

# BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI)

*EXPEDIENTE:* IDI-20150289

Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

*ACRÓNIMO DEL PROYECTO:* BOTBLOQ



Centro para el  
Desarrollo  
Tecnológico  
Industrial



UNIÓN EUROPEA

Fondo Europeo de  
Desarrollo Regional (FEDER)

*Una manera de hacer Europa*

## ENTREGABLE E.3.2.1 INFORME DE LA APLICACIÓN

---

### RESUMEN DEL DOCUMENTO

Para conseguir una mayor adaptabilidad de la aplicación, en la que los usuarios programan en bloques a un nivel más alto y con un hardware más afín a las características de los usuarios, se ha creado la aplicación para robots en Bitbloq. Una serie de bloques de programación que permiten programar funciones predefinidas por un conjunto de sensores y actuadores dentro de un mismo robot.

---

February 3, 2016

## Contents

<b>1</b>	<b>Introducción a los módulos de programación de Robots en Bit-bloq</b>	<b>2</b>
<b>2</b>	<b>Módulo de programación para alumnos entre 5 y 10 años</b>	<b>3</b>
2.1	Introducción al proyecto Zowi . . . . .	3
2.2	Diseño y desarrollo de los bloques Zowi . . . . .	5
2.3	Casos de uso . . . . .	6
2.3.1	Hello World . . . . .	6
2.3.2	Primera secuencia . . . . .	7
2.3.3	Secuencia en bucle principal . . . . .	7
2.3.4	Condicionando Sensores . . . . .	8
2.3.5	Funciones . . . . .	10
2.3.6	Juego de Programación (Coreografía Caótica Rotacional) . .	11
<b>3</b>	<b>Módulo de programación para alumnos entre 10 y 16 años</b>	<b>13</b>
3.1	Introducción al proyecto PrintBot . . . . .	13
3.2	Casos de uso . . . . .	13
3.2.1	¿Cómo funciona un sigue-líneas? . . . . .	14
3.2.2	Programación de un robot sigue-líneas . . . . .	14
3.2.3	Variantes de programación (Optimizando código) . . . . .	17

# 1 Introducción a los módulos de programación de Robots en Bitbloq

Se desarrollarán dos módulos, uno orientado a niños entre 5 y 10 años y otro para niños entre 10 y 16 años. Estos módulos deben permitir programar las plataformas robóticas, realizando acciones condicionales en función de los sensores de las plataformas y según las posibilidades de cada plataforma. Deberán ser compatibles con el sistema bitbloq actual.

