

# ROBÓTICA EDUCATIVA MEDIANTE PROGRAMACIÓN VISUAL

---

EDUCATIONAL ROBOTICS THROUGH VISUAL PROGRAMMING

Trabajo de Fin de Grado  
Andrea Rodríguez Rivarés

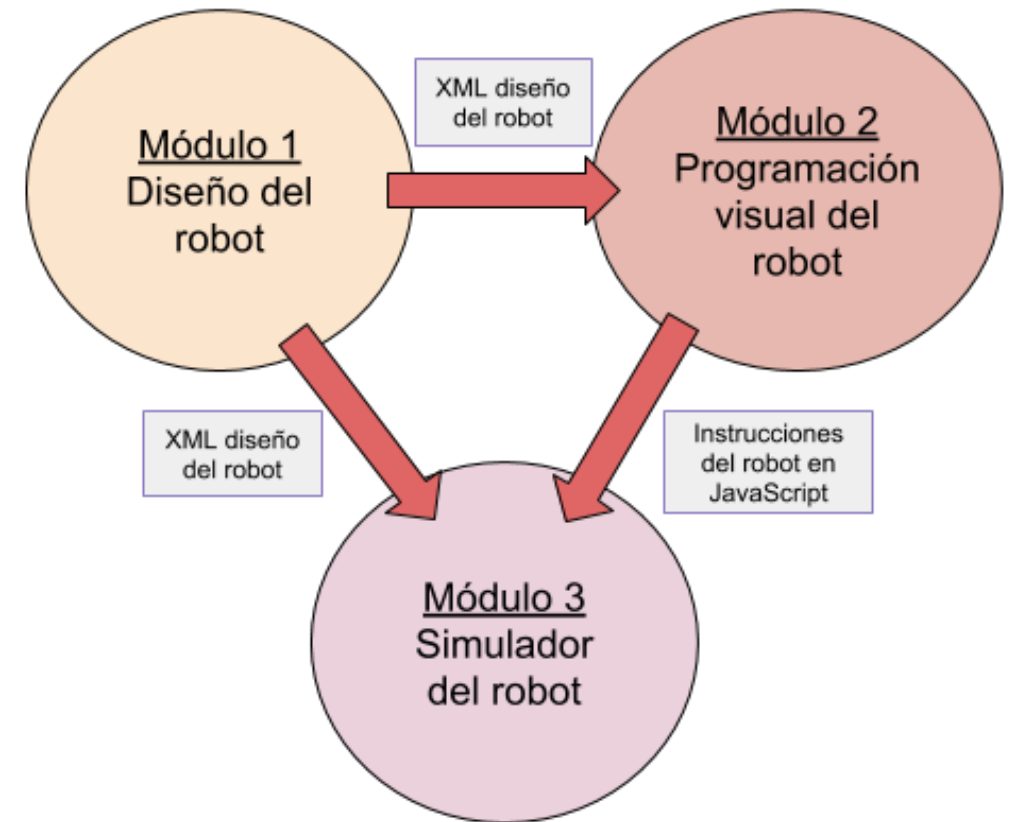
# Índice de contenido

1. Introducción
2. Contexto
  1. Antecedentes
  2. Lenguaje de Programación Visual
3. Fases del desarrollo
4. Desarrollo del proyecto
  1. Toolbox
  2. Botones
5. Funcionamiento a nivel interno
6. Problemas y soluciones
7. Caso de uso
  1. Ejemplo de funcionamiento
8. Conclusiones y trabajos futuros
9. Bibliografía

