

Manual de operación



por

Eduardo Pelaez Morales

Manuel Fernando Guzmán Muñoz

ÍNDICE

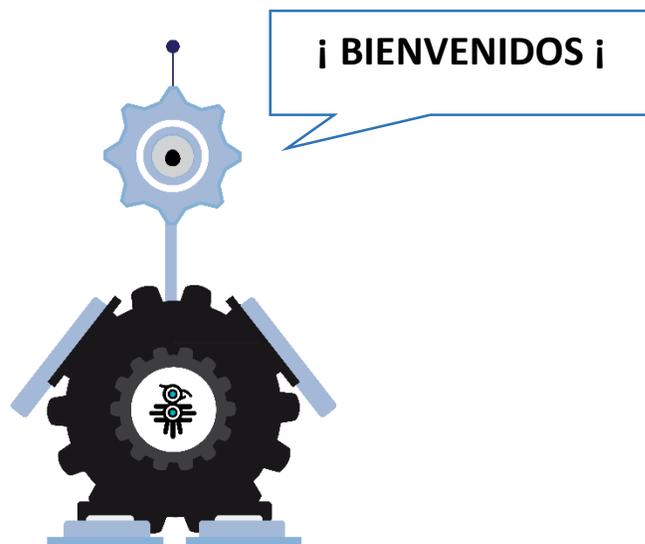
1	Introducción.....	3
2	Primeros pasos.....	3
3	Programación y categorías de los bloques.....	7
	3.1 Mi primer programa.....	7
	3.2 Mi segundo programa.....	8
	3.3 Los bloques SAY y GIRAR.....	9
	3.4 El bucle.....	9
4	Sensores.....	11
	4.1 El sensor ultrasónico.....	11
	4.2 El sensor de color.....	12
5	El primer reto.....	13
6	Uso de dos sensores.....	16
7	Seguidor de línea.....	17
	7.1 Seguidor de línea con 1 sensor.....	17
	7.2 Seguidor de línea con 2 sensores.....	18
8	Sumobot.....	19
	8.1 Programando un Sumobot.....	19
9	Generando escenarios.....	21

1. INTRODUCCIÓN

Open Roberta es una plataforma en internet, 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.

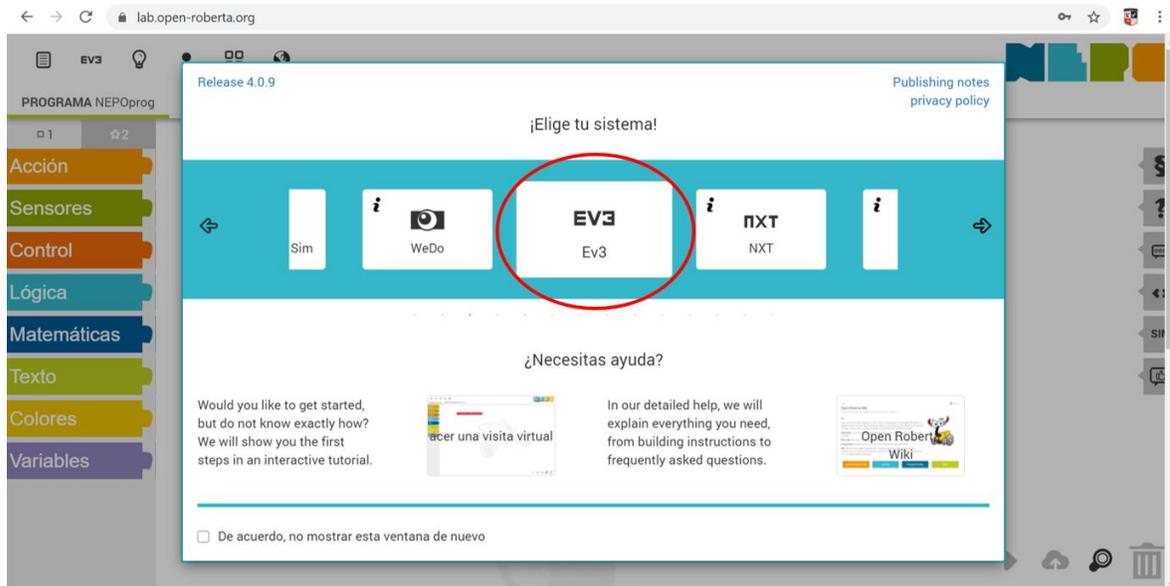
En esta guía describiremos los pasos para programar y mostraremos algunos ejemplos de códigos de programación de robots, haciendo semejanza a la programación del robot LEGO MINDSTORMS.