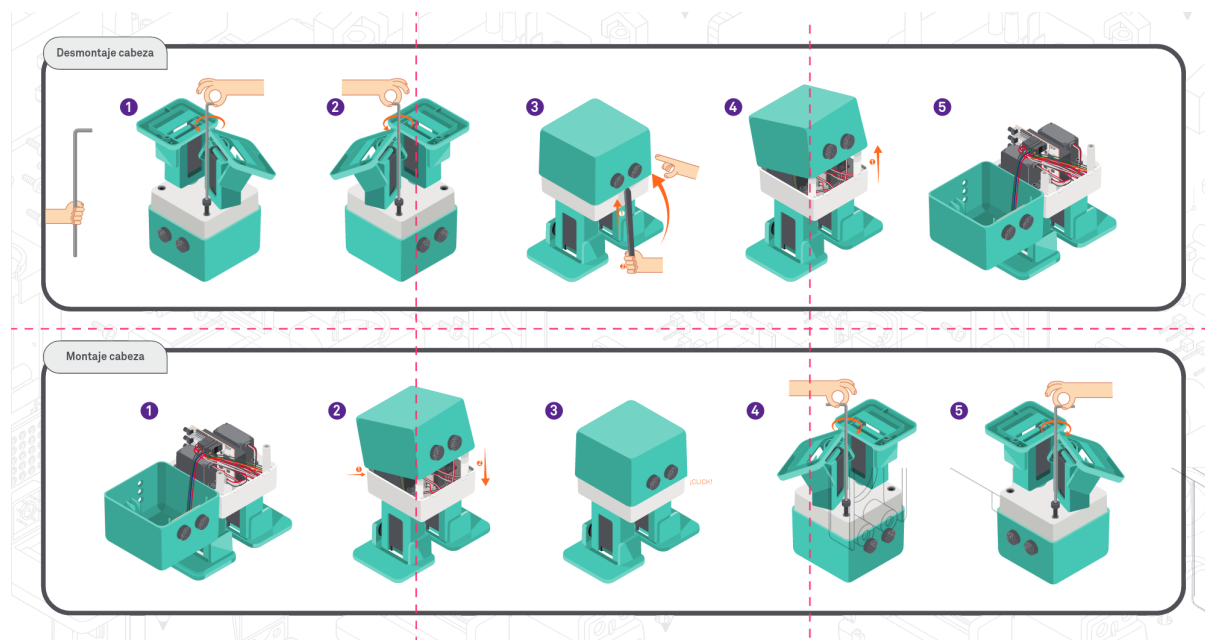
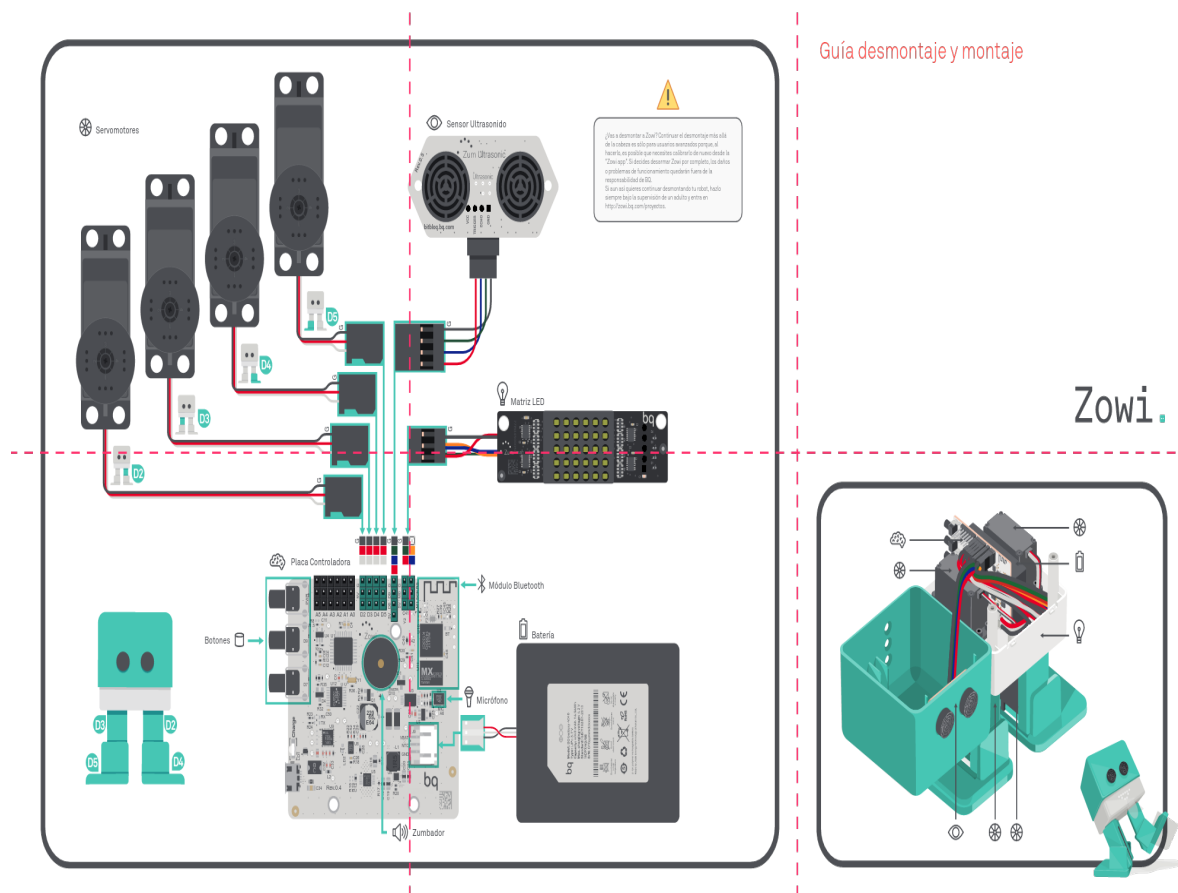
## 2 Módulo de programación para alumnos entre 5 y 10 años