# Introducción

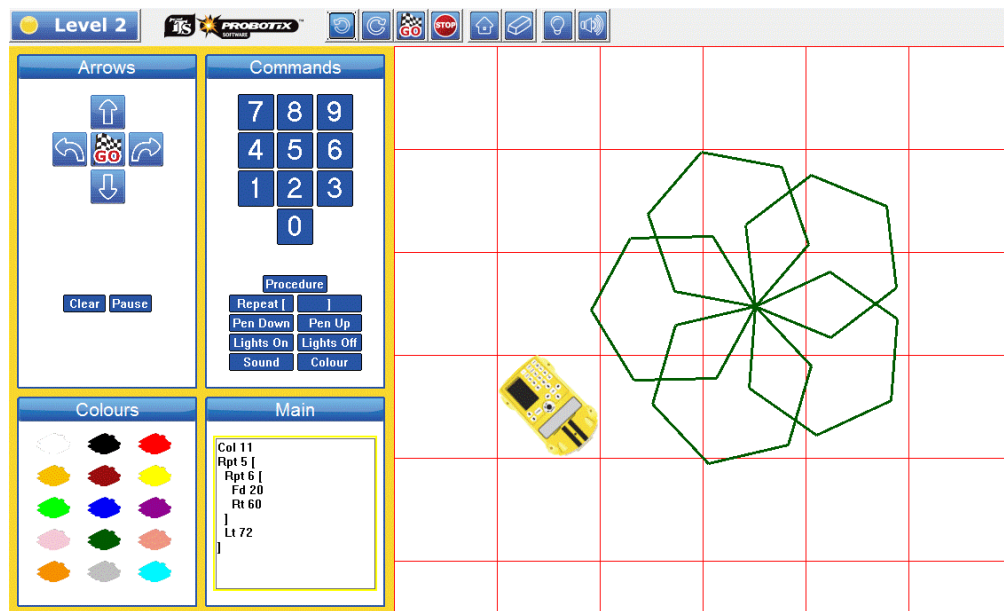
- Objetivo Principal

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

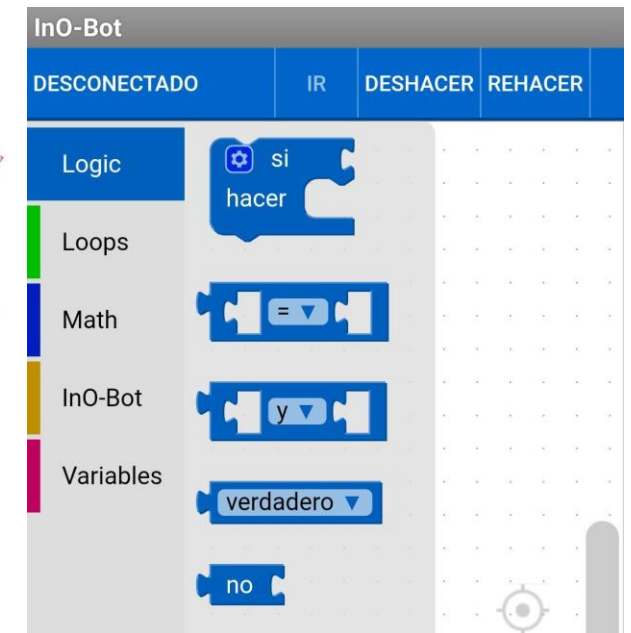
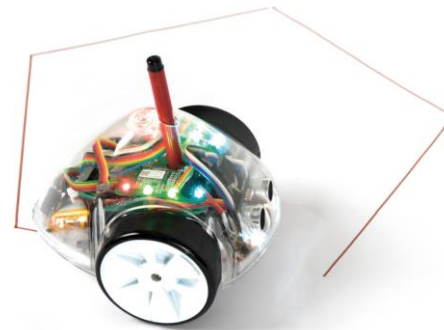