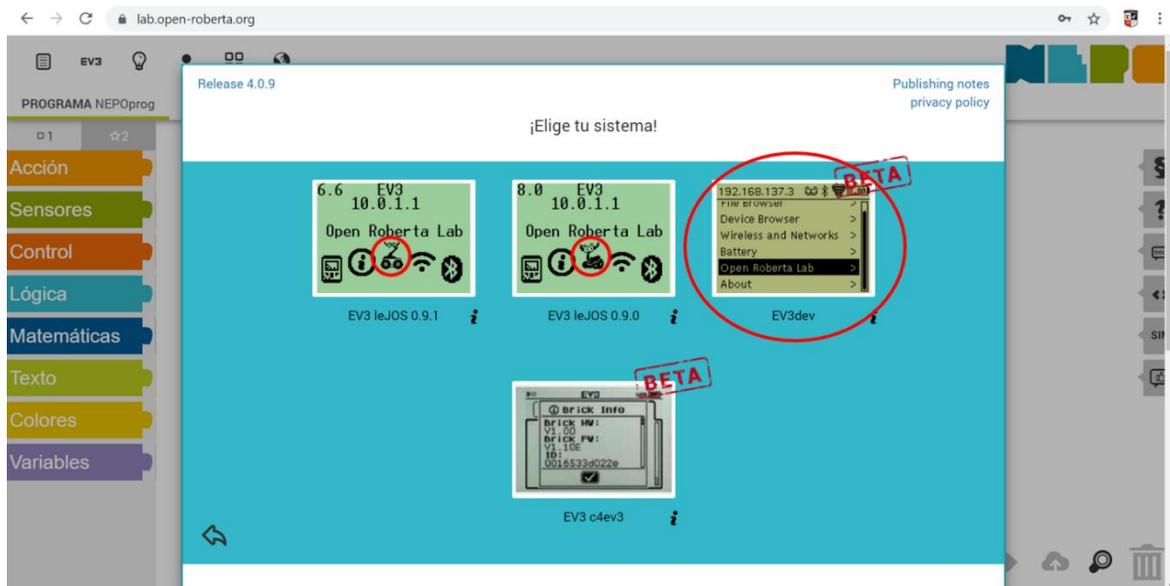
2. PRIMEROS PASOS

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: **lab.open-roberta.org**

Una vez ingresado, aparecerá una imagen como la que se muestra a continuación, en la cual seleccionaremos la opción de EV3.



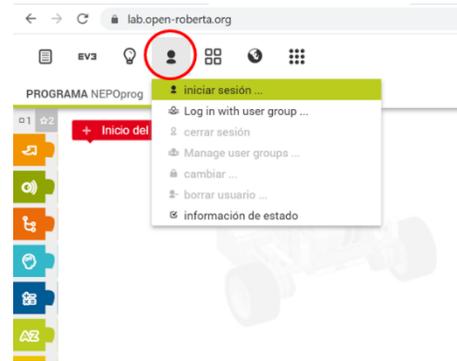
Posteriormente, seleccionaremos la opción EV3dev, como se muestra en la siguiente imagen:



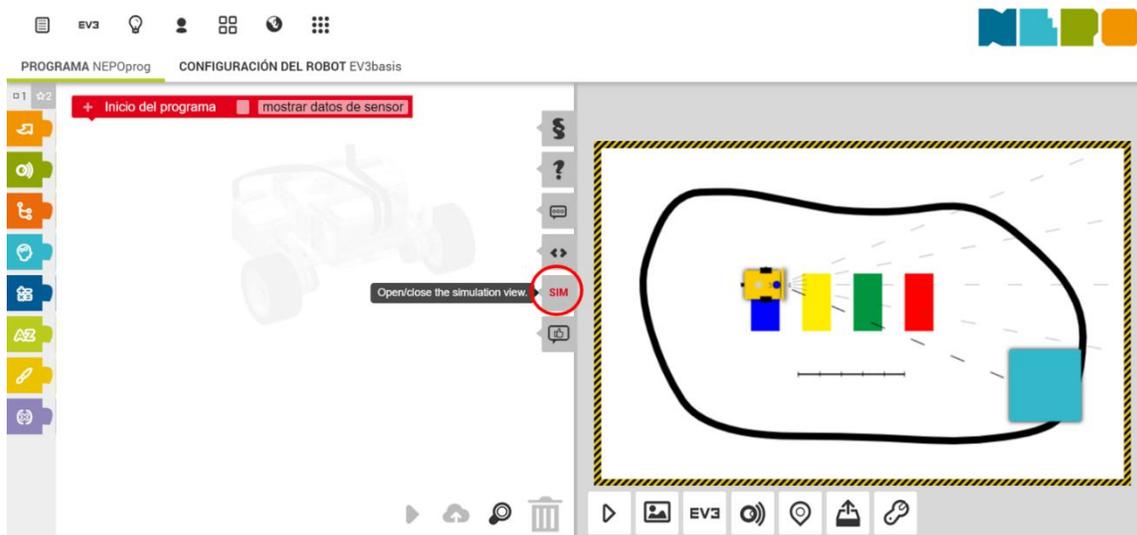
Ya que elegimos la opción EV3dev, nos enviará a la ventana donde comenzaremos a programar.

Para poder hacer uso de todas las herramientas de Open Roberta, es necesario en tu primer acceso **crear una cuenta**, únicamente se llena el formulario y con ello con ello podrás guardar, importar y exportar los códigos de programación que realices en la plataforma.

Ya con tu cuenta creada, podrás **iniciar sesión** con el usuario y contraseña que elegiste, tal como se muestra a continuación:



Para iniciar a trabajar debemos dar clic en el botón SIM ubicado a la derecha de tu pantalla y con ello se abrirá la ventana de los escenarios.



3. PROGRAMACIÓN Y CATEGORÍAS DE LOS BLOQUES

NEPO es un metalenguaje de programación de código abierto gratuito que pueden utilizar estudiantes, académicos, profesores y otras personas interesadas dentro del Open Roberta Lab. NEPO se traduce en New Easy Programming Online (o simplemente OPEN al revés). NEPO es el nombre del lenguaje de programación gráfica y su capa de conexión de hardware acoplada.