### 2.1 Introducción al proyecto Zowi

Con una versión estable de Bitbloq que facilitaba la programación de componentes hardware independientes, se planteó crear un robot que pudiese ser programado en Bitbloq por alumnos a partir de 5 años de la forma más sencilla e intuitiva posible.

En una primera fase, se establece que la programación de Zowi sea mediante acciones completas para el robot o, lo que es lo mismo, mediante un lenguaje de alto nivel, facilitando así su utilización para los alumnos de menor edad. Posteriormente, se planifica a futuro dar la posibilidad de controlar cada uno de los componentes de Zowi por separado, orientado a alumnos más avanzados.

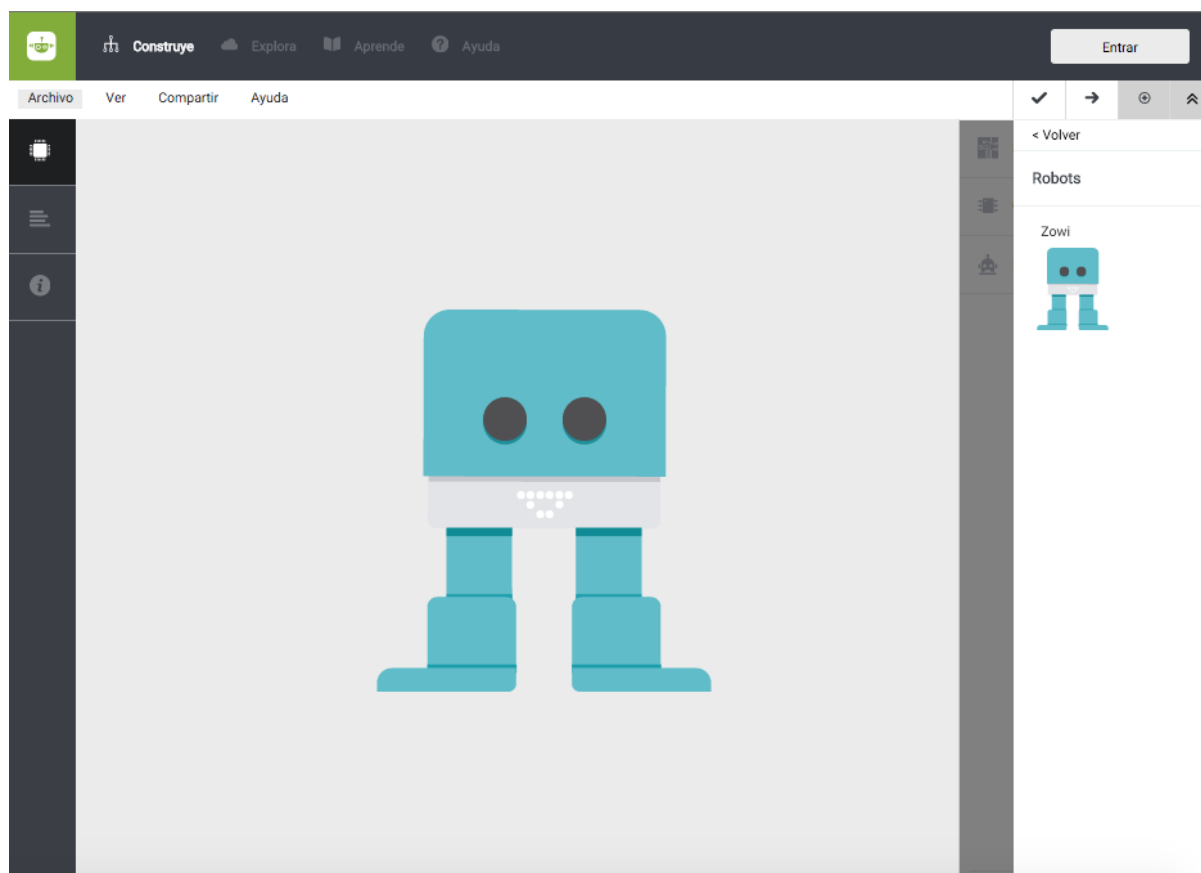
La arquitectura de Zowi se compone de un micrófono, un sensor de ultrasonidos, dos botones, un zumbador, una matriz led de 6x5 y cuatro servos.

