! Marcador no definido.

# ROBOT MÓVIL

## Ficha técnica

<b>Nivel educativo</b>	Secundario. Ciclo básico.
<b>Descripción general</b>	Armado y programación de un robot móvil que pueda cumplir instrucciones.
<b>Niveles de complejidad</b>	Nivel inicial: Armar y programar un robot móvil que sea capaz de seguir rutas/trayectorias. Nivel intermedio: Programar un robot que sea capaz de entregar un mensaje en un destino programado. Nivel avanzado: Programar un robot que posea un sensor ultrasónico que le permita detectar la proximidad de objetos para poder esquivarlos.
<b>Equipamiento</b>	<ul style="list-style-type: none"><li>• Computadora</li><li>• Kit Robobloq</li><li>• Pilas Recargables AA (x6)</li></ul>
<b>Otros requisitos</b>	<ul style="list-style-type: none"><li>• Conexión a internet</li><li>• Descargar el programa “Robobloq” <a href="https://www.robobloq.com/support/download">https://www.robobloq.com/support/download</a></li></ul>

# Presentación

## **Descripción ampliada del proyecto**

El proyecto propone la construcción y programación de un robot utilizando el kit de *Robobloq*. Este será capaz de realizar distintas acciones y recorridos. En primera instancia, estará programado para recorrer distintos caminos de forma automática. En el nivel intermedio incorporará un display y un sensor para poder llevar un mensaje de un lugar a otro. Por último, en el nivel avanzado, transitará de forma autónoma utilizando un sensor de distancia ultrasónico.

El kit de Robobloq es un kit educativo de robótica basado en Arduino, que incorpora conexiones simples. Se ensambla mediante piezas de metal de diversos tamaños, formas y colores, permitiendo modificar la forma del robot de acuerdo a lo que se desea que haga. También cuenta con ruedas, orugas, un sensor ultrasónico y un display para realizar y mostrar animaciones.

## **Objetivos**

- Introducirse en algunas nociones de mecánica mediante el armado de un robot.
- Comprender la lógica de los sistemas de movimiento.
- Analizar y desarrollar la programación secuencial de un programa que indique que el robot realice un recorrido determinado (nivel inicial), que lleve un mensaje (nivel intermedio) y que esquive objetos (nivel avanzado).
- Resolver el ensamble de las piezas del robot según las necesidades del diseño.

# Desarrollo

## Nivel Inicial

En una oficina de desarrollo tecnológico, los empleados encontraron una divertida manera de enviarse paquetes: construyeron y programaron un robot móvil que se desplaza desde el depósito hacia el laboratorio siguiendo una secuencia de movimientos que están programados de acuerdo con el espacio que debe recorrer. Como medida de seguridad, programaron los LED del robot para que se enciendan de modo tal que este sea fácilmente visible.

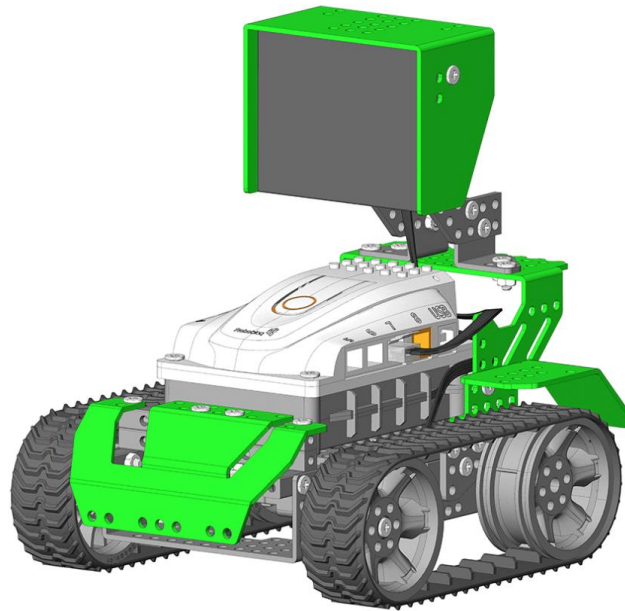
**En esta actividad se propone el diseño y el ensamble de un robot capaz de moverse siguiendo un recorrido previamente pautado, programando una serie de instrucciones.**

### Paso 1 - Realizar el montaje del robot

Realizaremos el ensamble de un robot móvil a partir de las piezas que vienen con el *kit*. El robot se desplazará en el espacio siguiendo una serie de instrucciones que serán programadas previamente.

El kit incluye un manual de ensamble junto con diseños alternativos que pueden obtenerse en el siguiente enlace: <https://static.roboblog.cn/wiki/02.pdf>

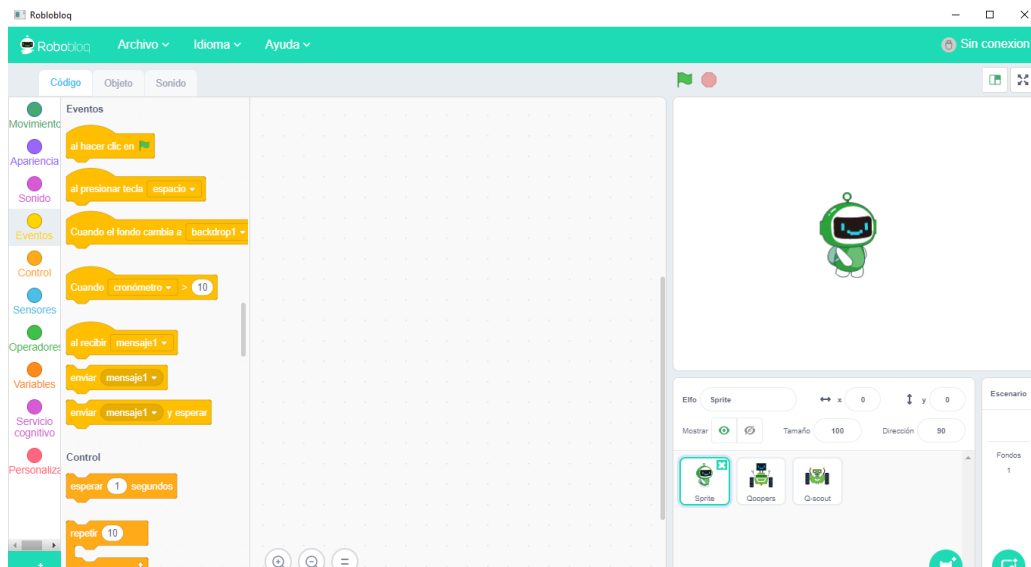
Luego de seguir las instrucciones de armado, el robot debería verse así:



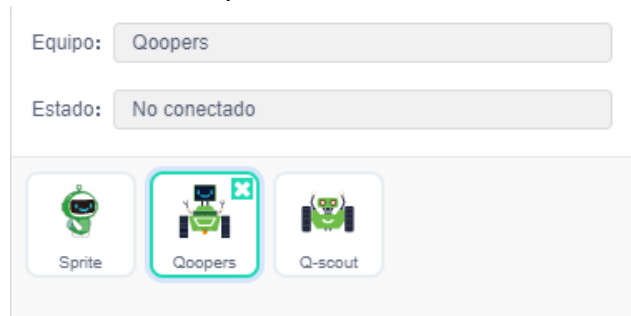
## Paso 2 - Conocer el entorno de programación

Para poder programar el robot móvil utilizaremos el *Software* de Robobloq, que es un entorno de programación que permite programar proyectos de Arduino utilizando bloques. Se puede descargar siguiendo este enlace: <https://www.robobloq.com/support/download>

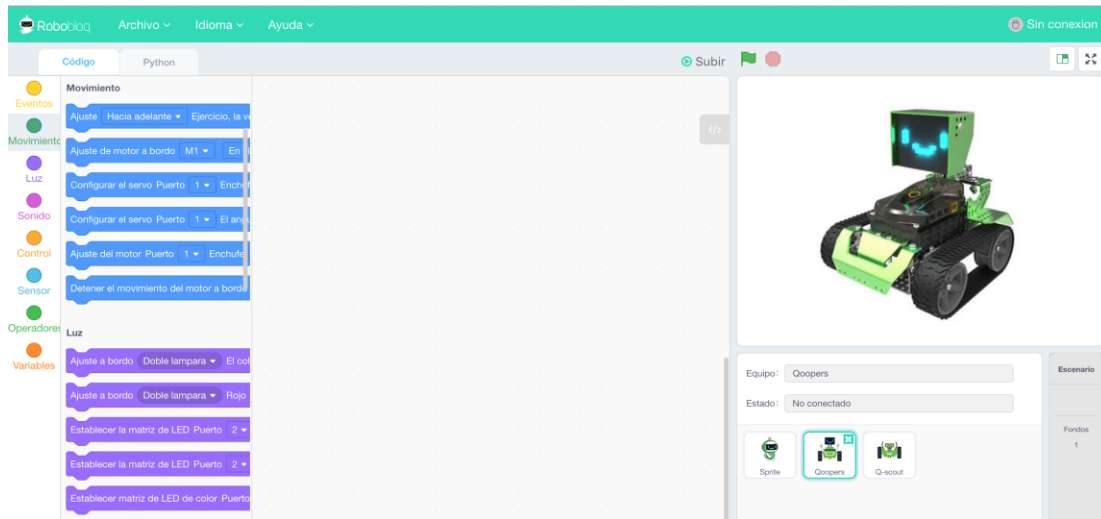
Cuando abrimos Robobloq, vemos una pantalla como la siguiente:



Para programar un proyecto y poder subirlo a la placa controladora con Robobloq, debemos seleccionar nuestro robot, llamado “Qoopers”.

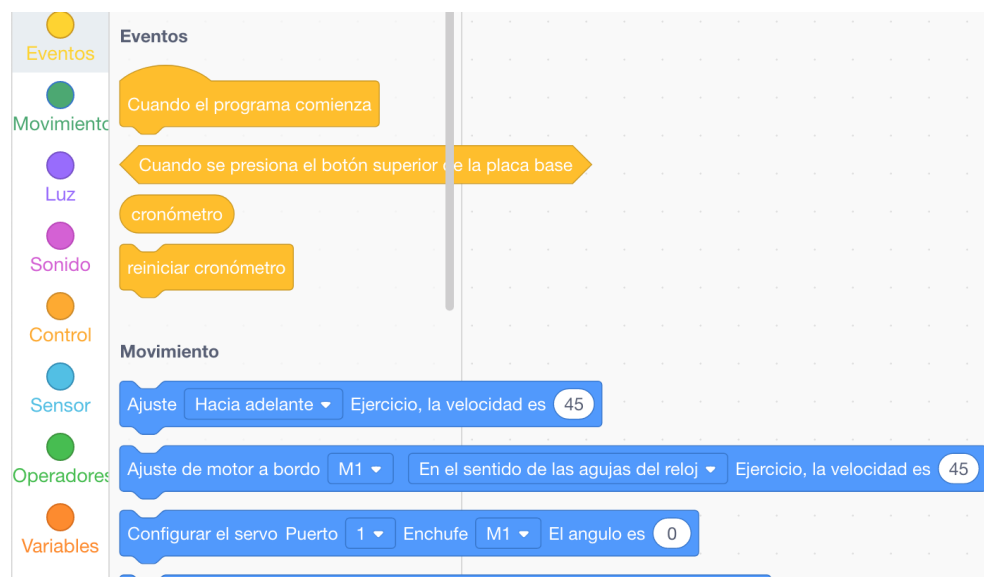


Al seleccionarlo, el programa cambiará de aspecto. Se verá nuestro robot en la pantalla y los bloques disponibles habrán cambiado específicamente para el uso de Qoopers.



A la izquierda de nuestra pantalla encontraremos los bloques agrupados en distintas categorías:

- Eventos
- Movimiento
- Luz
- Sonido
- Control
- Sensor
- Operadores
- Variables



### Paso 3 - Instalar los *drivers*



Antes de comenzar con la programación necesitamos instalar los *drivers* para nuestro robot. Hay que tener en cuenta que esto se realiza únicamente la primera vez que utilizamos el programa. Para instalarlo debemos seleccionar la opción “Instalar el controlador del puerto serie” que se encuentra dentro de la pestaña “Ayuda”.

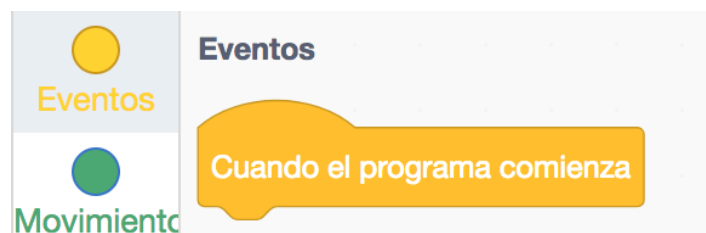
Un *driver* es un programa que sirve para controlar un dispositivo.