# Contexto. Antecedentes

## Pro-Bot

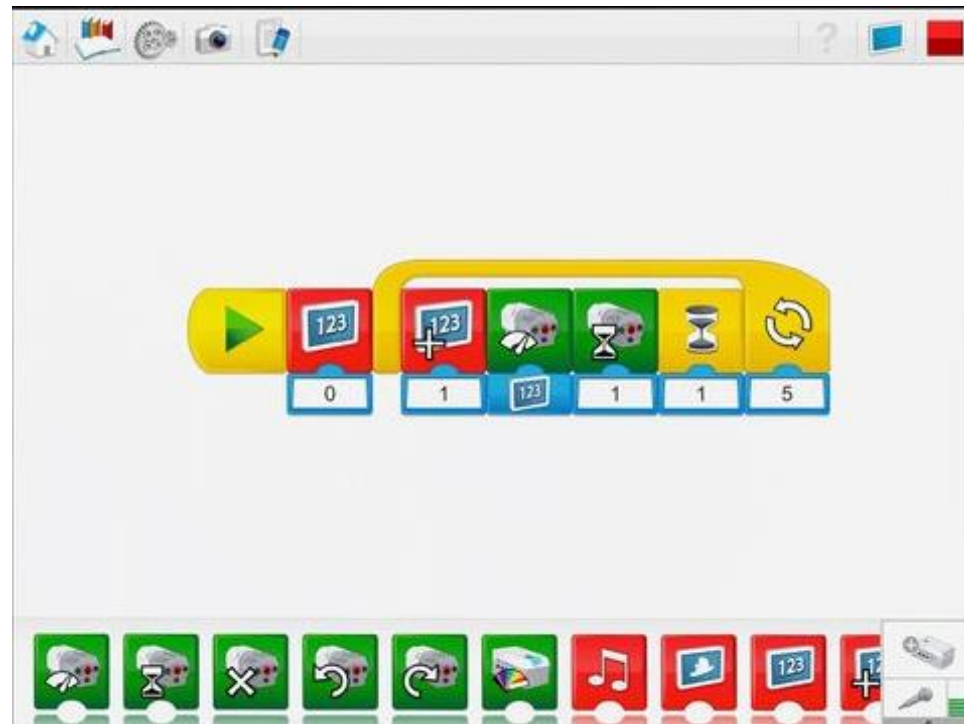


## Ino-Bot



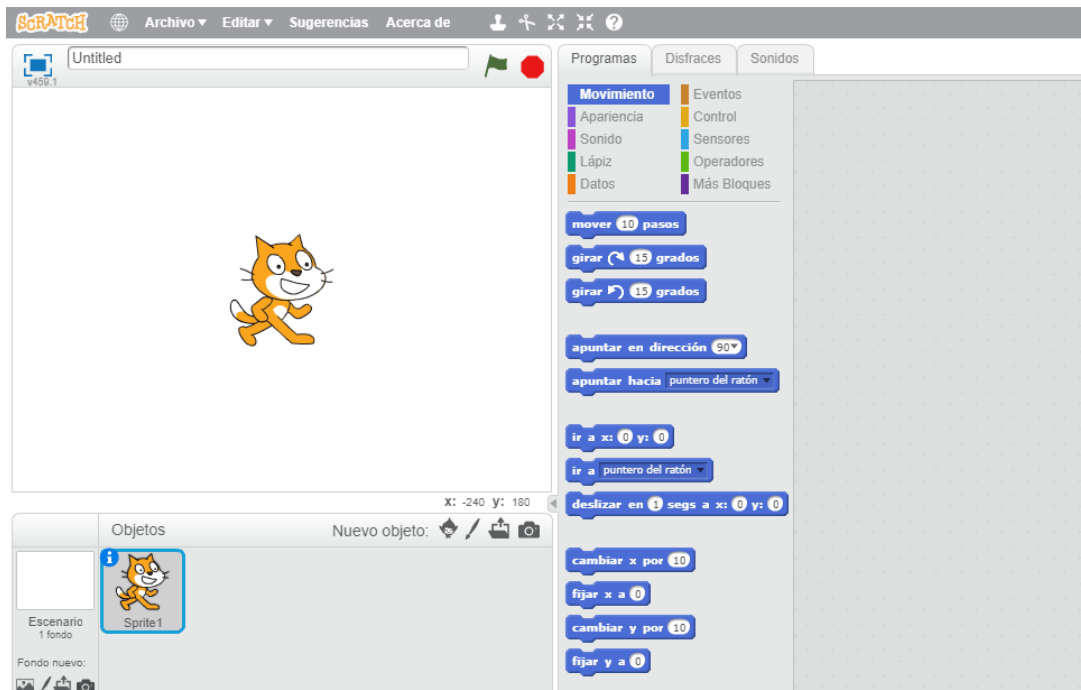