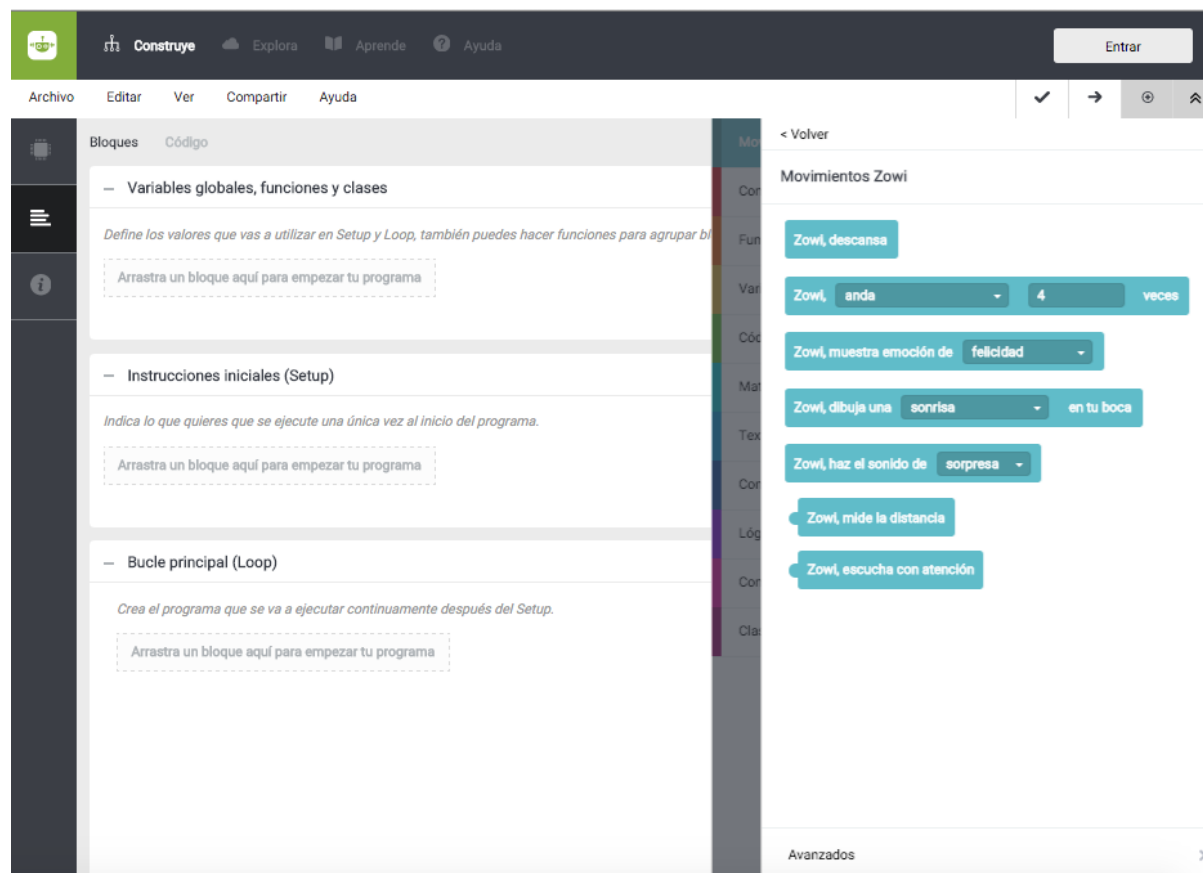


## 2.2 Diseño y desarrollo de los bloques Zowi

Aprovechando tanto la estructura de bloques como las librerías ya generadas para los diversos componentes, se desarrolló una nueva librería que utilizaba aquellas para definir funciones que ejecutaban acciones complejas. Esta librería se encuentra en: <https://github.com/bq/bitbloqLibs/tree/master/BitbloqZowi>