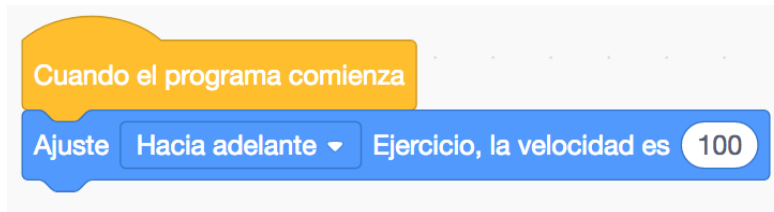
#### Paso 4 - Probar los motores

Lo primero que tenemos que hacer es familiarizarnos con los movimientos del robot. Para esto realizaremos un pequeño programa para que el robot se mueva hacia adelante, frene y se mueva hacia atrás.

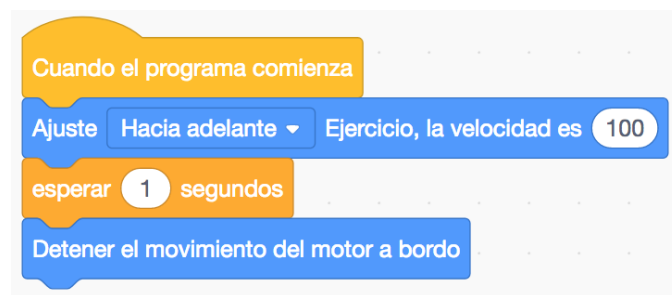
Tenemos que colocar es el bloque que da inicio al programa (“Cuando el programa comienza”). Este bloque lo encontramos dentro de la categoría “Eventos”.



Al comenzar el programa, lo primero que hacemos es pedirle al robot que se mueva hacia adelante. Es necesario establecer, además de la dirección del movimiento, la velocidad con la que se trasladará el robot (en una escala entre 0 y 100). En este caso, pondremos la velocidad máxima (100).



Debemos tener en cuenta que el robot no frenará a menos que le indiquemos que lo haga. Entonces, haremos que el robot avance durante un segundo, utilizando un bloque de espera y eligiendo el valor “1” (segundo), y luego agregaremos el bloque “Detener el movimiento del motor a bordo”.



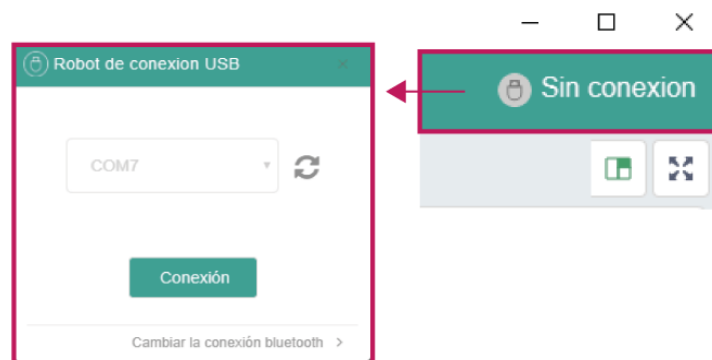
Queremos que, después de frenar, el robot se mueva hacia atrás. Para conseguirlo, colocaremos los bloques correspondientes del modo que se ve en la imagen:



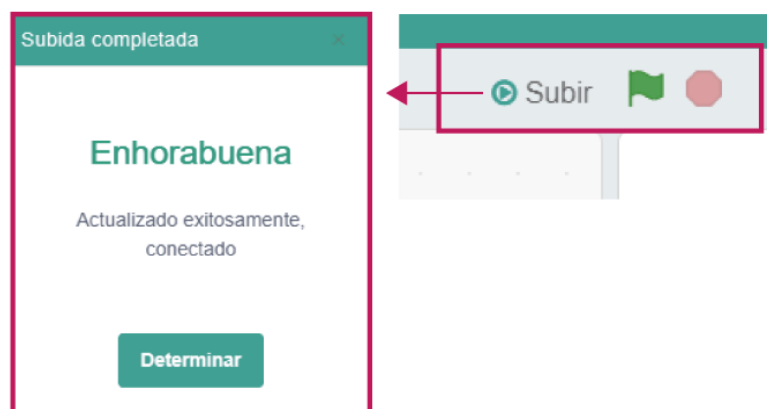
## Paso 5 - Subir el programa

Para subir el programa a la placa necesitamos:

1. Conectar el robot con nuestro cable USB.
2. Hacer *click* en el botón “Sin conexión” que se encuentra en la parte superior derecha (este nombre luego cambiará por “Conectado” una vez efectuada la conexión usb con el robot).
3. Seleccionar el puerto disponible, en nuestro caso el COM7 (puede ser distinto).

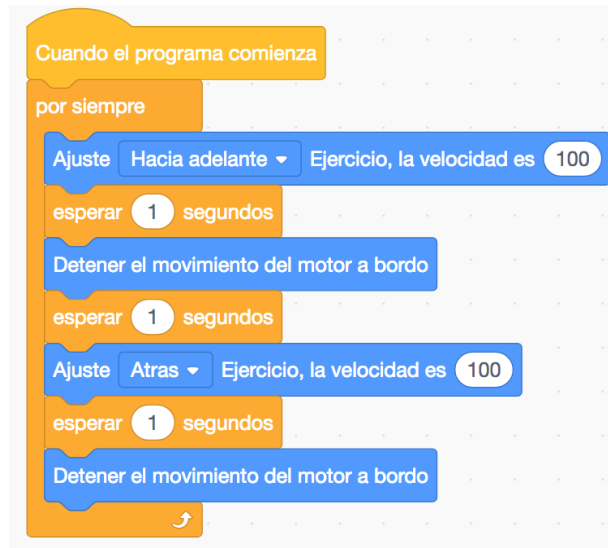


Por último debemos hacer click en “Subir”. Este botón se encuentra en la parte superior, al centro de la pantalla. Al hacer *click* se cargará una barra que mostrará el proceso de carga (puede llevar varios segundos). Una vez cargado nuestro programa a la placa, nos aparecerá una ventana con el cartel “Enhorabuena”.



