

ROBÓTICA EDUCATIVA MEDIANTE PROGRAMACIÓN VISUAL

EDUCATIONAL ROBOTICS THROUGH VISUAL PROGRAMMING

Trabajo de Fin de Grado
Andrea Rodríguez Rivarés

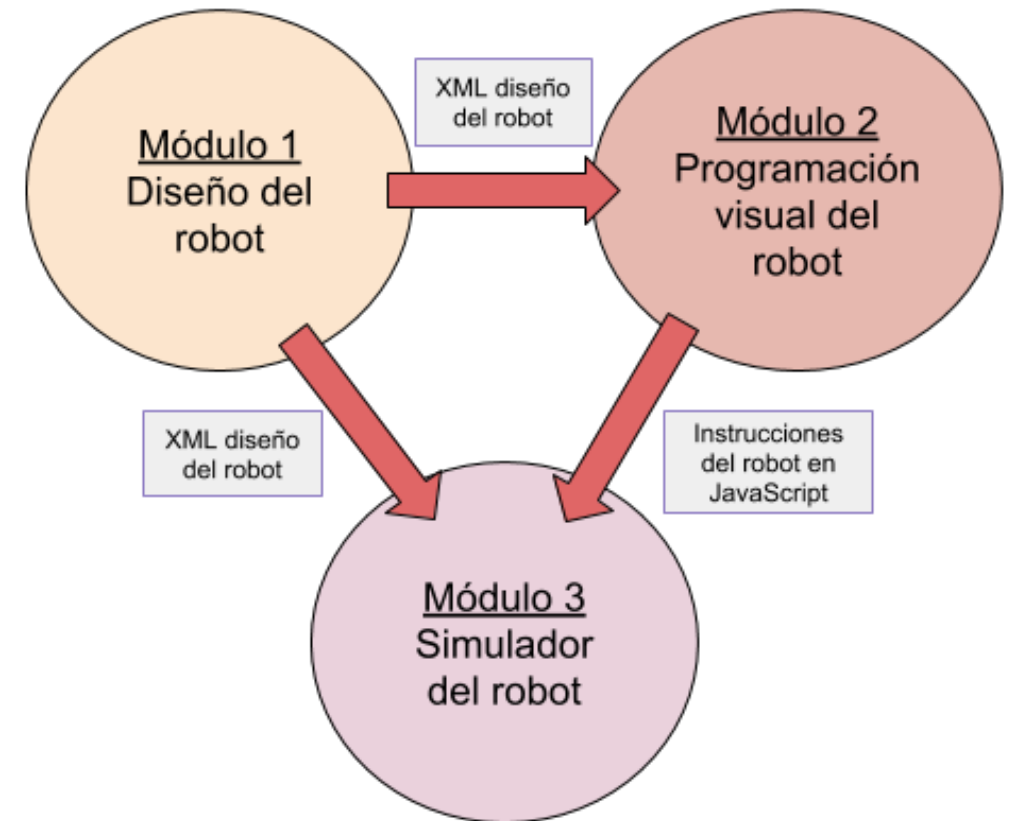
Índice de contenido

1. Introducción
2. Contexto
 1. Antecedentes
 2. Lenguaje de Programación Visual
3. Desarrollo del proyecto
 1. Toolbox
 2. Botones
4. Funcionamiento a nivel interno
5. Caso de uso
 1. Ejemplo de funcionamiento
6. Problemas y soluciones
7. Conclusions and future works
8. Bibliografía