A partir de esta librería, se generaron bloques con la misma estructura que el resto de Bitbloq utilizando dichas funciones, lo que permitía realizar conjuntos de acciones simplemente añadiendo un único bloque. EL repositorio de estos bloques se encuentra en: <https://github.com/bq/bitbloqLibs/tree/master/BitbloqZowi>





## 2.3 Casos de uso

De cara a analizar la experiencia del usuario, localizar posibles mejoras y dar a conocer el producto, se realizaron varios talleres en los que se presentó la herramienta Bitbloq y el robot Zowi siguiendo el siguiente guión:

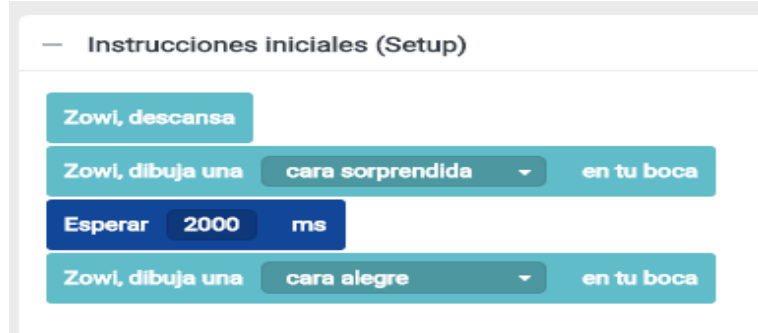
### 2.3.1 Hello World

Zowi pone carita en *Setup()* (Y también descansa)



### 2.3.2 Primera secuencia

>> *Ejercicio propuesto:* Zowi debe poner cara sorprendida y después una feliz.



CONCLUSIÓN: Se deben dar cuenta de meter **la espera**.

### 2.3.3 Secuencia en bucle principal



CONCLUSIÓN: explicación *Loop()* vs *Setup()* y necesidad de **segunda espera**.

NOTA: Dejar en el *Setup()* el bloque Zowi descansa.