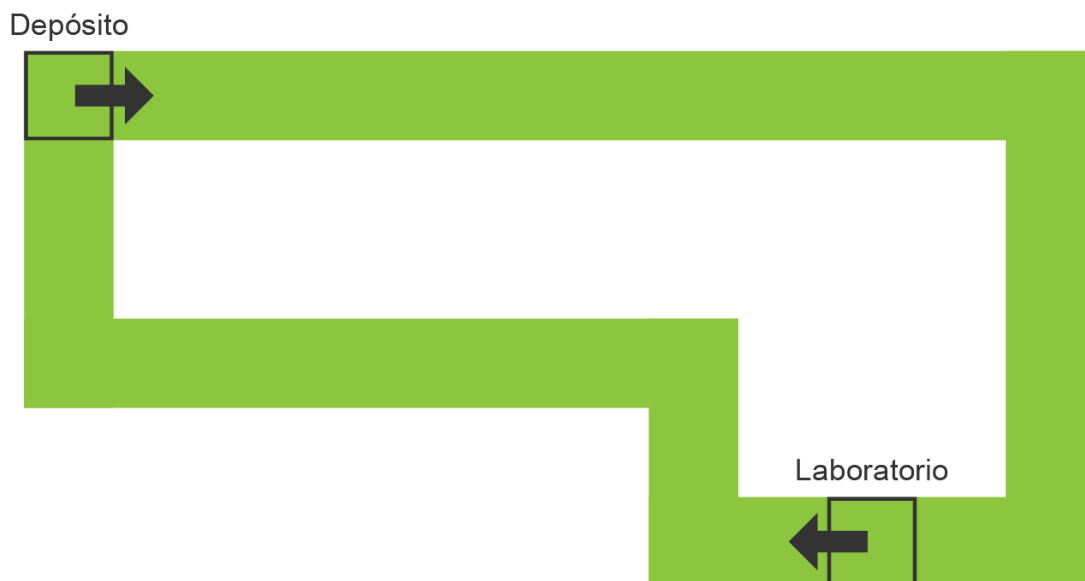
## Paso 6 - Bucle infinito

Con el programa que generamos recién, el robot realizará las acciones una sola vez. Si queremos que el movimiento se repita una y otra vez, podemos finalizar nuestro programa de prueba agregando el bloque “Por siempre”.



## Paso 7 - Comenzar el recorrido

En la situación problemática se había planteado que el robot debía ir desde el depósito al laboratorio. En la siguiente imagen se puede ver un croquis del recorrido que realizará el robot:

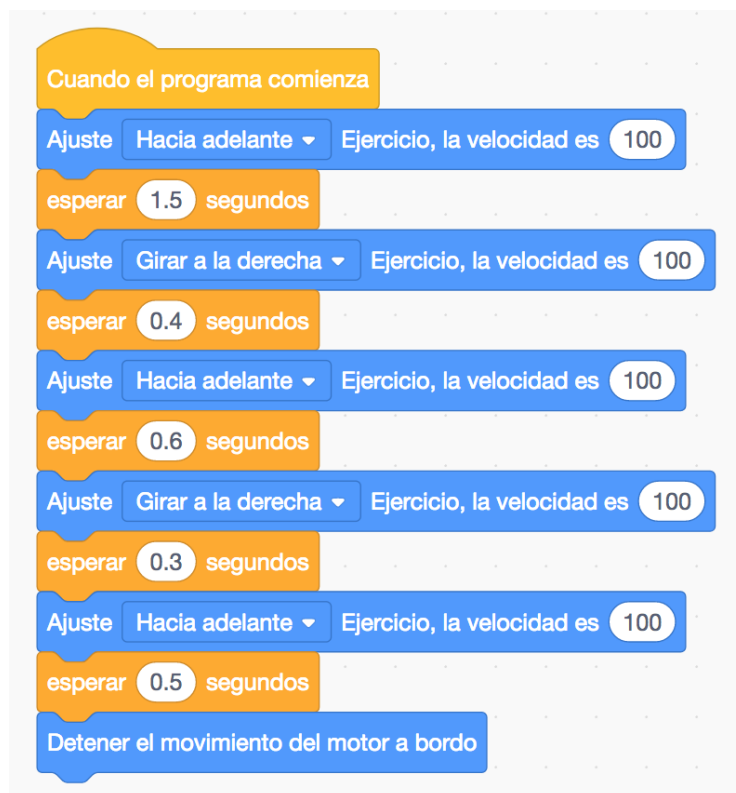


Controlaremos la distancia que recorrerá el robot en cada sentido manipulando la velocidad y el tiempo de duración que les asignemos a cada uno de los bloques.

Vamos a plantear las acciones principales necesarias para que el robot llegue al laboratorio desde el depósito siguiendo el recorrido que se muestra en el croquis:

1. Avanzar
2. Girar a la derecha
3. Avanzar
4. Girar a la derecha
5. Avanzar
6. Frenar

Tenemos que calcular los tiempos de duración que determinaremos para cada acción en función del recorrido y de la velocidad de movimiento del robot (que en este caso mantendremos constante a 100). Para que realice el recorrido del croquis exitosamente, podríamos completar el programa como se ve a continuación:



## Paso 8 - Completar el recorrido

Una vez que nuestro robot llegue a destino, se quedará allí por un tiempo y luego volverá a su origen. Las acciones principales que debe realizar para llegar al depósito son:

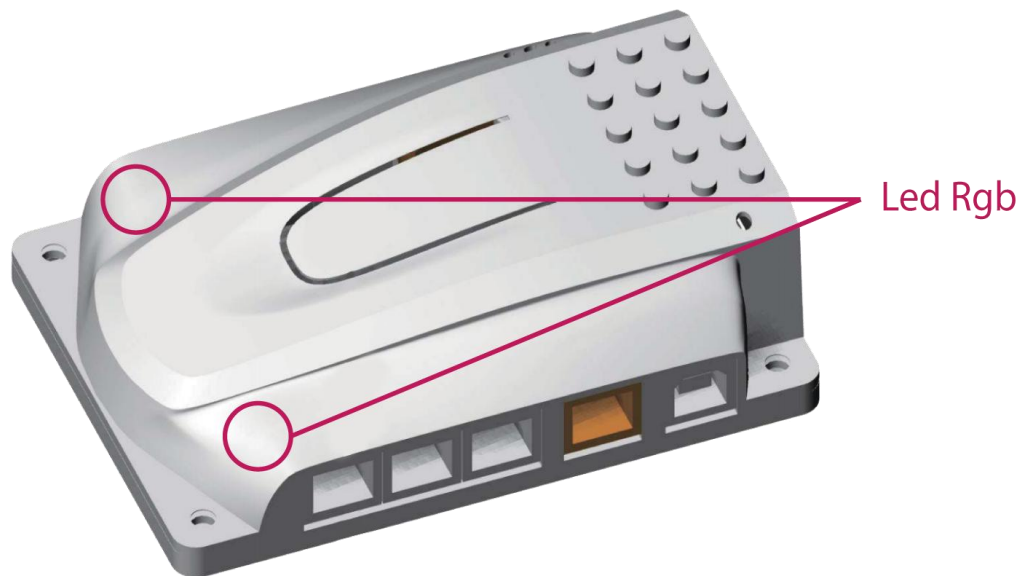
7. Avanzar
8. Girar a la derecha
9. Avanzar
10. Girar a la izquierda
11. Avanzar
12. Girar a la derecha
13. Avanzar
14. Frenar

Nuevamente tenemos que calcular los tiempos de cada acción en función del recorrido a completar y la velocidad de movimiento del robot. Podríamos completar el programa como se ve a continuación:

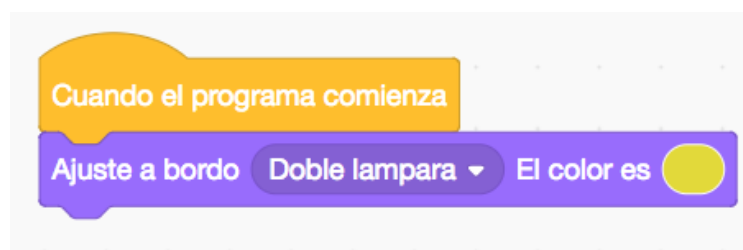


## Paso 9 - Prender los LED RGB

Como se anuncia en el planteo inicial, incorporaremos el encendido de los LED que están integrados al robot. Estos se encuentran al frente de la placa.