NEPO utiliza la biblioteca Blockly disponible gratuitamente. Además, dentro de NEPO hay funcionalidades y mejoras adicionales que se han adaptado para Open Roberta. El paradigma de programación de NEPO está inspirado en Scratch, que fue desarrollado por el Instituto de Tecnología de Massachusetts. Un bloque NEPO siempre representa y encapsula una determinada funcionalidad de robot. Un conjunto de funciones de bloques se puede reconocer fácilmente a través de la categoría de bloque asociada, por ejemplo, »sensores«. La programación con NEPO sigue un principio simple. Los bloques están interconectados y serán ejecutados por el robot según su orden. Este principio se denomina "operación secuencial".

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.

BOTONES de CONTROL

Son los ubicados en el área inferior de la pantalla donde el robot realiza sus misiones

	Inicia la simulación del robot en base al código que le programaste		Reinicia el escenario y coloca al robot en su posición original
	Te permite cambiar de escenarios		Permite agregar escenarios propios
EVE	abre o cierra la vista del robot		Entra en el modo de depuración manual
	Abre o cierra el panel de datos de los sensores		

3.1 MI PRIMER PROGRAMA – Mover el robot hacia adelante

Es momento de hacer nuestro primer programa. Los bloques principales para hacer mover al robot son:

- Mover hacia adelante
- Mover hacia atrás
- Parar

En nuestro primer programa, el robot se desplazará hacia adelante y posteriormente esperará por 1 segundo.

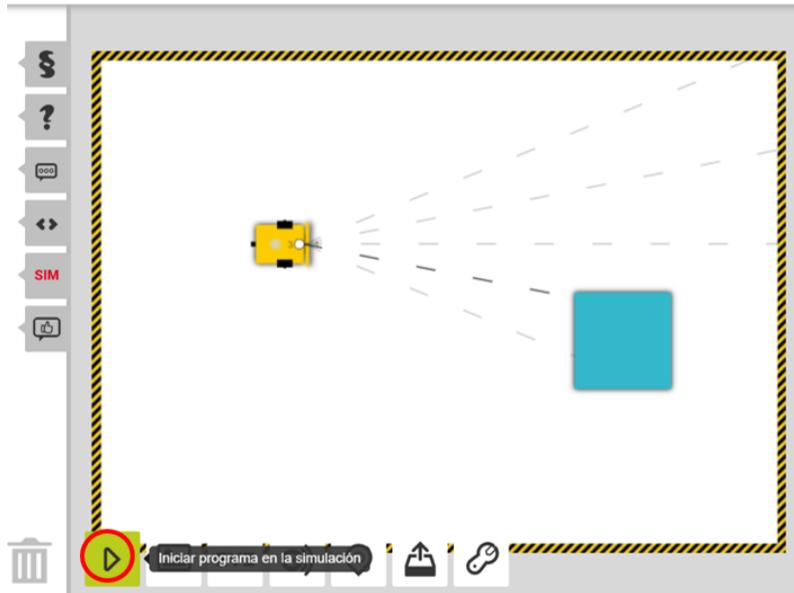
La programación es la siguiente:



El código anterior cumple con el siguiente algoritmo:

- 1.- Mover el robot hacia adelante, velocidad de 100, distancia de 40cms.
- 2.- Esperar 1000 milisegundos (1000 milisegundos = 1 segundo)

Para probar el robot, podemos utilizar la escena básica, que únicamente consiste en el robot sobre un fondo blanco y posteriormente debemos dar clic en el botón **inicia** como se muestra en la siguiente imagen:



3.2 MI SEGUNDO PROGRAMA – Avanzar y retroceder

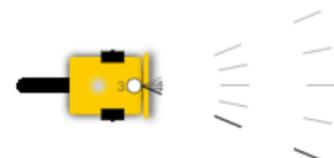
Debido a que en Open Roberta se programa de forma secuencial, vamos a ir conectando los bloques de *avanzar*, *esperar*, *retroceder* y nuevamente un bloque de *esperar* al final.



El código anterior cumple con el siguiente algoritmo:

- 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)

Al realizar el robot los movimientos, es normal que vaya dejando una línea que muestra la trayectoria de este.



3.3 LOS BLOQUES SAY (decir) Y GIRAR

Vamos ahora a incrementar los movimientos de nuestro robot con dos bloques nuevos de programación. El bloque SAY se utiliza para que el robot emita una voz, la cual viene predeterminada por la plataforma. Más adelante utilizaremos este bloque para que el robot nos avise cuando detecta algún color, objeto, o cuando haya completado una misión, etc.

El bloque GIRAR se utiliza para poder girar el robot un determinado número de grados. En el siguiente programa de ejemplo haremos que el robot se mueva hacia adelante, gire 180°, diga: 180 GRADOS y que al final espere por 1 segundo.



La trayectoria que el robot deja después de haber realizado el código es el siguiente:



3.4 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, podemos utilizar el bucle de forma INFINITA o podemos asignarle que se repite un determinado número de veces.

A continuación, mostramos un código que hace uso del bucle, donde el robot cumplirá con el siguiente algoritmo por **5 veces**:

- 1.- El robot se moverá hacia enfrente
- 2.- Dirá UNO
- 3.- Esperara 1 segundo
- 4.- Se moverá hacia atrás
- 5.- Dirá CERO
- 6.- Esperará 1 segundo

Los bloques de programación quedarían de la siguiente forma:

The screenshot displays a programming interface. On the left, a code block is visible, starting with a red 'Inicio del programa' block and a 'mostrar datos de sensor' block. Below these is a 'repetir 5 veces' loop containing a 'hacer' block with the following steps: 'mover hacia delante' (velocidad 100, distancia 40), 'say UNO', 'esperar 1000', 'mover hacia atrás' (velocidad 100, distancia 40), 'say CERO', and 'esperar 1000'. On the right, a simulation shows a yellow robot in a corner with a black line drawn on the floor.