# Contexto. Antecedentes

## LEGO MINDSTORMS Education

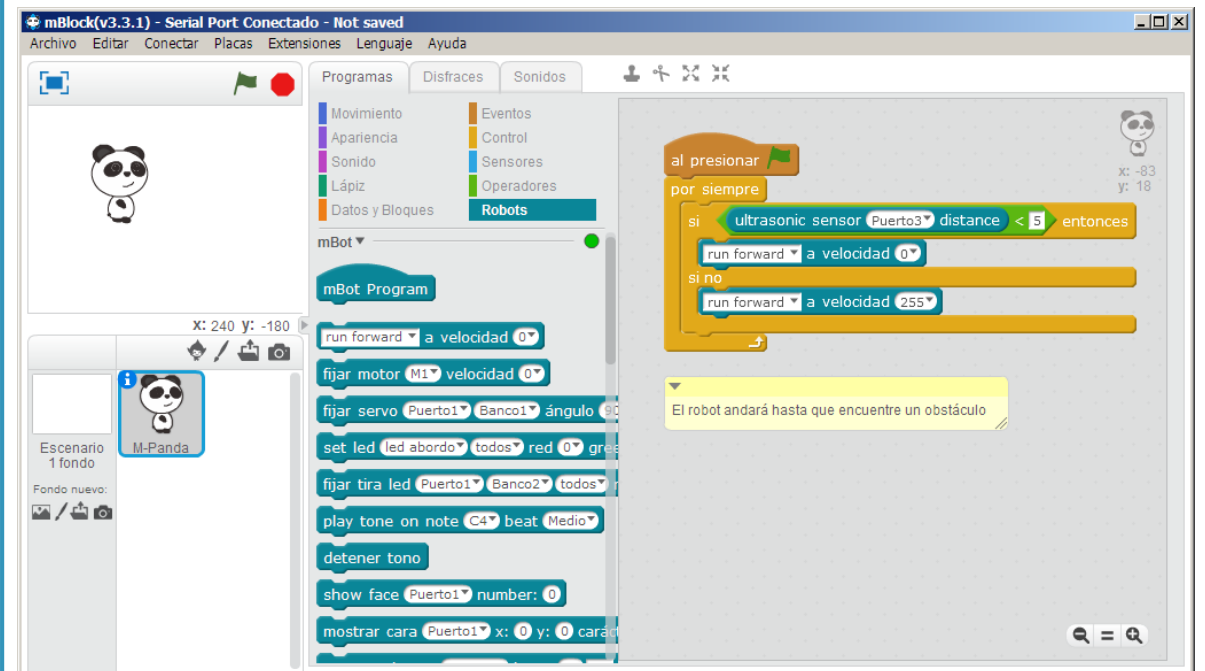


# Contexto. Lenguaje de Programación Visual

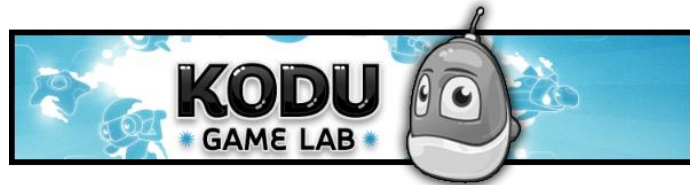