Introducción

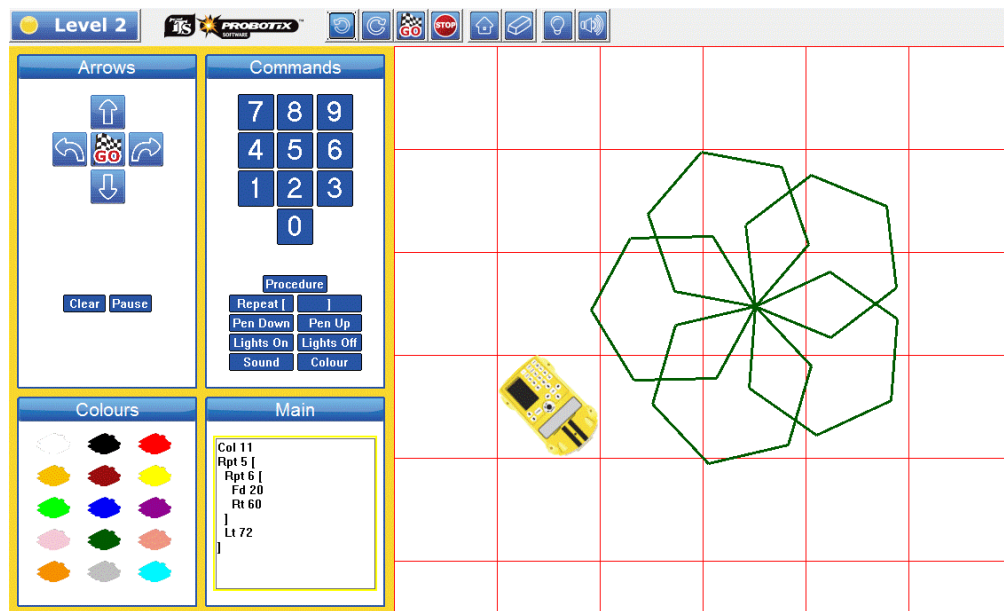
- Objetivo Principal

Proporcionar un lenguaje de programación visual que permita llevar a cabo la programación del comportamiento de un robot virtual en un entorno simulado.

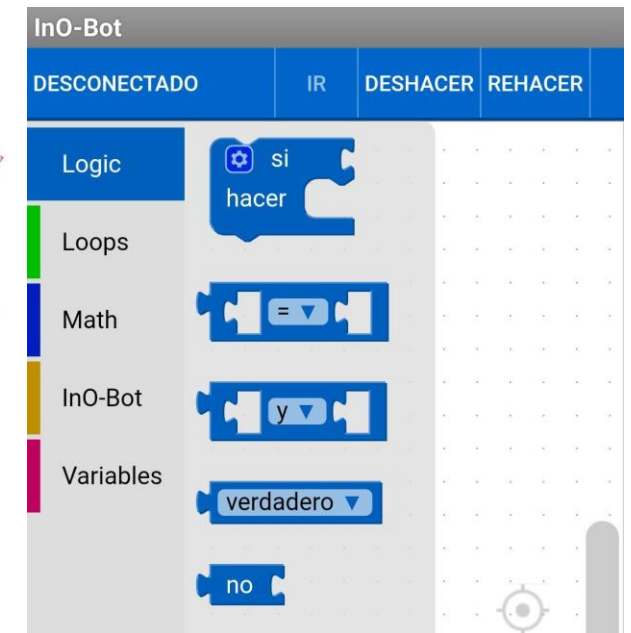
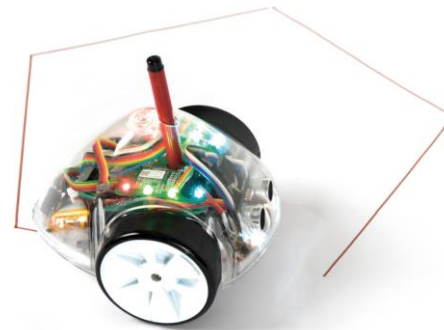