Ahora utilizaremos el BUCLE para hacer que el robot dibuje un cuadrado, utilizaremos la instrucción **repetir** 4 veces incluyendo lo siguiente:

- 1.- Mover hacia adelante con velocidad de 30, distancia de 30cms.
- 2.- Girar 90 grados hacia la derecha a velocidad de 30
- 3.- Lo anterior repetirlo 4 veces

The screenshot displays a programming interface. On the left, a code block is visible, starting with a red 'Inicio del programa' block and a 'mostrar datos de sensor' block. Below these is a 'repetir 4 veces' loop containing a 'hacer' block with the following steps: 'mover hacia delante' (velocidad 30, distancia 30) and 'gira derecho' (velocidad 30, grado 90). On the right, a simulation shows a yellow robot in a corner with a black square drawn on the floor.

Del lado izquierdo podemos ver los bloques de programación y del lado derecho el cuadrado generado por la trayectoria del robot después de haber hecho los movimientos.

4. SENSORES

Nuestro robot cuenta con sensores que le permiten saber cuándo está cerca de un objeto o color, estos sensores, así como los motores que usa nuestro robot se configuran desde la opción CONFIGURACION DEL ROBOT.



Para utilizar los sensores al programar, utilizamos el bloque **esperar hasta**, y posteriormente elegimos el sensor dependiendo el propósito del desafío o la tarea a realizar.

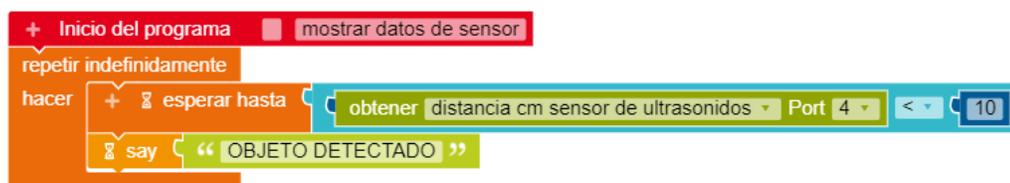
4.1 EL SENSOR ULTRASÓNICO (detecta objetos)

El ejemplo más simple para utilizar el sensor ultrasónico es cuando avanza el robot y al detectar un objeto a cierta distancia, le decimos que se detenga.

Antes de realizar el ejercicio revisa en la **configuración del robot** que el "Sensor 4" tenga seleccionado al "sensor de ultrasonidos" y estés en la escena  que tiene un cuadro azul. Coloca tu robot frente al cuadro azul como lo marca la imagen.



Otro ejemplo del sensor ultrasónico, es cuando ejecutas tu código y manualmente arrastras al recuadro azul hacia tu robot, al estar a una distancia menor de 10cms., Se emitirá un sonido que dice "OBJETO DETECTADO".

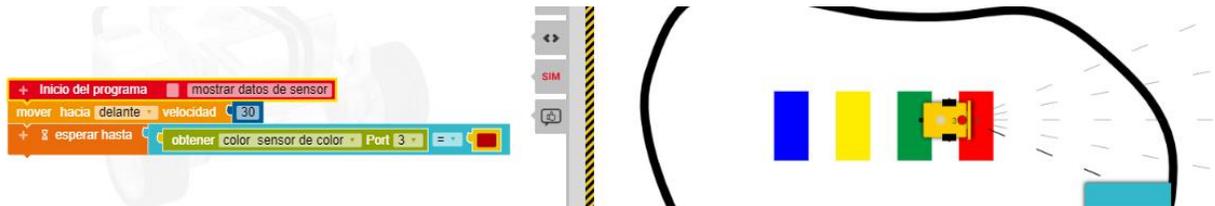


4.2 EL SENSOR DE COLOR

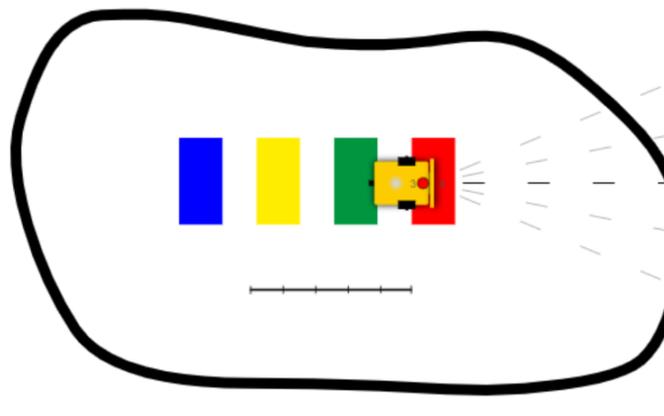
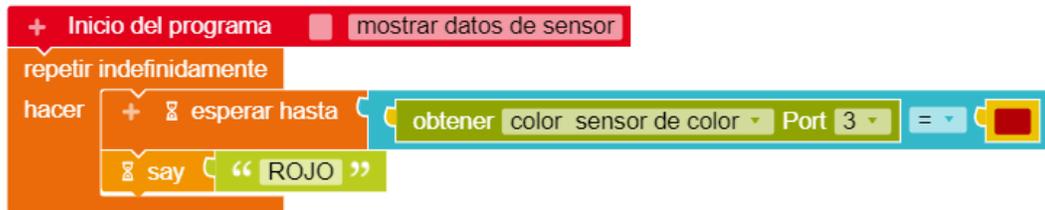
El sensor de color detecta 7 diferentes colores: rojo, azul, verde, amarillo, café, negro y blanco.