## SCRATCH



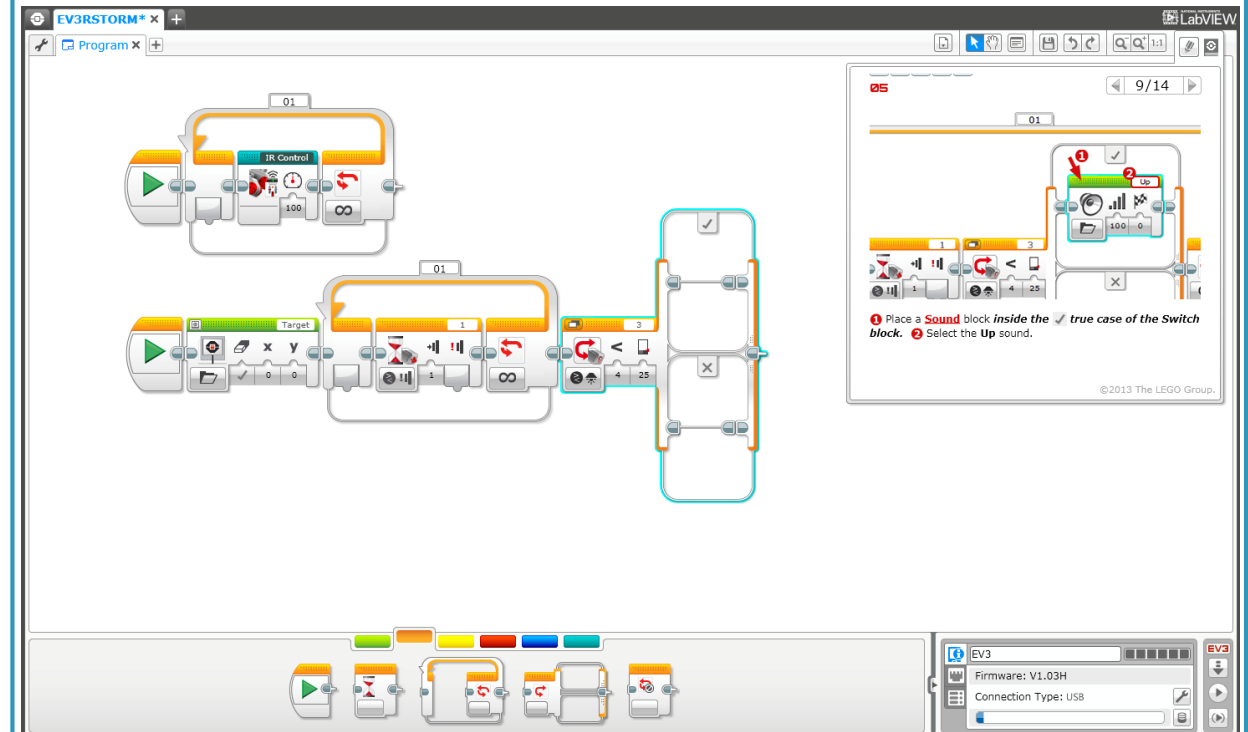
## mBlock



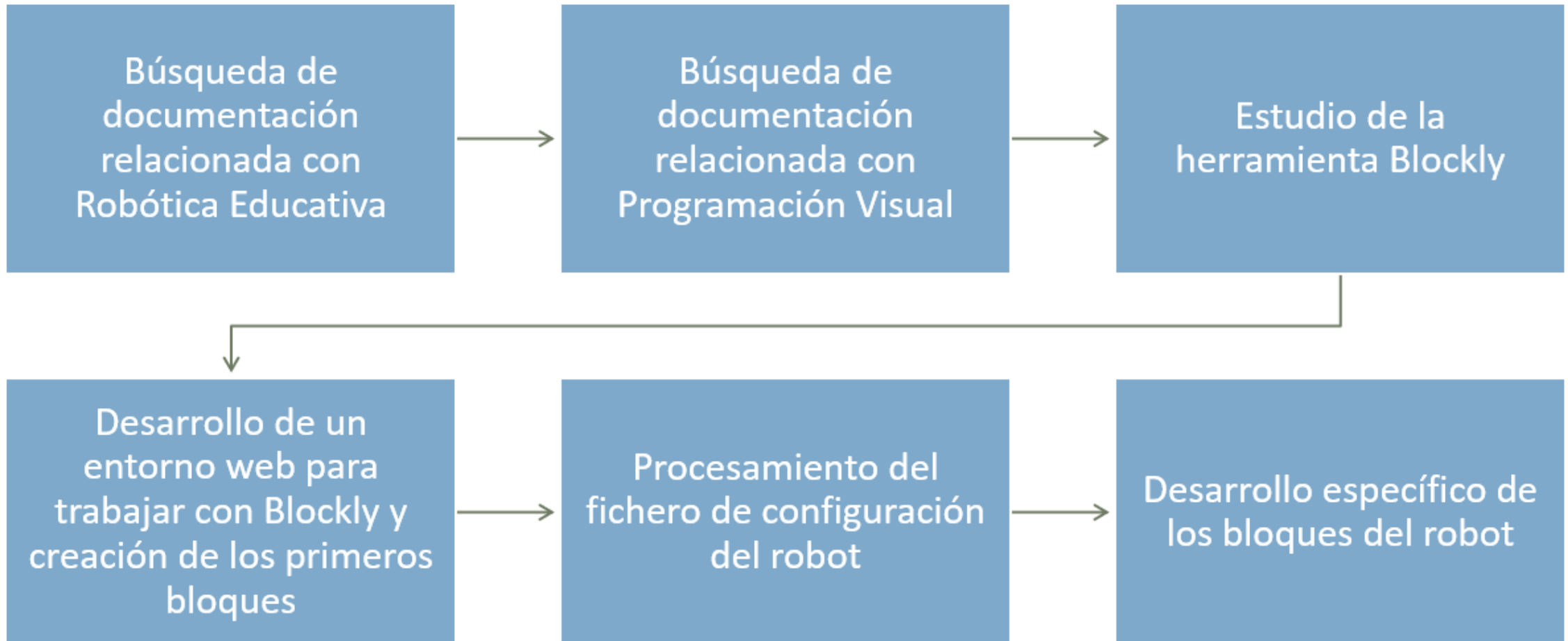
# Contexto. Lenguaje de Programación Visual