Contexto. Antecedentes

Pro-Bot

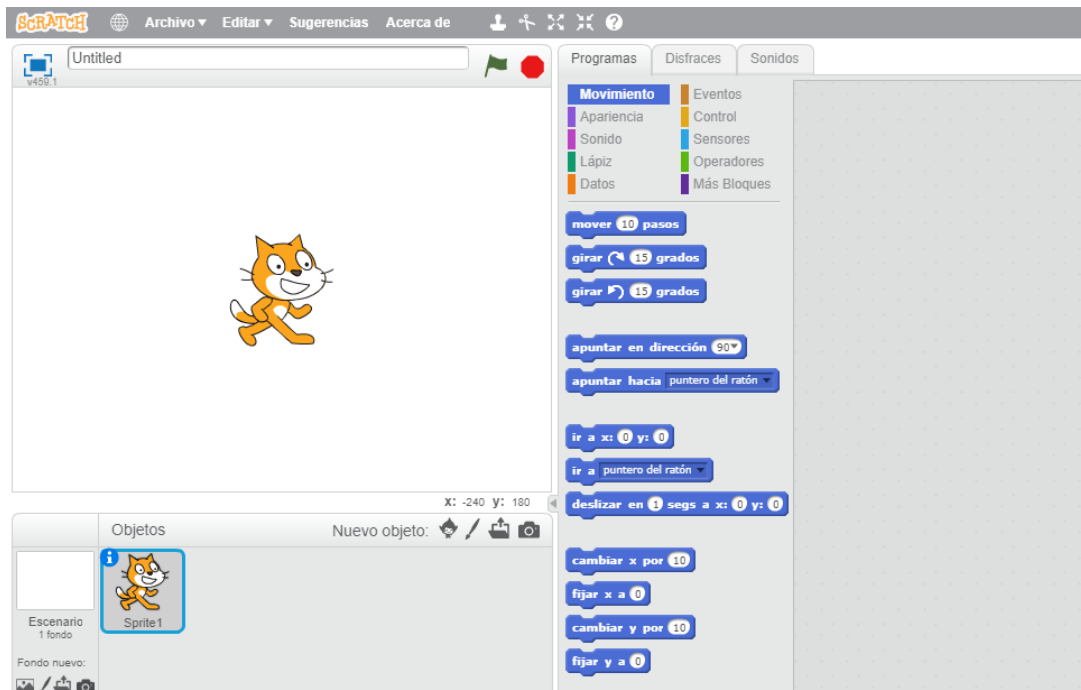


Ino-Bot

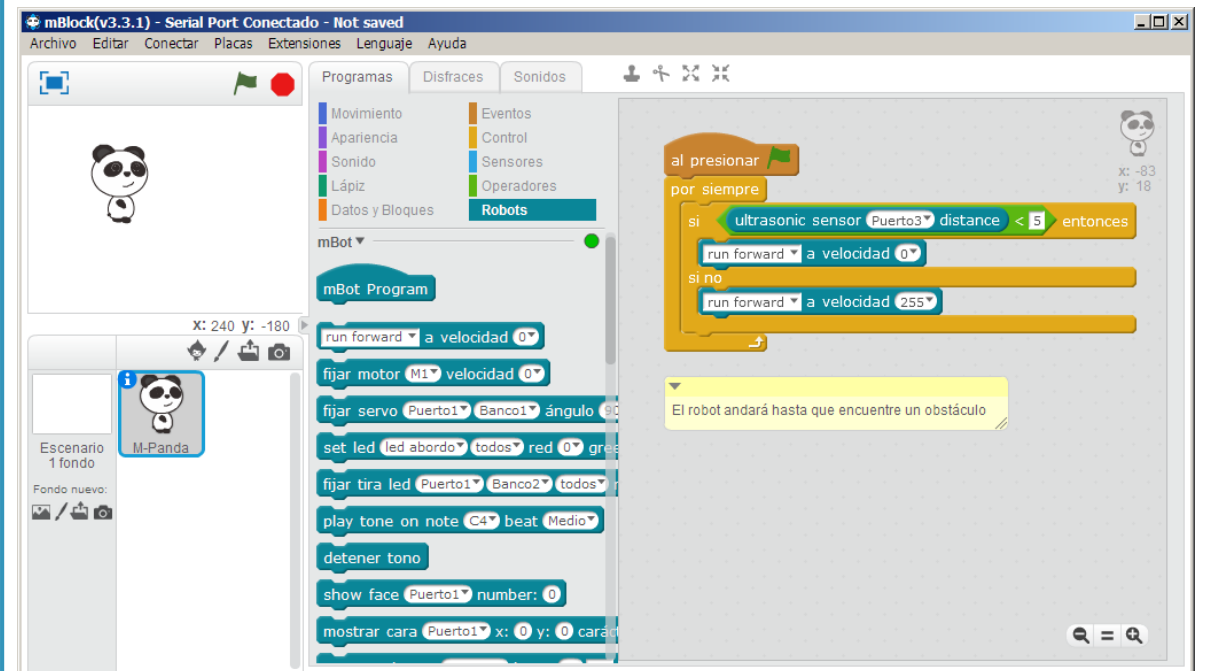


Contexto. Lenguaje de Programación Visual

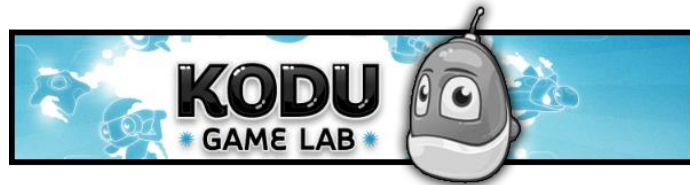
SCRATCH



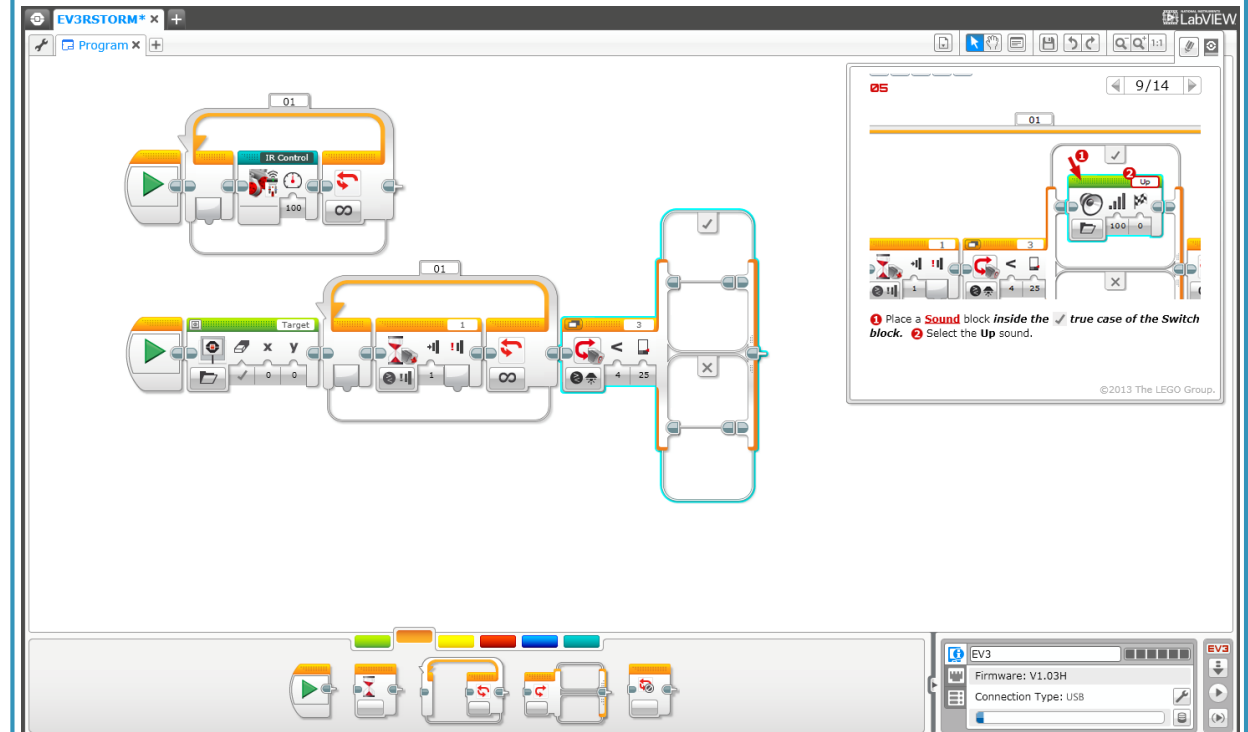
mBlock



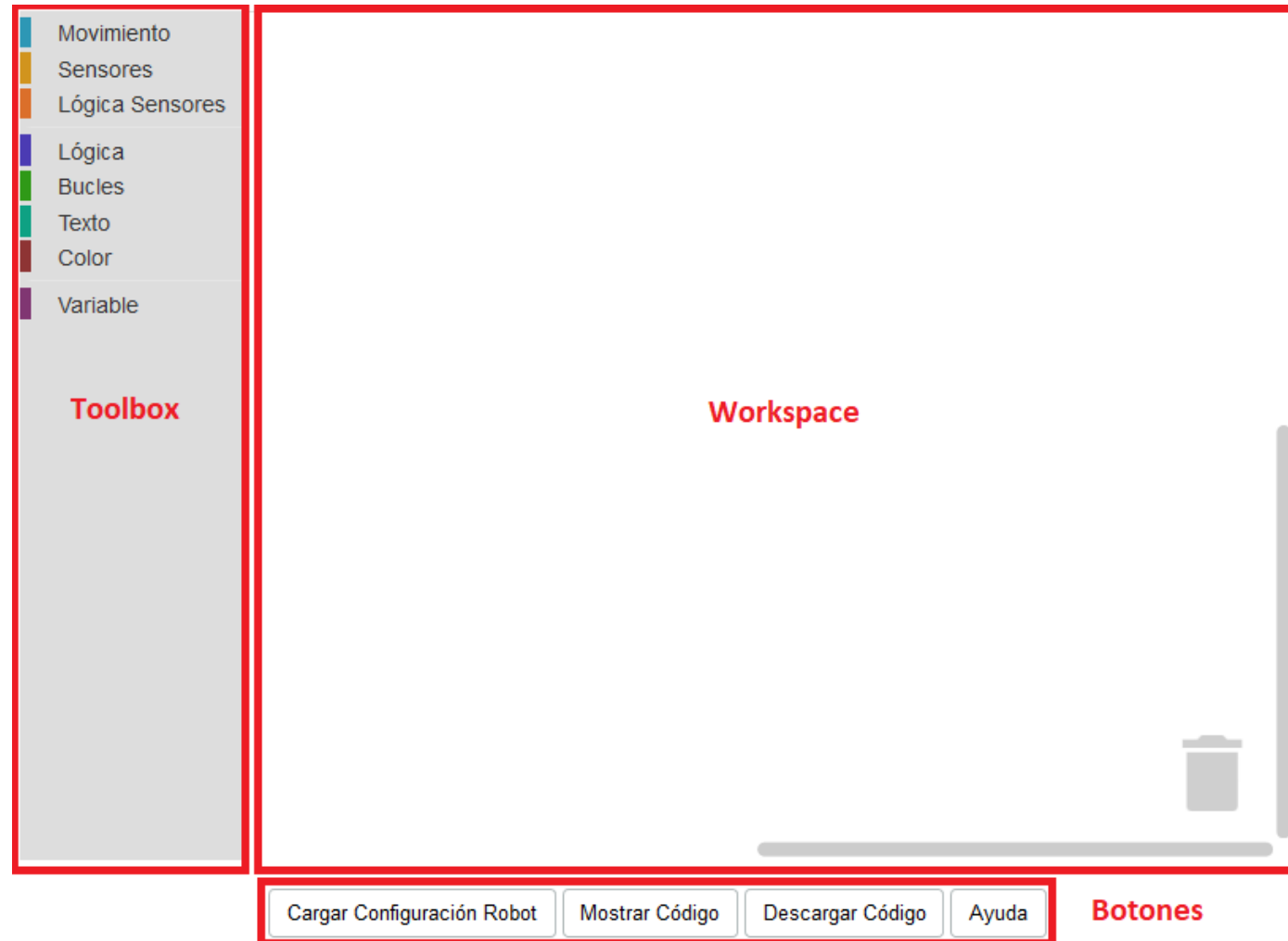
Contexto. Lenguaje de Programación Visual



APP EV3 PROGRAMMER



Desarrollo del proyecto



Desarrollo del proyecto. Toolbox

Movimiento

Sensores

Lógica Sensores

Lógica

Bucles

Texto

Color

Variable

Avanzar 5 metros por 1 seg

Retroceder 5 metros por 1 seg

Girar 90° izquierda

Detener movimiento

Movimiento

Sensores

Lógica Sensores

Lógica

Bucles

Texto

Color

Variable

si sensor = hacer

sino

sino si sensor = hacer

Movimiento

Sensores

Lógica Sensores

Lógica