Para poder programar estos LED RGB utilizaremos el bloque "Ajuste a bordo" que se encuentra dentro de la categoría "Luz", de la siguiente manera:



El LED RGB es un LED que contiene tres colores (rojo, verde y azul). Mediante la combinación de estos se pueden formar más de 16 millones de tonos de luz.

## Paso 10 - Completar el programa

Para completar nuestro programa, haremos que el robot se detenga y vuelva a repetir todo el recorrido. Nuestro código completo en bloques quedaría de la siguiente manera:



Si presionamos en el ícono *Script* que se encuentra debajo del botón “Subir” podremos ver el código escrito que corresponde a nuestro programa en bloques.





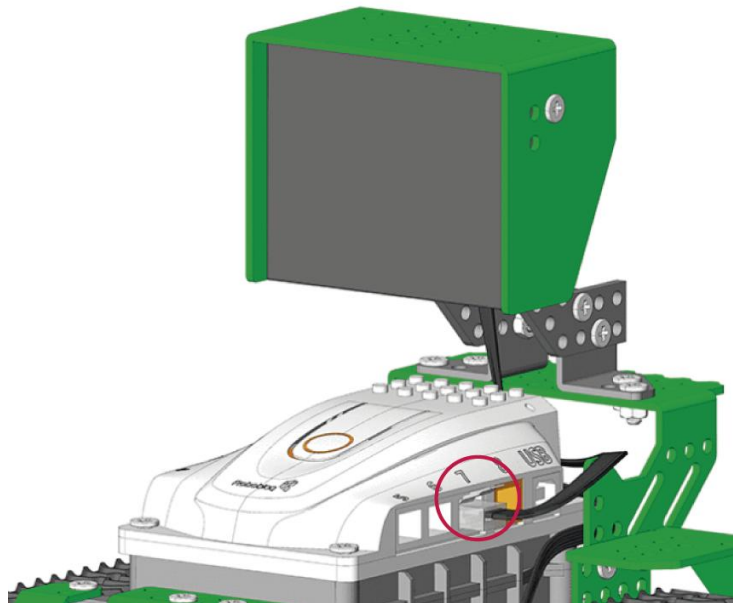
## Nivel Intermedio

Al poco tiempo, los empleados de la oficina quisieron agregarle una nueva funcionalidad al robot móvil: la posibilidad de enviar mensajes desde el depósito al laboratorio. Para conseguirlo, le agregaron al robot una pantalla programable capaz de graficar los mensajes que ellos necesitaban transmitir. Por seguridad, decidieron que estos mensajes sólo serían visibles cuando se pulsara el botón que posee la placa.

**En esta actividad se propone instalarle al robot un display para poder graficar mensajes. Se programará el pulsador que posee la placa para que, al presionarlo, el robot muestre dichos mensajes en el display.**

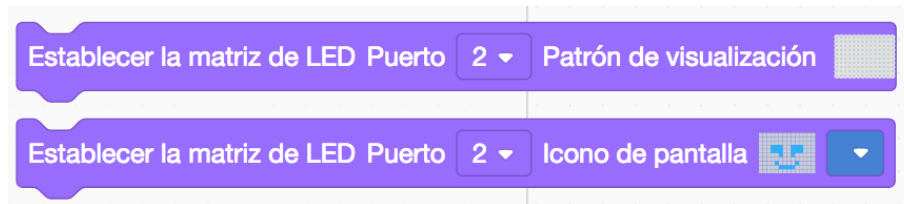
### Paso 1 - Conectar el *display*

Para poder programar mensajes necesitamos utilizar el *display* LED. Lo conectamos con el cable al puerto número 7.



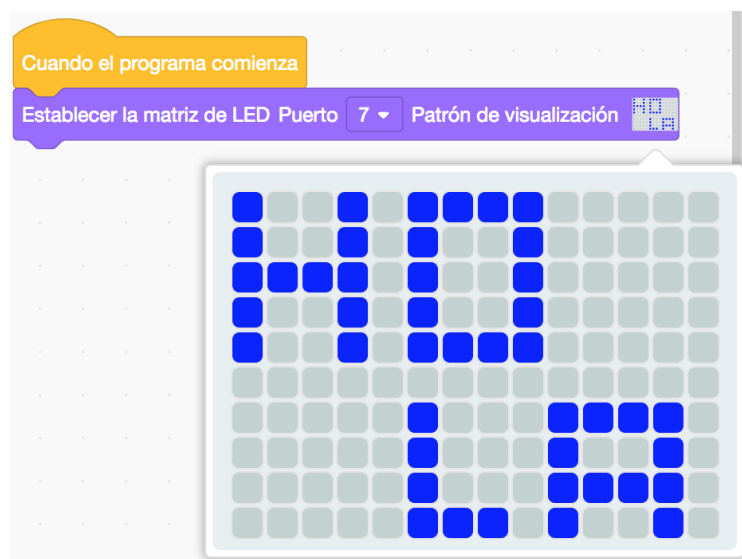
## Paso 2 -Dibujar un mensaje.

Una vez que tengamos el display conectado, podemos programarlo utilizando los siguientes bloques:



El primer bloque nos permite crear nuestro propio dibujo. El segundo nos permite seleccionar entre distintos íconos ya programados.

En este caso utilizaremos el primer bloque (no olvidar seleccionar el Puerto 7) y dibujaremos nuestro propio mensaje, como se ve a continuación:

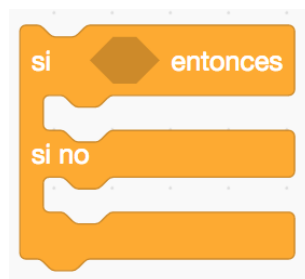


### Paso 3 - Utilizar el botón de la placa

Como mencionamos en el planteo del problema, el mensaje quedará oculto hasta que el robot llegue al laboratorio y alguien presione el botón que se encuentra integrado en la placa.