## APP EV3 PROGRAMMER

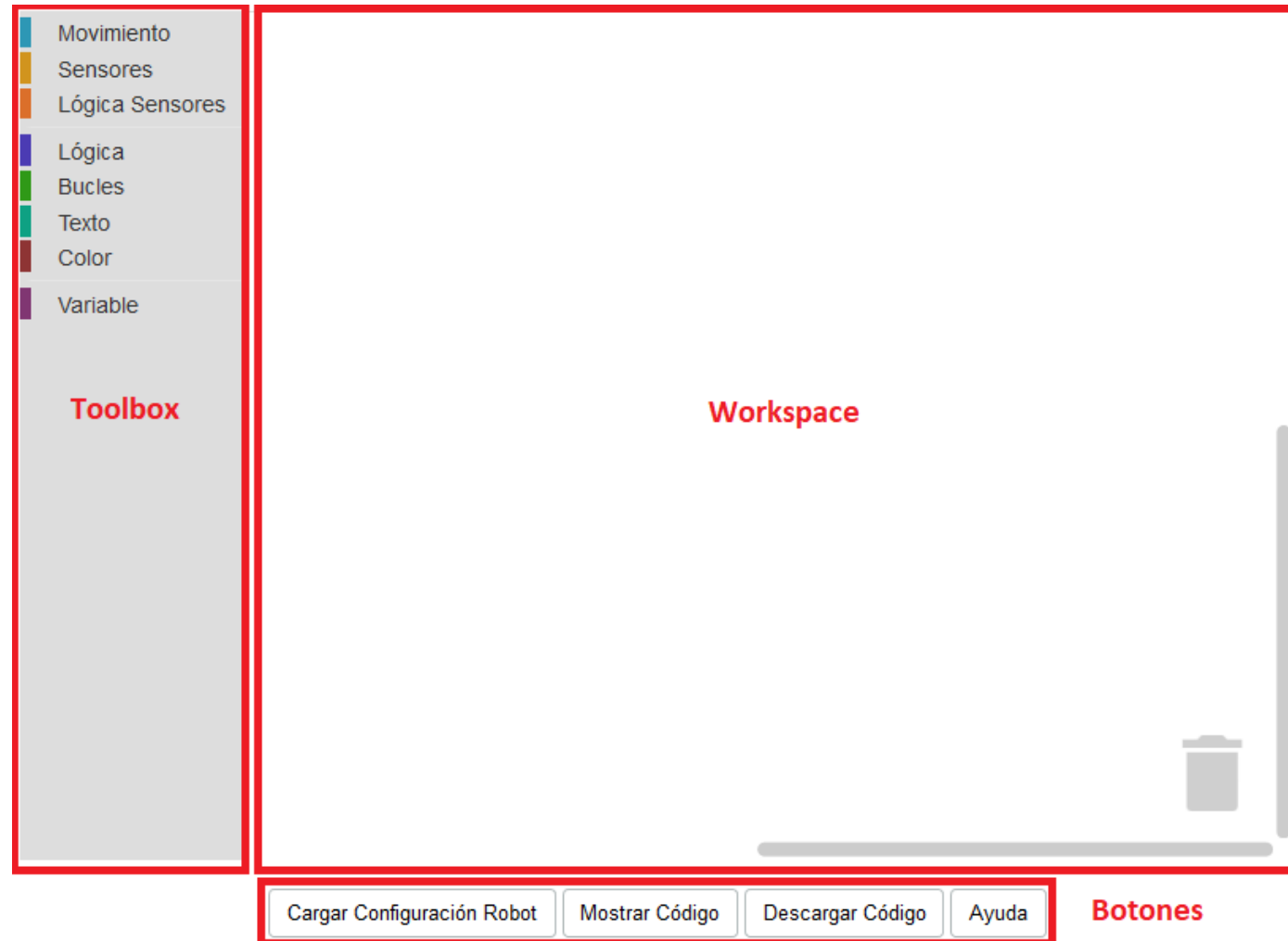


# Fases del desarrollo





# Desarrollo del proyecto



# Desarrollo del proyecto. Toolbox

The image displays the LEGO Mindstorms software toolbox, organized into several categories:

- Movimiento** (Movement):
  - Avanzar 5 metros por 1 seg
  - Retroceder 5 metros por 1 seg
  - Girar 90° izquierda
  - Detener movimiento
- Sensores** (Sensors):
  - Sensor Láser nº 1
  - Sensor de Contacto nº 1
  - Sensor de Ultrasonido nº 1
  - Sensor Infrarrojo nº 1
  - Sensor Odómetro nº 1
- Lógica Sensores** (Sensor Logic):
  - si sensor =
  - hacer
  - sino
  - sino si sensor =
  - hacer
- Lógica** (Logic):
- Bucles** (Loops):
- Texto** (Text):
- Color** (Color):
- Variable** (Variable):

# Desarrollo del proyecto. Toolbox

The image displays the Scratch toolbox, organized into categories on the left and showing various code blocks in the main area.

**Categories (Left Sidebar):**

- Movimiento
- Sensores
- Lógica Sensores
- Lógica**
- Bucles
- Texto
- Color
- Variable

**Code Blocks Shown:**

- Lógica:** "si" block with a "hacer" sub-block.
- Lógica:** "y" block.
- Lógica:** "verdadero" block.
- Lógica:** "0" block.
- Lógica:** "número aleatorio entre 1 y 10" block.
- Lógica:** "nulo" block.
- Lógica:** "+" block.
- Bucles:** "repetir veces" block.
- Bucles:** "repetir mientras" block.
- Bucles:** "contar con i desde hasta" block.
- Variable:** "Crear variable..." block.
- Variable:** "establecer var1 a" block.
- Variable:** "añadir 1 a var1" block.
- Variable:** "var1" block.
- Color:** "color aleatorio" block.
- Texto:** "imprimir" block.