Bucles

Texto

Sensor Láser nº 1

Sensor de Contacto nº 1

Sensor de Ultrasonido nº 1

Sensor Infrarrojo nº 1

Sensor Odómetro nº 1

Desarrollo del proyecto. Toolbox

The image displays the Scratch toolbox, organized into categories. The 'Lógica' category is highlighted in purple. The toolbox contains the following blocks:

- Movimiento** (Movement): 'si hacer' (if-do), 'repetir veces' (repeat times), 'repetir mientras' (repeat while), 'contar con i desde hasta' (count with i from to), 'color aleatorio' (random color), '“ ”' (quote), and 'imprimir' (print).
- Sensores** (Sensors): 'número aleatorio entre 1 y 10' (random number between 1 and 10), 'nulo' (null), and 'color aleatorio' (random color).
- Lógica Sensores** (Logic Sensors): 'si hacer' (if-do), 'repetir veces' (repeat times), 'repetir mientras' (repeat while), 'contar con i desde hasta' (count with i from to), 'color aleatorio' (random color), '“ ”' (quote), and 'imprimir' (print).
- Lógica** (Logic): 'si hacer' (if-do), 'repetir veces' (repeat times), 'repetir mientras' (repeat while), 'contar con i desde hasta' (count with i from to), 'color aleatorio' (random color), '“ ”' (quote), and 'imprimir' (print).
- Bucles** (Loops): 'repetir veces' (repeat times), 'repetir mientras' (repeat while), and 'contar con i desde hasta' (count with i from to).
- Texto** (Text): '“ ”' (quote) and 'imprimir' (print).
- Color** (Color): 'color aleatorio' (random color).
- Variable** (Variable): 'Crear variable...' (create variable...), 'establecer var1 a' (set var1 to), 'añadir 1 a var1' (add 1 to var1), and 'var1' (variable).