Para conseguir que el robot se comporte de esta manera, utilizaremos el bloque "Si... entonces/ Si no...". Así, el programa accionará de una forma o de otra en función de una condición (en este caso, si alguien presionó o no el botón). Si el botón está presionado, se mostrará el mensaje; si no, se ocultará. Este bloque se encuentra dentro de la categoría "Control".

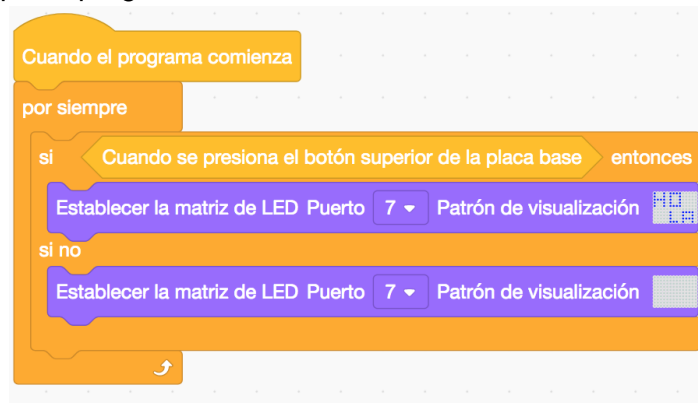


El bloque "Cuando se presiona el botón", que utilizaremos como condición, se encuentra dentro de la categoría "Eventos".

La programación de estas acciones debería verse de forma similar a la siguiente:

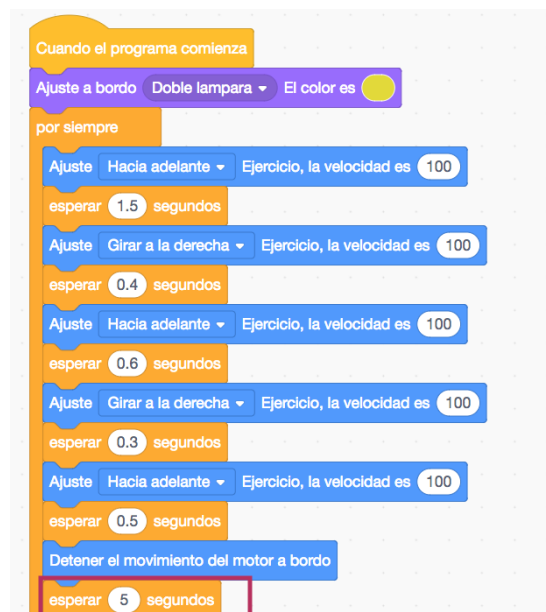


Esta condición que establecimos para que se muestre o se oculte el mensaje deberá repetirse todo el tiempo para que el programa funcione correctamente.



#### Paso 4 - Incorporar el mensaje al recorrido

Una vez que programamos que el robot muestre u oculte el mensaje, debemos incorporar esos bloques al código que habíamos armado en el nivel anterior para programar el movimiento del robot.



Ahora editaremos el programa para que solo habilite el uso del botón para mostrar el mensaje una vez que el robot haya llegado al laboratorio.

Colocaremos el bloque que evalúa si se presiona o no el botón al final del bucle, es decir, luego del bloque “Esperar 5 segundos”. Debemos también agregar un mecanismo para esperar el

pulsado del botón, con un bloque “repetir” y algunas esperas es suficiente. El código nos quedará de la siguiente manera:



## Paso 5 - Continuar el recorrido una vez que se haya leído el mensaje

Una vez que alguien haya pulsado el botón y leído el mensaje, este se borrará y el robot volverá al depósito. Los bloques que indican el recorrido de vuelta que debe seguir el robot deben colocarse debajo de los que condicionan la visualización del mensaje. Es necesario que el robot espere un determinado tiempo desde el momento en que se oprime el botón antes de borrar el mensaje, para permitir que este sea leído (por ejemplo, 5 segundos). El código completo quedaría ahora de la siguiente manera:



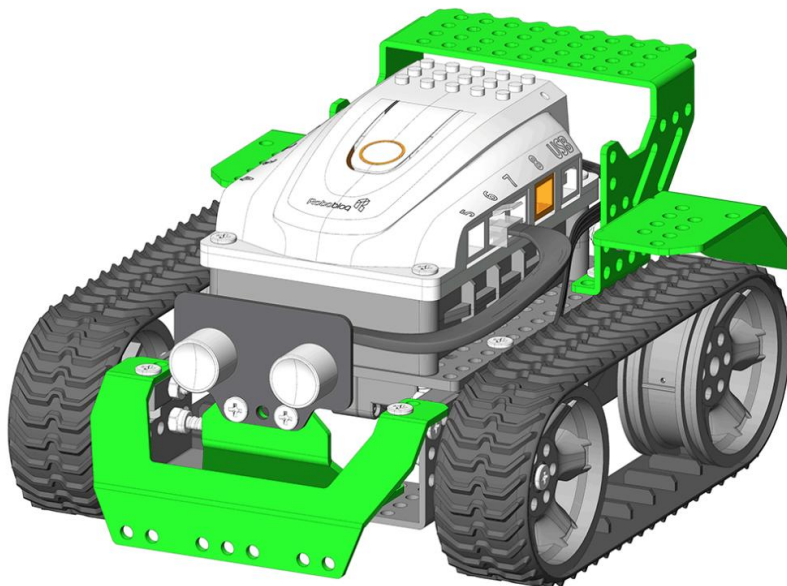
## Nivel Avanzado

Luego de reiteradas fallas en la entrega de los mensajes, se observó que a veces los robots móviles quedaban varados por algún obstáculo que encontraban en su recorrido. Por esta razón, decidieron desarrollar un robot que incluyera sensores ultrasónicos, para que pueda detectar los obstáculos con los que se cruce, y un zumbador en forma de bocina, para que pueda alertar sobre la presencia de obstáculos en su camino.