Ahora con el **sensor de color** haremos que el robot avance, y cuando detecte el color seleccionado, se detendrá.

Antes de realizar el ejercicio revisa en la **configuración del robot** que el “Sensor 3” tenga seleccionado al “sensor de color” y estés en la escena  que tiene las barras con los 4 colores. Coloca tu robot antes del cuadro azul y dale Play  a tu programa.



Otro ejemplo básico para utilizar el sensor de color es que cuando el robot detecte algún color, este emita algún sonido de acuerdo al color detectado, por ejemplo:



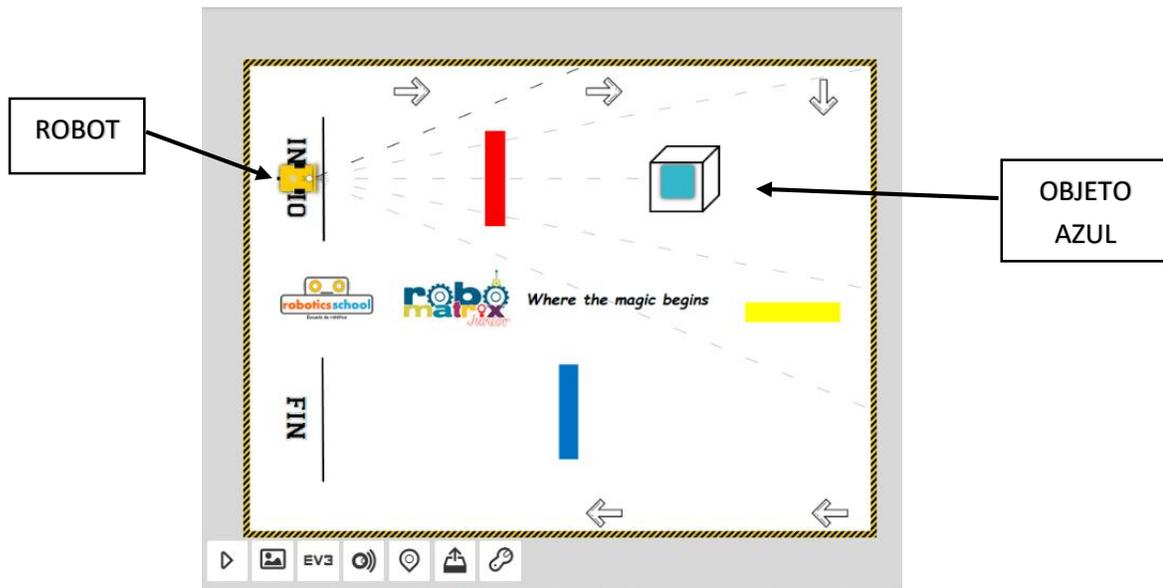
5. PRIMER “RETO”

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.

Antes de iniciar:

- 1) Descarga el mundo ([aquí](#)) a tu computadora y después cárgalo en Open Roberta con la opción 
- 2) Arrastra el Cuadro Azul (objeto) que aparece en la parte inferior derecha del mundo y colócalo dentro del cubo.
- 3) Coloca a tu robot sobre las letras de INICIO del mundo.
- 4) Confirma en **Configuración del Robot** que el sensor 3 = “sensor de color” y el sensor 4 = “sensor de ultrasonidos”

Tu mundo deberá verse así:



El algoritmo para completar el primer reto es 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 10cms. 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

El código propuesto es el siguiente:

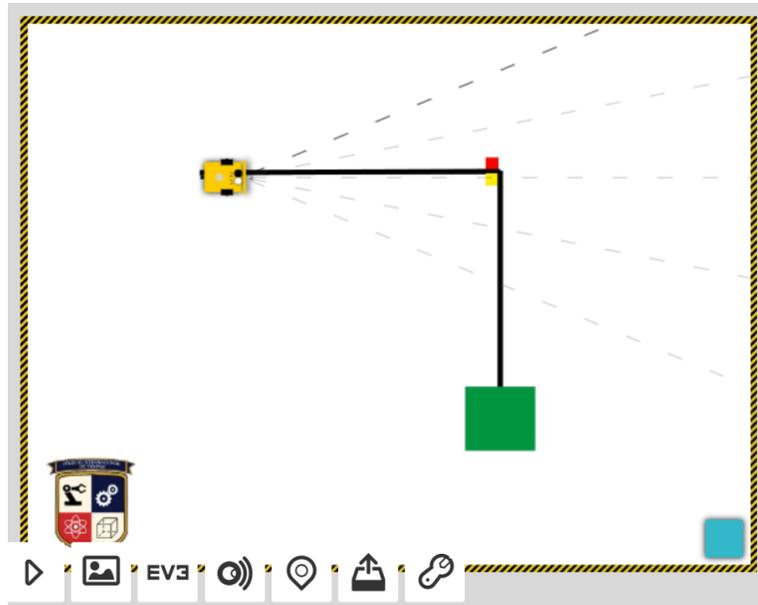
```
+ Inicio del programa  mostrar datos de sensor
mover hacia delante  velocidad 50
+ esperar hasta  obtener color sensor de color Port 3 = [Red]
para
  esperar 1000
  mover hacia delante  velocidad 50
  + esperar hasta  obtener distancia cm sensor de ultrasonidos Port 4 <= 10
  para
    gira derecho  velocidad 50
                    grado 90
    mover hacia delante  velocidad 50
                    distancia en cm 30
    gira izquierdo  velocidad 50
                    grado 90
    mover hacia delante  velocidad 50
                    distancia en cm 80
    gira derecho  velocidad 50
                    grado 90
  mover hacia delante  velocidad 90
  + esperar hasta  obtener color sensor de color Port 3 = [Yellow]
  para
    esperar 1000
    mover hacia delante  velocidad 50
                    distancia en cm 60
    gira derecho  velocidad 50
                    grado 90
  mover hacia delante  velocidad 50
  + esperar hasta  obtener color sensor de color Port 3 = [Blue]
  para
    esperar 1000
    mover hacia delante  velocidad 50
                    distancia en cm 140
```