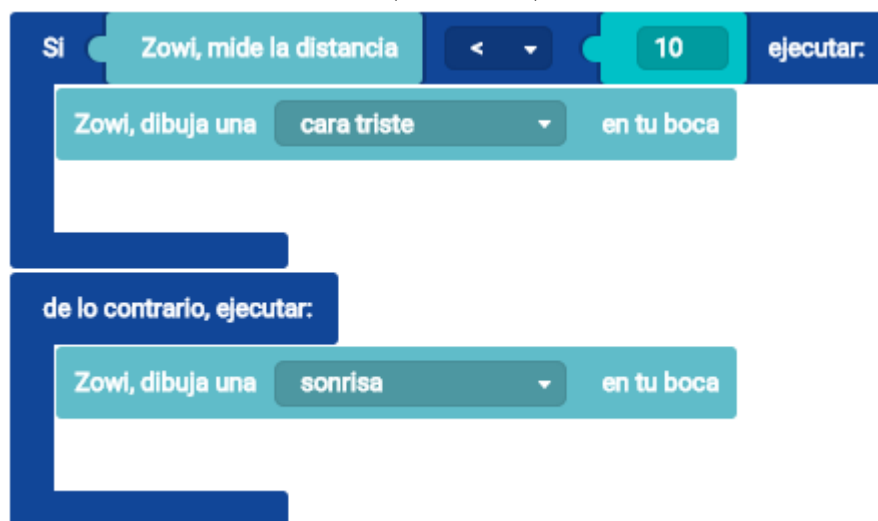
### 2.3.4 Condicionando Sensores

#### ● Zowi reacciona Ruido

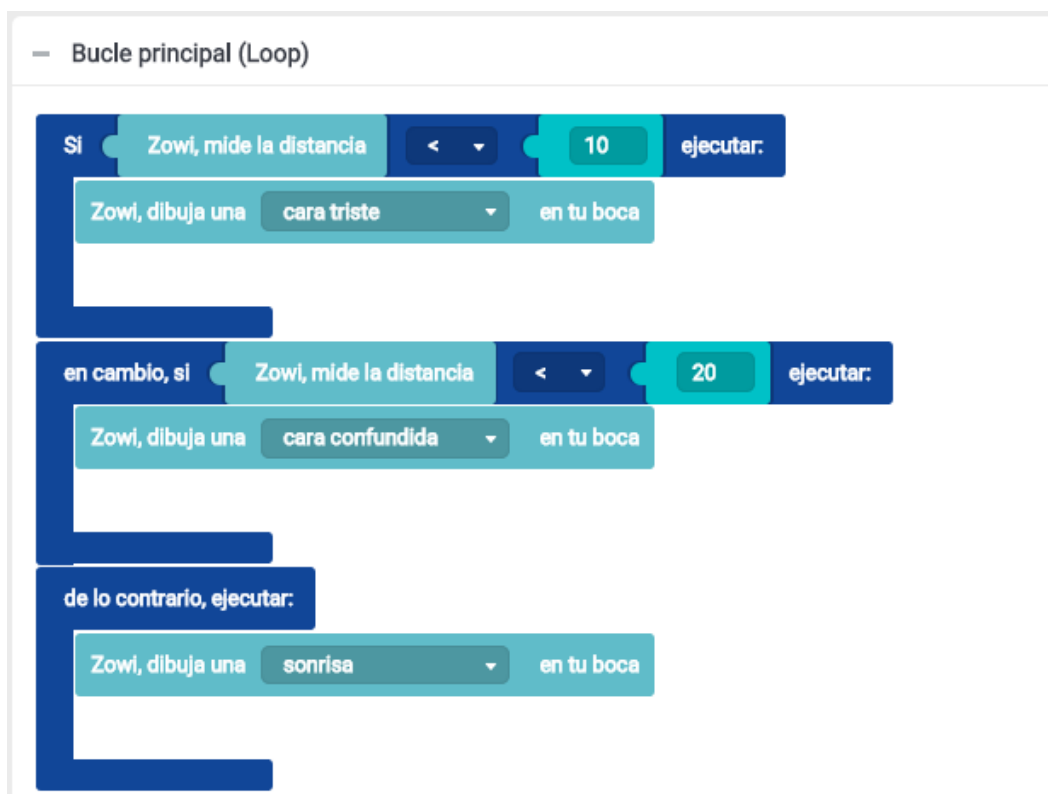
Si... ejecutar // de lo contrario...



#### ● Zowi reacciona Distancia (sensor US)



En cambio si...



Mientras....



### 2.3.5 Funciones

Si tenemos una secuencia muy larga y con “**partes que se repiten**” como esta...



Podemos reducirla a...



Esto se hace declarando lo que se llama una “función”:



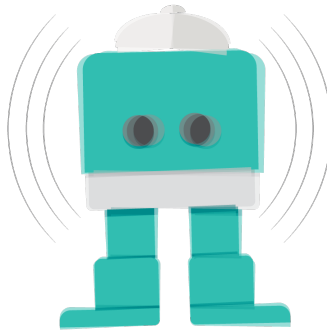
### 2.3.6 Juego de Programación (Coreografía Caótica Rotacional)

Se trata de crear una fantástica coreografía caótica y rotacional entre todos. Al final se verán todas y se votará la más original y caótica. Las pasos a seguir con los siguientes:

- Cuando el instructor de la señal, baja los brazos.
- Programa a toda prisa a Zowi con movimientos, funciones, gestos, sonido...
- Cuando suene la alarma... *¡cambio de puesto!*
- Pon las dos manos en la cabeza y muévete al siguiente puesto (ordenador).