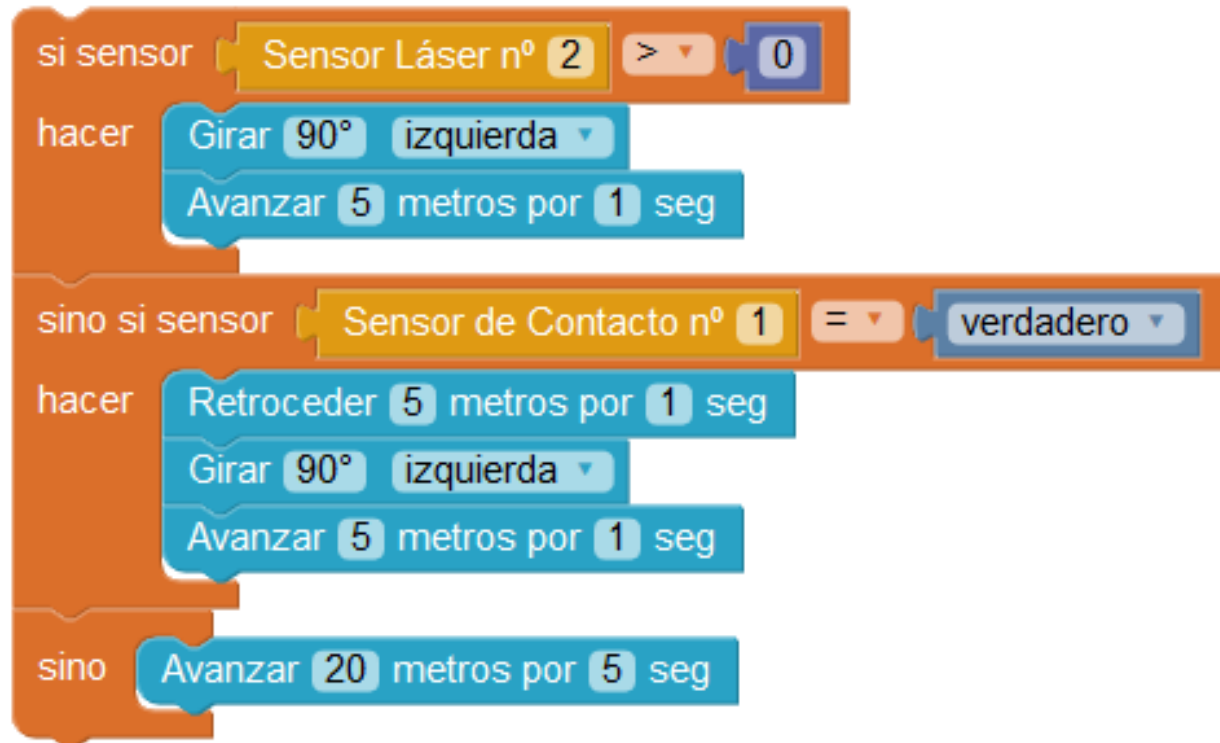
# Desarrollo del proyecto. Botones

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

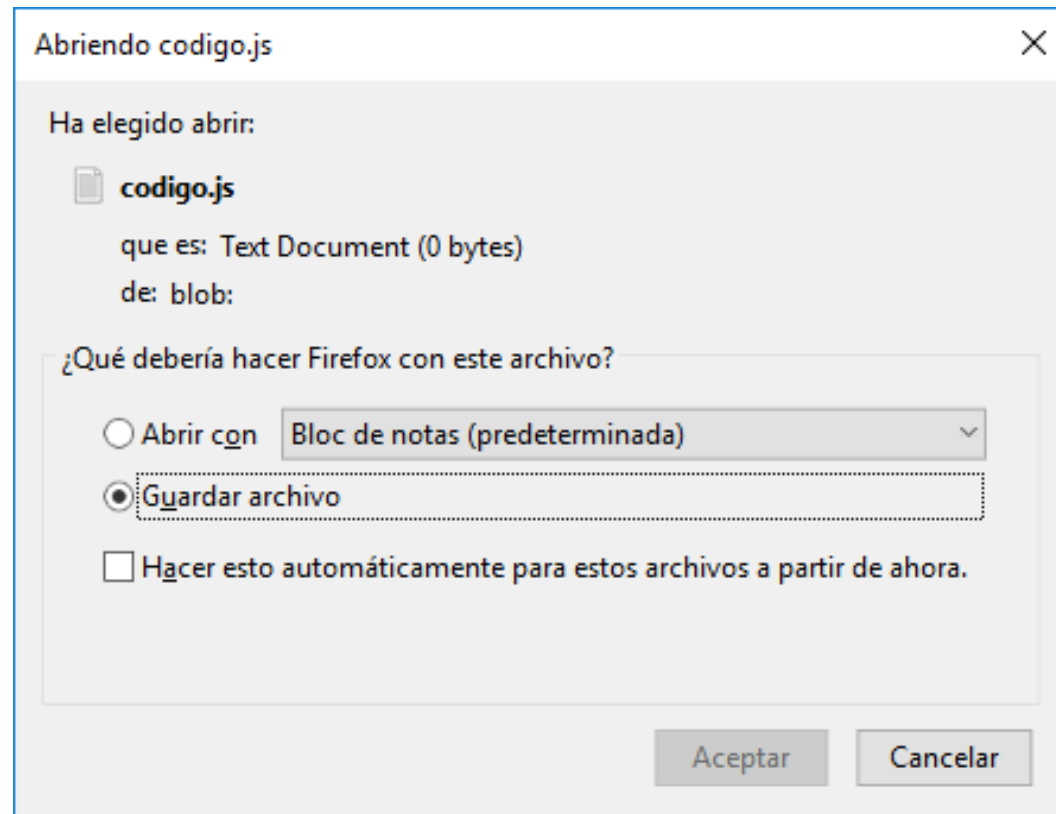
# Desarrollo del proyecto. Botones

Cargar Configuración Robot

Mostrar Código

Descargar Código

Ayuda



# Desarrollo del proyecto. Botones

Cargar Configuración Robot

Mostrar Código

Descargar Código

Ayuda

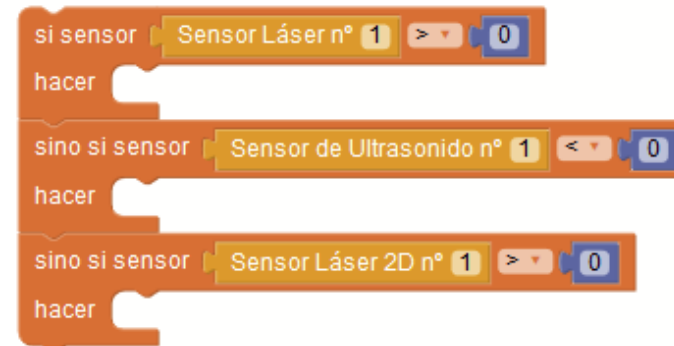
Ejemplo 1: Uso de la aplicación



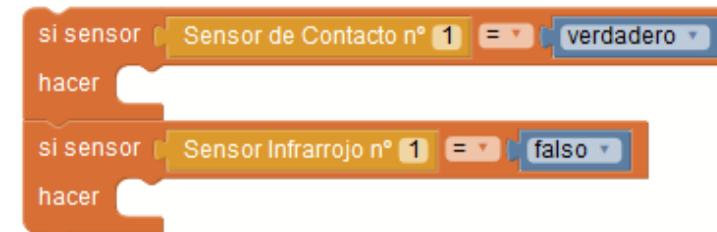