Prueba tu código y juega cambiando algunos comandos en la programación para ver el comportamiento de tu robot virtual. Por ejemplo: Programa que esquive el objeto por la parte superior y no la inferior.

6. USO DE DOS SENSORES

Para resolver un reto con múltiples colores

Tenemos un escenario donde se necesitan utilizar dos sensores: uno para seguidor de línea y otro sensor para detectar color y tomar decisiones.



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.- Hace nuevamente el seguidor de línea con el sensor 2 hasta que detecte el color verde con el sensor 3, ahí habrá completado su recorrido

Antes de iniciar:

- 1) Descarga el mundo ([aquí](#)) a tu computadora y después cárgalo en Open Roberta con la opción 
- 2) Arrastra el Cuadro Azul (objeto) que aparece en la parte inferior derecha del mundo y colócalo dentro del cubo.
- 3) Coloca a tu robot con el sensor 2 sobre la línea negra.
- 4) Confirma en **Configuración del Robot** que el sensor 2 y sensor 3 tengan activados el "sensor de color"

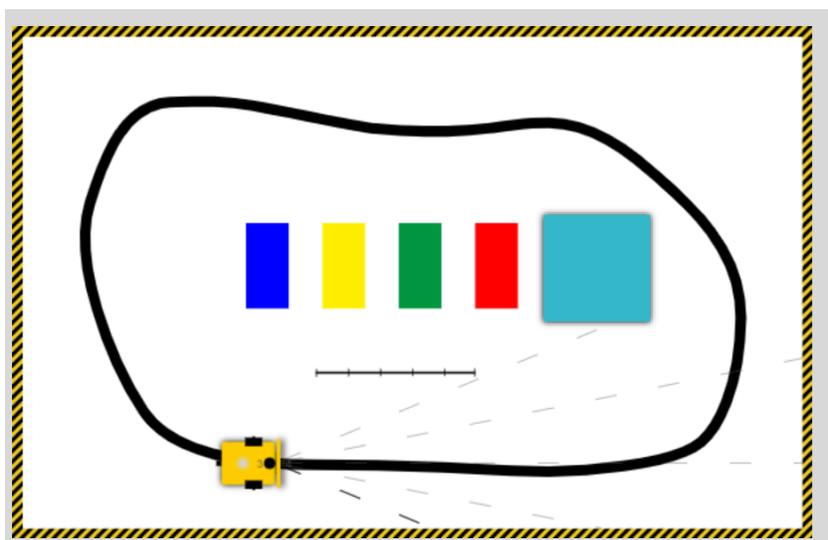
El código propuesto es el siguiente:

```
+ Inicio del programa  mostrar datos de sensor
say "iniciamos"
repetir indefinidamente
  hacer
    + - esperar hasta obtener color sensor de color Port 2 = [ ]
    hacer
      steer hacia delante velocidad izquierdo 40
      velocidad derecho 10
    o esperar a obtener color sensor de color Port 2 = [ ]
    hacer
      steer hacia delante velocidad izquierdo 10
      velocidad derecho 40
    o esperar a obtener color sensor de color Port 3 = [ ]
    hacer
      para
        say "Detecte amarillo"
        mover hacia delante velocidad 30
        distancia en cm 10
        gira derecho velocidad 20
        + esperar hasta obtener color sensor de color Port 2 = [ ]
      para
        esperar 500
    o esperar a obtener color sensor de color Port 3 = [ ]
    hacer
      say "Fin"
      para
```

7. 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. A través de una toma de decisiones, el robot va diferenciando entre el color negro y el color blanco, para así hacer un movimiento de zig-zag e ir avanzando.

7.1 SEGUIDOR DE LÍNEA CON 1 SENSOR DE COLOR



El seguidor de línea con 1 sensor de color, consiste en un robot con un sensor conectado en el puerto 3 para que cumpla un algoritmo de seguir la línea negra en el menor tiempo posible.