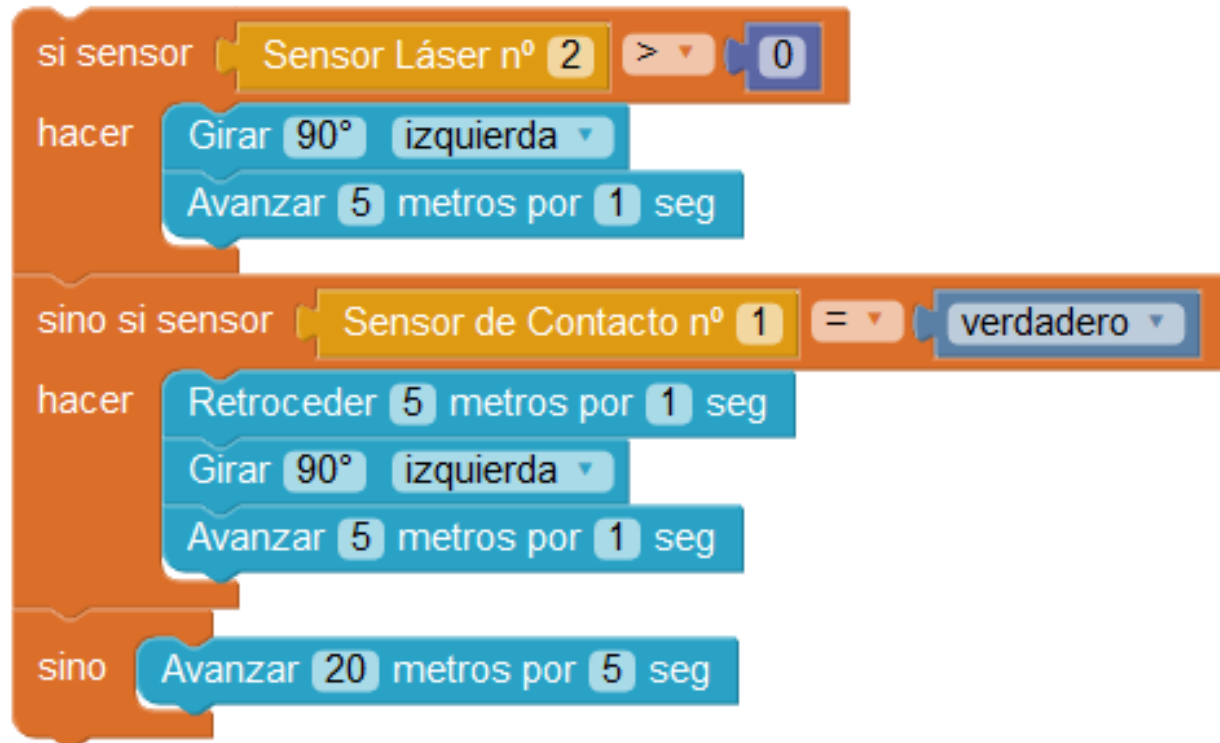
Desarrollo del proyecto. Botones

Cargar Configuración Robot

Mostrar Código

Descargar Código

Ayuda



```
if ( sensor_laser_2 > 0 ) {  
  Girar(90,'0');  
  Avanzar(5,1);  
}  
else if ( sensor_contacto_1 = true ) {  
  Retroceder(5,1);  
  Girar(90,'0');  
  Avanzar(5,1);  
}  
else {  
  Avanzar(20,5);  
}
```