Ejemplo 2: Uso de la aplicación



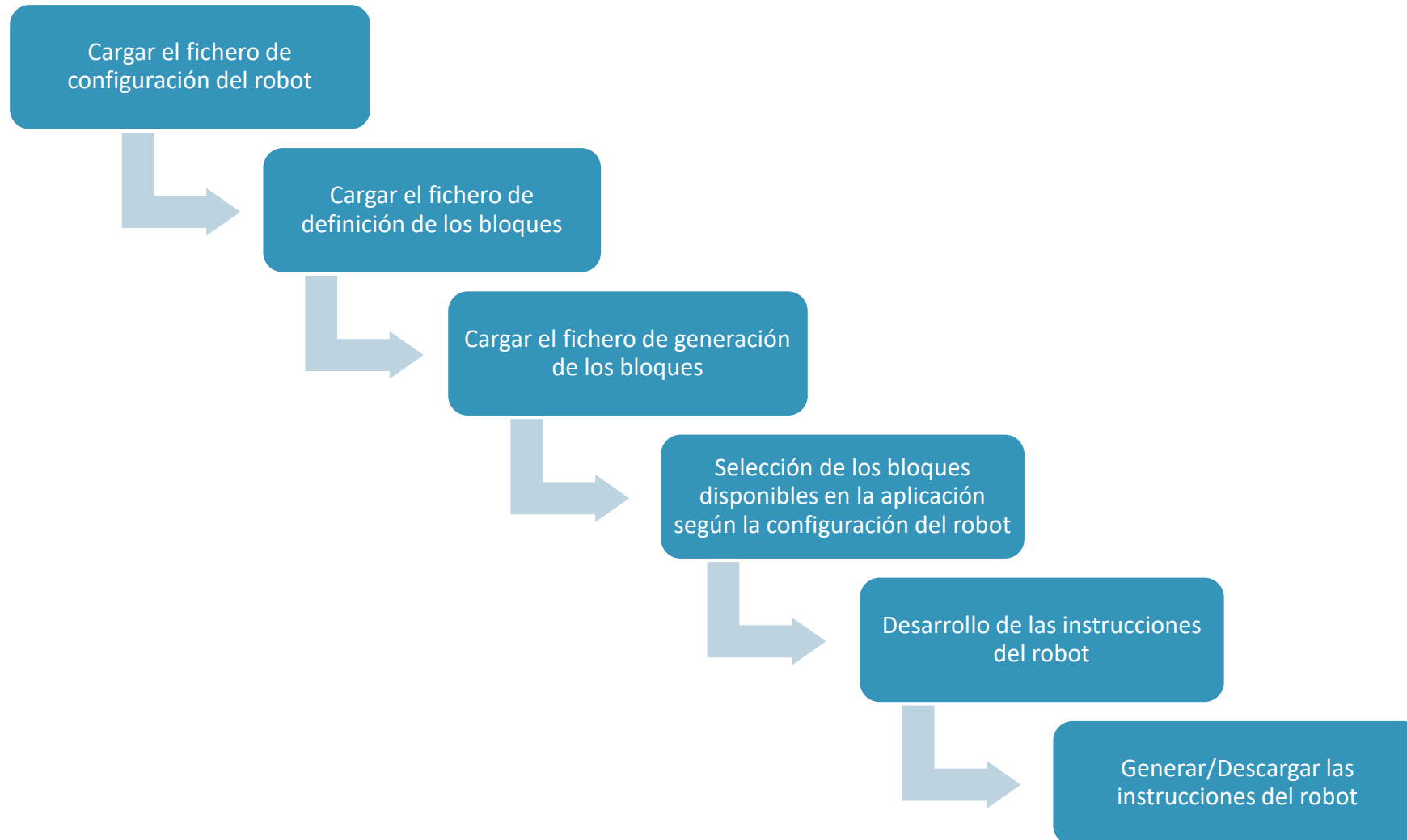
Ejemplo 3: Uso de los sensores Láser, Ultrasonido y Láser 2D



Ejemplo 4: Uso de los sensores de Contacto e Infrarrojo



# Funcionamiento a nivel interno



# 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.



# Caso de uso. Ejemplo de funcionamiento

Enlace a la aplicación desplegada en Heroku:

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

# Conclusiones y trabajos futuros

# Bibliografía