NOTA: Cada cierto tiempo avisar de programación obligatoria: *¡ORDEN SUPREMA!*

- - *Debes meter una cara\_sorpresa y sonido Oh Oh.*
  - *Una espera de medio segundo. → (500ms)*
  - *Meter una cara que no se vea. → (porque va después de otra cara)*
  - *Que aparezca el número “3” en algún sitio.*
  - *Meter algo en Setup()*



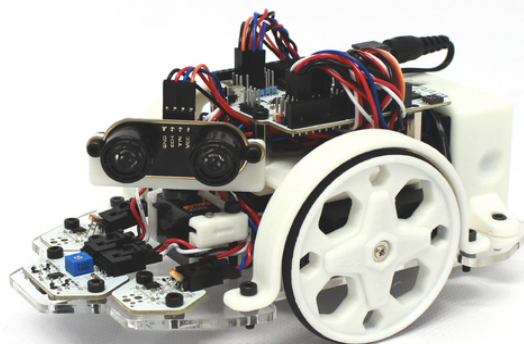
## 3 Módulo de programación para alumnos entre 10 y 16 años

### 3.1 Introducción al proyecto PrintBot

De la misma forma que Zowi está orientado para alumnos de menor edad, el proyecto PrintBot está más planteado para la programación en Bitbloq en alumnos a partir de 10 años.

Sin embargo, al contrario que Zowi, su primera fase consiste en la programación de cada uno de sus componentes de forma individual, lo que requiere un mayor conocimiento de programación. Del mismo modo, se planifica a futuro la posibilidad de establecer una programación para los PrintBots que sea mediante acciones completas, es decir, mediante instrucciones de alto nivel.

Aunque la arquitectura de cada uno de los PrintBots es diferente, la mayoría comparten unos componentes comunes: dos sensores IR y dos servos de rotación continua.



### 3.2 Casos de uso

Siguiendo con la metodología empleada en los talleres de Zowi, el guión utilizado para los talleres de los Printbots fue el siguiente:

### 3.2.1 ¿Cómo funciona un sigue-líneas?

El objetivo de nuestro sigue-líneas es mantenerse dentro de una línea negra. Para ello contamos con los **dos sensores IR** capaces de distinguir entre el color negro y otro color que no sea el negro, lo que nos permitirá saber si nuestro PrintBot está, no está o se está saliendo de una línea negra.

Además contamos con los **dos servos de rotación continua** como actuadores, la rueda izquierda y la rueda derecha.

Si analizamos la situación, vemos que al avanzar por una línea negra, los dos sensores IR van a poder encontrarse con **cuatro situaciones diferentes**:

1. Los dos sensores detectan negro
2. El izquierdo detecta negro y el derecho blanco (se sale por la derecha)
3. El derecho detecta negro y el izquierdo blanco (se sale por la izquierda)
4. Los dos sensores detectan blanco

Cuando se salga por un lado, tenemos que frenar la rueda correcta para orientar al sigue-líneas de nuevo hacia la línea negra, mientras la otra sigue avanzando. Y si los dos sensores IR detectan blanco, detenemos al robot para que no se pierda.

### 3.2.2 Programación de un robot sigue-líneas

Una vez entendido qué hacer en cada estado, tendremos que traducir esto a un lenguaje de programación que entienda nuestra placa controladora.

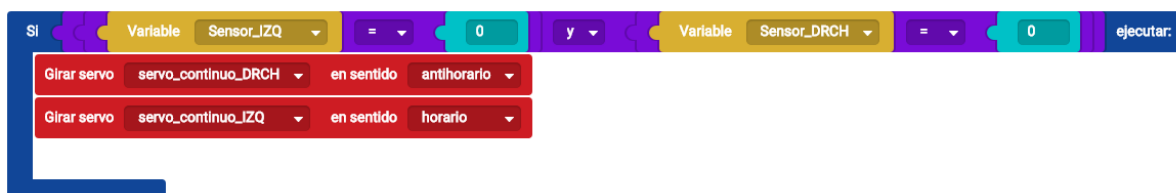
Lo primero será crear las variables necesarias para almacenar el valor de los sensores IR:



En estas variables se guardarán un 0 si el sensor detecta Negro y un 1 si detecta otro color. Aunque tengáis el sensor IR doble, vamos a usar los bloques de sensores individuales porque son más versátiles.