Aceptar

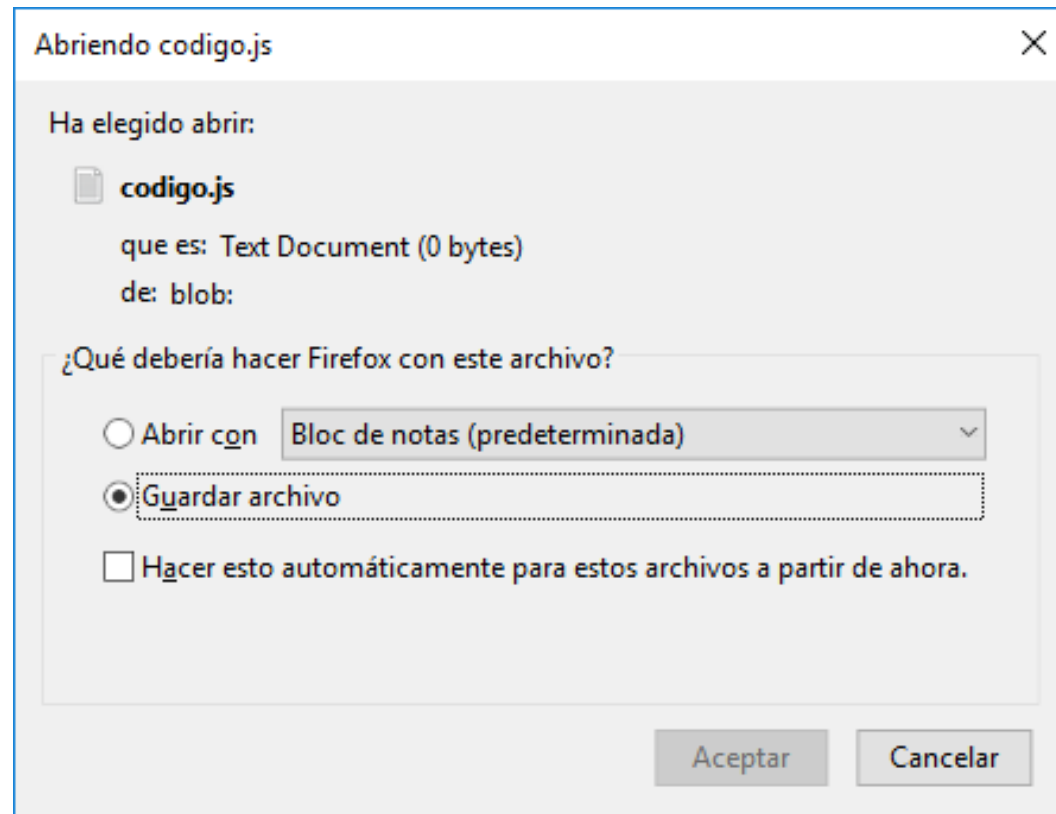
Desarrollo del proyecto. Botones

Cargar Configuración Robot

Mostrar Código

Descargar Código

Ayuda



Desarrollo del proyecto. Botones

Cargar Configuración Robot

Mostrar Código

Descargar Código

Ayuda

Ejemplo 1: Uso de la aplicación



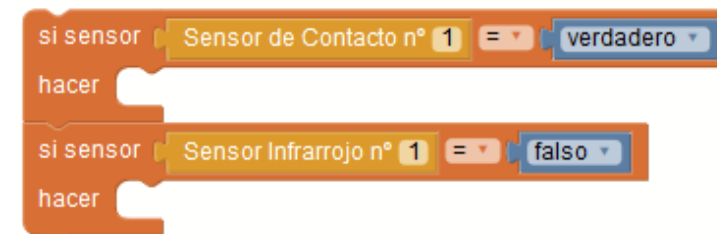
Ejemplo 2: Uso de la aplicación



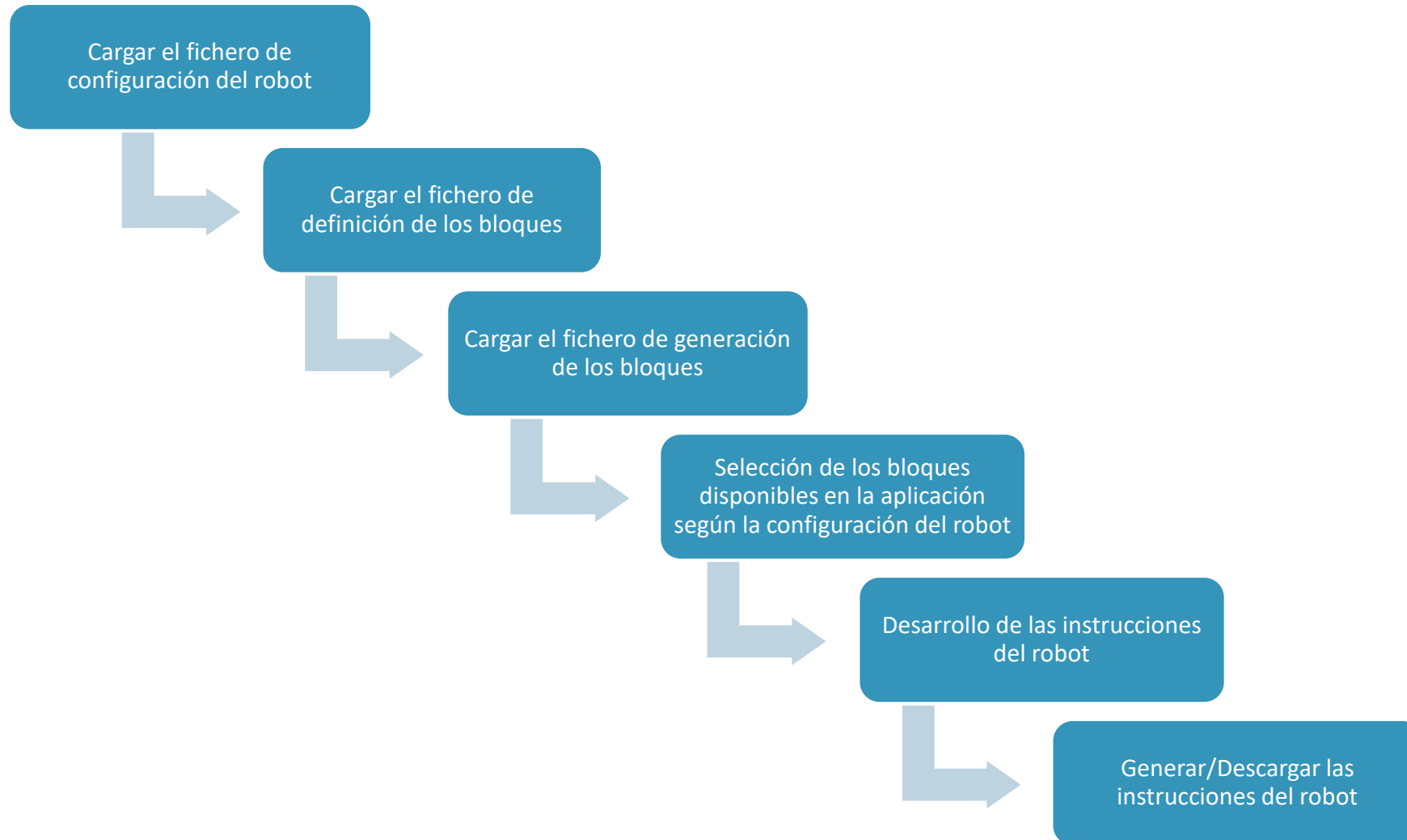
Ejemplo 3: Uso de los sensores Láser, Ultrasonido y Láser 2D



Ejemplo 4: Uso de los sensores de Contacto e Infrarrojo



Funcionamiento a nivel interno



Caso de uso. Ejemplo de funcionamiento

Enlace a la aplicación desplegada en Heroku:

<http://blockly-tfg.herokuapp.com/>

Problemas y soluciones

Estudio de la herramienta Blockly y desarrollo de bloques básicos

- **Problema:** Documentación abundante pero pocos ejemplos.
- **Solución:** Dedicación de más horas de lo esperado para desarrollar el entorno de Blockly y bloques básicos.

Lenguaje de salida de los bloques

- **Problema:** El lenguaje de salida de los bloques debe ser leído por el simulador.
- **Solución:** Utilizar JavaScript como lenguaje de salida de los bloques.

Paso de ficheros entre los distintos módulos

- **Problema:** Al encontrarse desarrollados los módulos en distintas plataformas, el paso de los ficheros entre los mismos resulta ser complejo.
- **Solución:** Carga manual de los ficheros en los distintos módulos.

Conclusions and future works

Bibliografía

- Google. Manual Blockly. Accedido 20/03/2018. Sitio web: <https://developers.google.com/blockly/>
- Google. Accedido 20/03/2018. Sitio web:
<https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>
- Robótica Educativa Wikipedia. Accedido 11/06/2018. Sitio web:
https://es.wikipedia.org/wiki/Rob%C3%B3tica_educativa
- RO-BOTICA. (2017). PRO-BOT Robot infantil programable. Accedido 08/06/2018. Sitio web:
<http://ro-botica.com/Producto/PRO-BOT/>
- RO-BOTICA. (2017). InO-Bot Robot infantil programable. Accedido 08/06/2018. Sitio web:
<https://www.ro-botica.com/Producto/INO-BOT/>
- Scratch. Accedido 12/06/2018. Sitio web: <https://scratch.mit.edu/>