**Aquí se propone trabajar con el sensor ultrasónico y el zumbador que están incorporados en la placa del robot. Se deberá programar el sensor ultrasónico para que, cuando el robot se encuentre a cierta distancia de un obstáculo, se detenga la marcha, el zumbador suene en forma de bocina y el robot móvil cambie su dirección esquivando el obstáculo.**

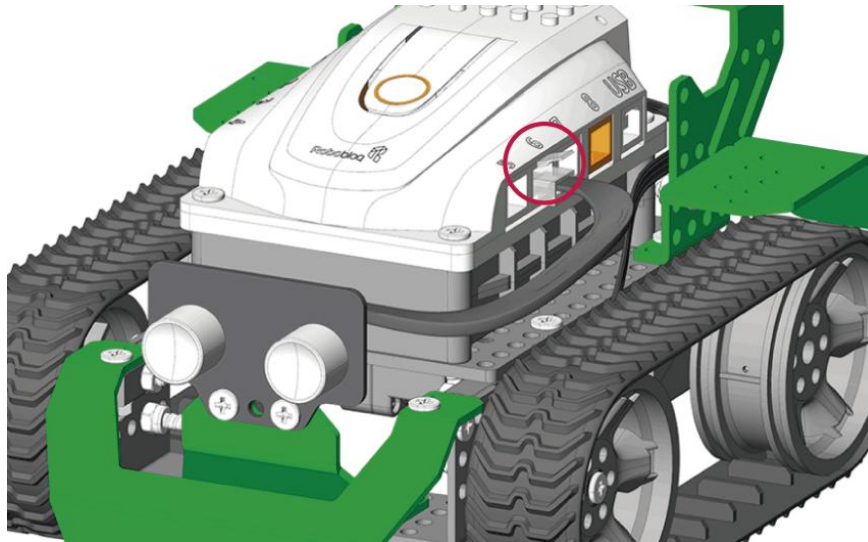
### Paso 1 - Optimizar el robot para esquivar objetos

Para este nivel, construimos otro modelo de robot, como el que se muestra en la imagen. Las instrucciones para construirlo se pueden descargar desde el siguiente enlace: <https://static.roboblog.cn/wiki/03.pdf>



## Paso 2 - Conectar los sensores

Para que nuestro robot pueda esquivar obstáculos, necesitamos conectarle un sensor ultrasónico. En este caso, el sensor estará conectado en el puerto número 6, como se muestra en la imagen.



El sensor ultrasónico es un tipo de sensor que se utiliza para medir distancias. El principio de funcionamiento del mismo se basa en emitir un pulso de sonido ultrasónico y medir el tiempo que pasa hasta registrar que dicho pulso regresó a la fuente tras rebotar en un obstáculo. El tiempo transcurrido es directamente proporcional a la distancia que se encuentra el objeto.

También utilizaremos un zumbador que sonará cuando el sensor ultrasónico detecte un obstáculo. El zumbador está integrado a la placa.



Un **Zumbador** (*buzzer*) es un dispositivo que genera sonido. Algunos lo hacen con una



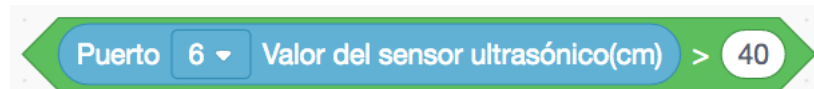
frecuencia determinada y otros permiten controlar la frecuencia (tono). Al recibir alimentación eléctrica comienza a sonar, controlando la misma se pueden generar sonidos continuos o pulsos. Sirve como mecanismo de señalización o aviso.

### Paso 3 - Detectar la distancia

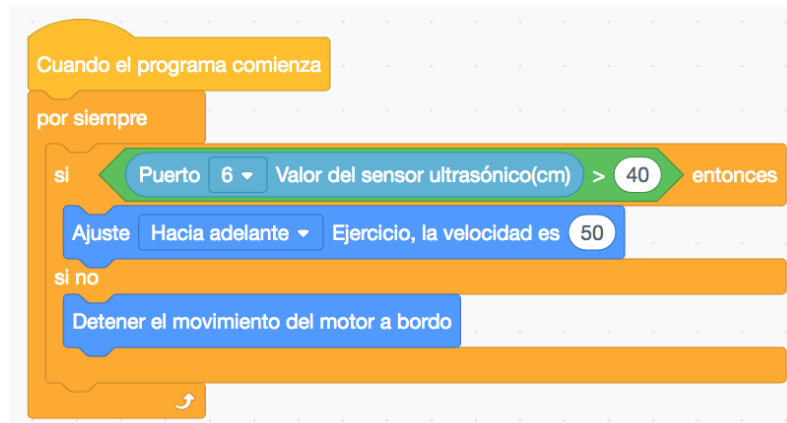
Nuestro robot esquivará los objetos que obstaculicen su trayectoria cuando se encuentren a una distancia previamente determinada por nosotros. El bloque que nos permite determinar esa distancia se encuentra dentro de la categoría “Sensor”:



El robot debe avanzar sólo si la distancia que mide el sensor es mayor, por ejemplo, a 40 cm; en caso contrario deberá frenar. El bloque que nos permite evaluar si la distancia es “mayor que” un valor lo encontraremos dentro de “Operadores”:



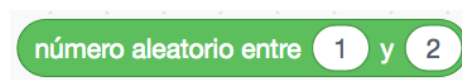
Para poder detectar objetos y esquivarlos, el robot debe ir a una velocidad moderada. En este caso, estableceremos que el valor de la velocidad sea 50. El programa nos quedará entonces de la siguiente manera:



## Paso 4 - Elegir una dirección

Programaremos nuestro robot para que, cuando detecte un objeto en su trayectoria, gire aleatoriamente hacia la izquierda o hacia la derecha. Para esto, utilizaremos el bloque “Número aleatorio entre” que se encuentra dentro de la categoría “Operadores”.

Como indica el nombre del bloque, se elegirá al azar entre “números” y no entre “direcciones”. Entonces, debemos asignarle un valor numérico a cada dirección. En nuestro caso, habrá dos posibilidades: el valor “1” hará que el robot gire a la izquierda y el valor “2”, que gire a la derecha.



A continuación de los bloques que indican girar a la izquierda y girar a la derecha, debemos hacer que el programa espere un tiempo antes de continuar, para permitir que el robot pueda girar (por ejemplo, 0.3 segundos).

El código nos quedaría, entonces, de la siguiente manera:



## Paso 5 - Programar el zumbador

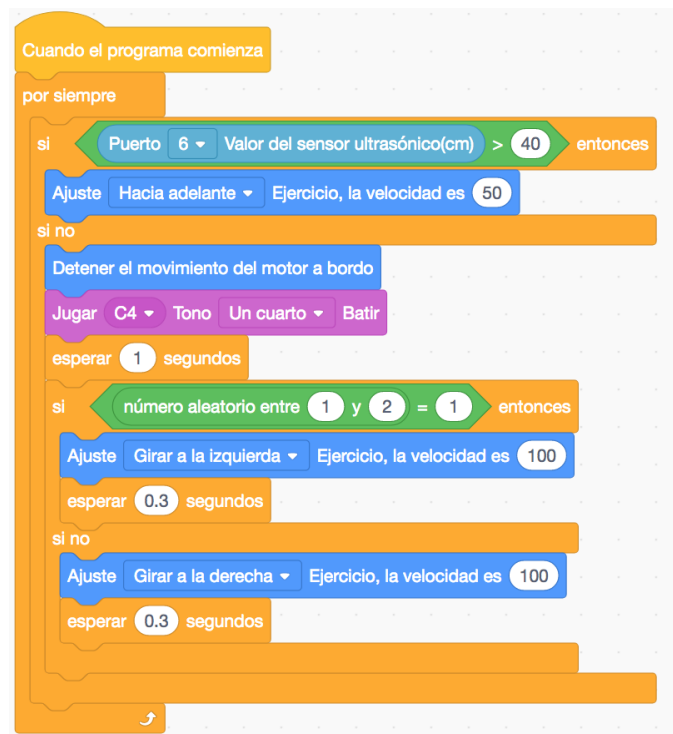
Cuando el robot detecte un objeto emitirá un zumbido que avisará que hay un obstáculo en su recorrido. Para hacer sonar el zumbador, utilizaremos el bloque “Jugar Tono”. En este bloque podremos seleccionar la nota musical MIDI que se reproducirá y el tiempo de duración de su reproducción:



Los tiempos posibles de reproducción del bloque se dividen en fracciones de “tiro” (que es equivalente a un segundo):

- La mitad (0.5 segundos)
- Un cuarto (0.25 segundos)
- Un octavo (0.125 segundos)
- Tiro entero (1 segundo)
- Doble tiro (2 segundos)

El sonido se ejecutará luego de que el robot se frene. El código completo nos quedaría ahora de la siguiente manera:



## Cierre

Una vez finalizado este proyecto, es posible extenderlo si se quiere continuar. Estas son algunas opciones sugeridas:

- Agregarle un sensor seguidor de líneas para automatizar y delimitar un recorrido.
- Programarle melodías a nuestro robot, utilizando el zumbador incluido.
- Programar los LED RGB del ultrasonido para poder diferenciar y personalizar nuestros robots.
- Descargar la aplicación de Robobloq, para controlar y programar el robot mediante el Bluetooth incorporado. Se explica a continuación:

### Aplicación móvil y bluetooth

Ahora necesitamos utilizar el bluetooth con nuestro robot. Para hacerlo, podemos descargar la app de Robobloq en el siguiente enlace:


<http://www.robobloq.com/support/download>



Para poder utilizar la aplicación es necesario sincronizar el bluetooth del dispositivo móvil con el bluetooth de nuestro robot.



La aplicación nos permite controlar el robot con la funcionalidad de *joystick*, realizar animaciones en el display, generar melodías y programar su funcionamiento con bloques.



Si nuestro robot se sincroniza al bluetooth, pero no podemos controlar el robot con la aplicación, debemos restaurar el firmware de fábrica. Esto se realiza desde el software de Robobloq que instalamos en la PC.

Es necesario realizar esto cada vez que subimos un programa nuevo a la placa, ya que se borra el que viene de fábrica que incluye las funciones de bluetooth.

El proceso de resolución de problemas como los que se han planteado aquí permite la movilización y la integración de distintos saberes en la búsqueda de soluciones posibles a una situación dada. Si bien la información aquí fue presentada a modo de instructivo, se espera que sean los estudiantes organizados en pequeños grupos quienes vayan encontrando las mejores formas para construir los dispositivos. Esto implica preparar los materiales para que cada grupo cuente con todo lo necesario para la construcción del proyecto. Además, al interior de cada grupo, los estudiantes deben distribuirse los roles y las tareas de acuerdo a las demandas que van teniendo en las actividades.

Es importante que los docentes acompañen las producciones de cada grupo monitoreando los avances de todos los estudiantes y presentando la información que se considere necesaria para continuar la tarea. Pero, al mismo tiempo, es necesario que habiliten espacios para que los alumnos realicen hipótesis, planteen interrogantes, indaguen, prueben y realicen ajustes de acuerdo a lo que ellos mismo van pensando sobre cómo llevar a cabo el proyecto.

En este sentido, registrar lo que se va haciendo, las preguntas de los alumnos, las pruebas, los errores y cómo se fueron construyendo los dispositivos, permite reflexionar sobre la propia práctica, reforzar los aprendizajes construidos a lo largo de este proceso y poder volver a ese material disponible para próximos proyectos que se realicen.

Una vez terminado el proyecto, se sugiere reunir y organizar con el grupo el registro que se hizo del proceso realizado. Esta instancia de sistematización también permite movilizar capacidades vinculadas a la comunicación porque implica tomar decisiones respecto a cómo se quiere mostrar el proyecto a otros (otros grupos, otras escuelas, otros docentes, a la comunidad, etc.).

## Reconocimientos

Este trabajo es fruto del esfuerzo creativo de un enorme equipo de entusiastas y visionarios de la pedagogía de la innovación, la formación docente, la robótica, la programación, el diseño y la impresión 3D. Les agradecemos por el trabajo en equipo inspirador para traer a la realidad la obra que, en forma conjunta, realizamos INET y EDUCAR del Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación Argentina.

### **Contenidos**

#### **Equipo INET**

Alejandro Anchava  
Joreliz Andreyana Aguilera Barragán  
Omar Leandro Bobrow  
Alejandro Cesar Cáceres  
Ezequiel Luberto  
Gustavo Roberto Mesiti  
Alejandro Palestrini  
Judit Schneider  
Pablo Trangone

#### **Equipo Educar:**

Pablo Aristide  
Mayra Botta  
Anabela Cathcarth  
Eduardo Chiarella  
María Laura Costilla  
Diego Dorado  
Facundo Dyszel  
Federico Frydman  
Matías Rinaldi  
Uriel Rubilar  
Camila Stecher  
Carolina Sokolowicz  
Nicolás Uccello

Para la confección de esta obra se contó con el apoyo de la Universidad Pedagógica Nacional "UNIPE". En particular en el desarrollo de los capítulos 1 y 2, los cuales estuvieron a cargo de los profesores Fernando Raúl Alfredo Bordinon y Alejandro Adrián Iglesias.

### **Producción y comunicación**

Juliana Zugasti

### **Diseño y edición**

Leonardo Frino  
Mario Marrazzo

## **Corrección de estilo**

María Cecilia Alegre

## **Agradecimientos especiales**

Mariano Consalvo. Equipo ABP

Damián Olive. Equipo de ABP

María José Licio Rinaldi, Directora Nacional de Asuntos Federales INET, quien siempre acompañó a este equipo en todas las gestiones para su implementación

Estamos comprometidos en instalar la innovación en la escuela secundaria técnica: la robótica, la programación, el pensamiento computacional, los proyectos tecnológicos, el ABP, la impresión 3D, de manera más accesible para todos.

Agradecemos enormemente, docente, tu continua dedicación y compromiso con el futuro de tus estudiantes.

***¡Estamos ansiosos por saber qué es lo que vamos a crear juntos!***