A continuación tenemos que ir definiendo los 4 estados del robot. Para ello usaremos el bloque del condicional “si...ejecutar” y como condición, nos ayudaremos del bloque “...Y...” para darle dos condiciones a cada estado (El valor de cada sensor).

- El estado en el que los dos sensores IR detectan negro y el robot avanza sería el siguiente:



Fijaros en que cada motor gira en una dirección diferente. Esto es porque al estar uno al lado del otro mirando en dirección contraria, sus sentidos de giro están invertidos el uno del otro.

Para que los dos vayan en la misma dirección, tenemos que hacer girar los servos de la forma aquí indicada. Si no lo entendéis del todo, imaginad que estáis nadando a estilo mariposa y analizar el sentido de giro de cada brazo.

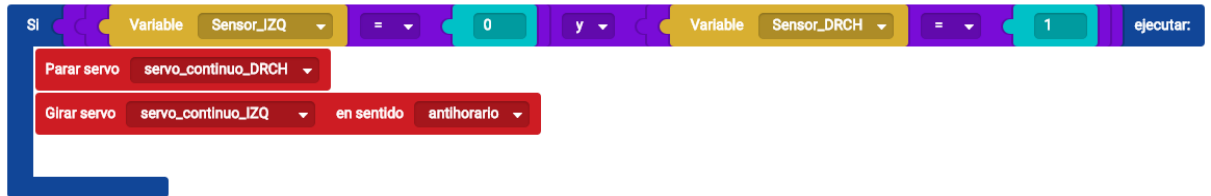
Solo queda añadir cada estado según lo que hemos analizado anteriormente:

- El sensor izquierdo detecta negro y el derecho blanco (se sale por la derecha)

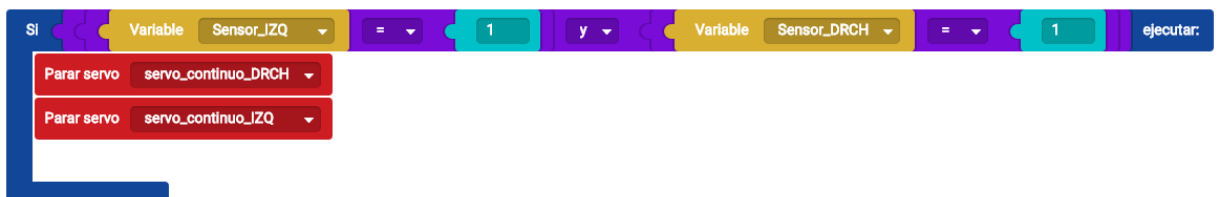




- El sensor derecho detecta negro y el izquierdo blanco (se sale por la izquierda)



- Los dos sensores detectan blanco



Con esto ya tendríamos nuestro robot sigue-líneas.



### 3.2.3 Variantes de programación (Optimizando código)

El código que hemos creado antes es completamente funcional, de hecho, nos permite modificar que queremos que haga el robot cuando se salga completamente de la línea, ya que tenemos todo un estado dedicado a esa posibilidad.

Pero existe una forma mucho más corta de realizar el programa de arriba. A esto le llamamos “Optimización del código”. Vamos a analizarla:



Si pensamos detenidamente en el funcionamiento del sigue-líneas, podemos observar que el sensor IR derecho es el único que determina el funcionamiento del servo izquierdo: Si detecta negro, avanza, si detecta blanco, se detiene. Lo mismo pasa con la otra pareja de componentes. Probar el nuevo código y veréis cómo funciona también.

Por otro lado, hemos decidido que cuando los dos sensores detecten blanco, el PrintBot se detenga, pero podríamos cambiar este comportamiento por otro cualquiera, como rotar sobre sí mismo para encontrar la línea negra, retroceder o avanzar recto hasta encontrar de nuevo la línea.