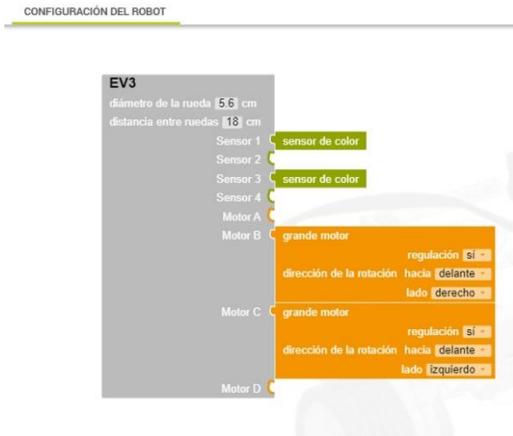
A continuación, se muestra un código propuesto de programación:

```

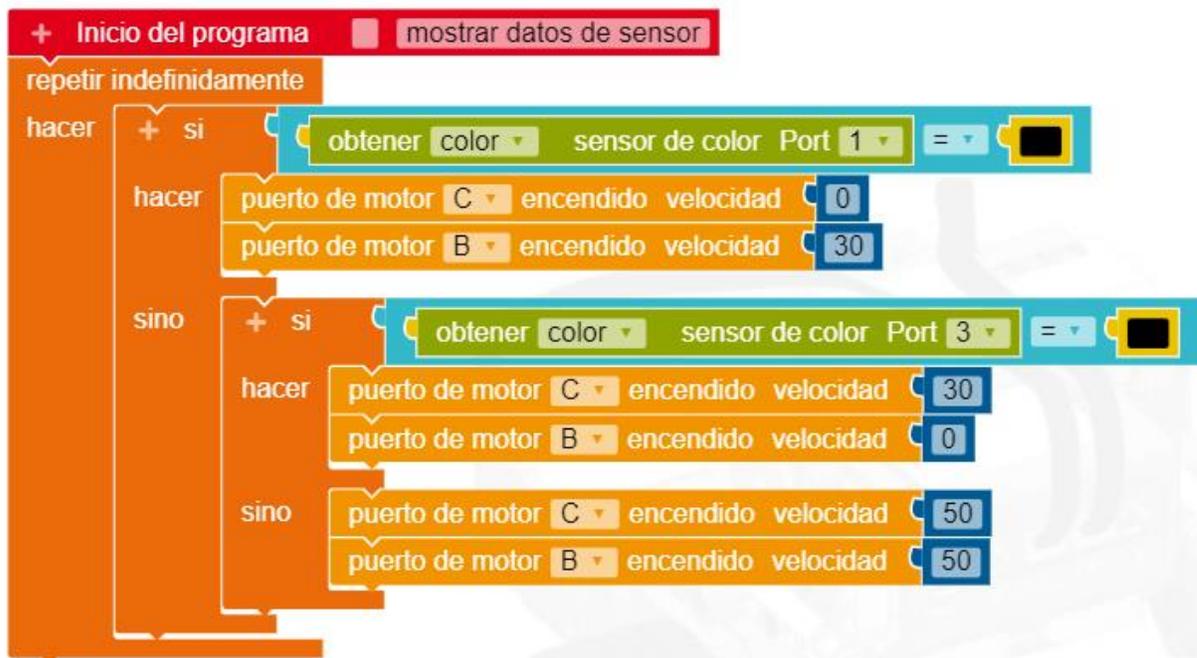
+ Inicio del programa  mostrar datos de sensor
repetir indefinidamente
hacer
+ ⌚ esperar hasta  obtener color sensor de color Port 3 =  [Negro]
steer hacia delante velocidad izquierdo 100
                                velocidad derecho 60
+ ⌚ esperar hasta  obtener color sensor de color Port 3 =  [Blanco]
steer hacia delante velocidad izquierdo 60
                                velocidad derecho 100
    
```

7.2 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. Un sensor va conectado 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, debemos ir a CONFIGURACIÓN DEL ROBOT y confirmar tenemos un panel como el siguiente:



El código propuesto es el siguiente:

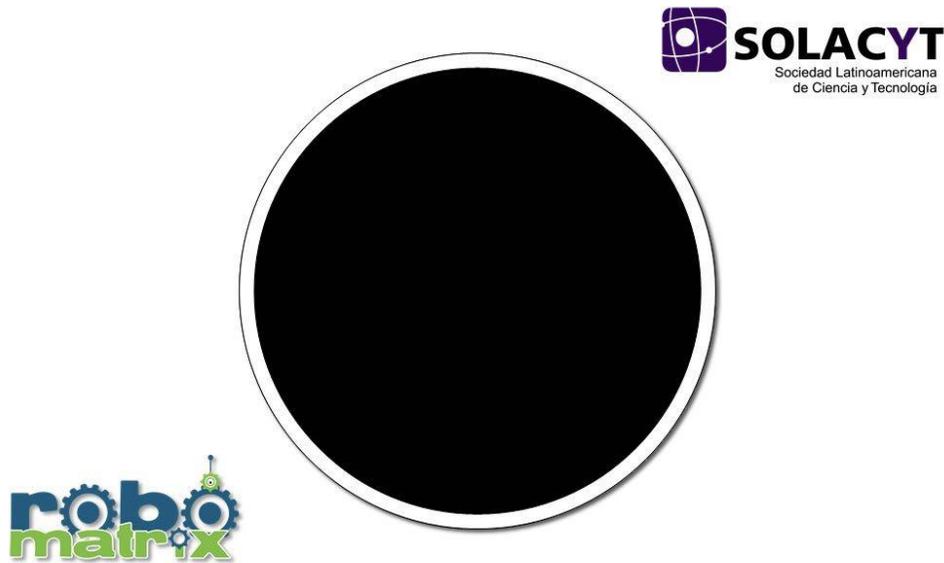


[Ver aquí](#) el video tutorial de este Reto

8. 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:

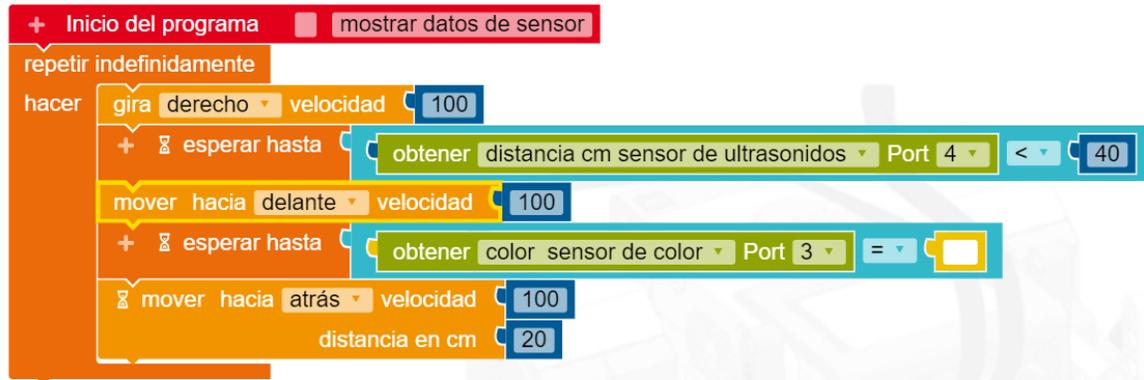
- 1.- El robot debe esperar 5 segundos.
- 2.- El robot debe localizar al oponente, haciendo uso de sus motores ya sea para girar o avanzar.
- 3.- Una vez que el sensor de presencia (puede ser un ultrasónico, infrarrojo, etc.) detecte al oponente, el robot activa sus motores y avanza hasta empujar al oponente.
- 4.- 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.
- 5.- Repetir las acciones anteriores hasta sacar al oponente del dojo.

8.1 PROGRAMANDO EL SUMOBOT

Ahora programaremos un sumobot con un código sencillo pero efectivo, que consiste en los siguientes componentes:

- 1 sensor ultrasónico
- 2 motores
- 1 sensor de color

El programa propuesto para el sumobot es el siguiente:



```
+ Inicio del programa mostrar datos de sensor
repetir indefinidamente
hacer
  gira derecho velocidad 100
  + esperar hasta obtener distancia cm sensor de ultrasonidos Port 4 < 40
  mover hacia delante velocidad 100
  + esperar hasta obtener color sensor de color Port 3 = 
  mover hacia atrás velocidad 100
  distancia en cm 20
```

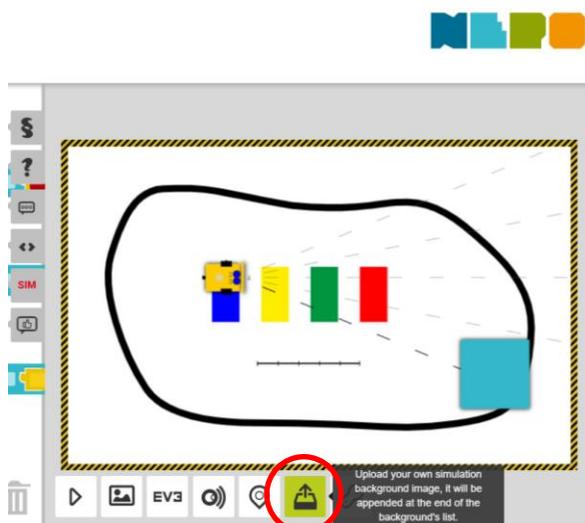
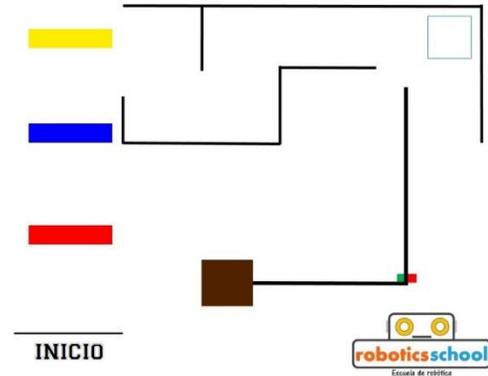
El código anterior cumple con el algoritmo para que nuestro sumobot haga la tarea sencilla de localizar al oponente y sacarlo del área de combate, pero que tu sumobot sea el mejor dependerá de los algoritmos que lores tu implementar para que tu robot tenga un mejor rendimiento en el combate.

9. 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. Pondremos de ejemplo el siguiente, el cual fue creado en power point haciendo uso únicamente de líneas, colores y formas

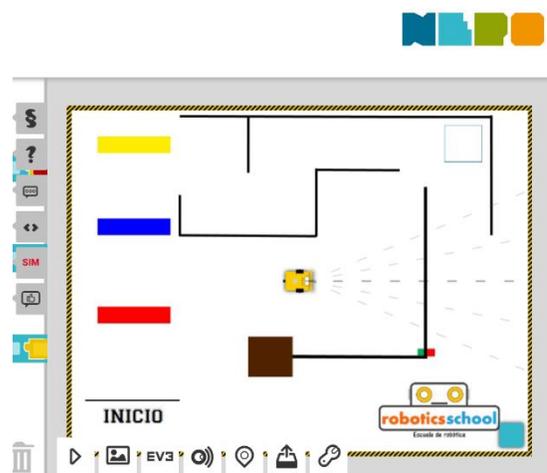


2. Cargamos la imagen a la plataforma

Para cargar la imagen a la plataforma de open Roberta, debemos seleccionar la opción “Upload your own simulation background image”:

Una vez que damos clic, debemos seleccionar el escenario que deseamos poner en Open Roberta. Posteriormente, te quedará algo como lo siguiente:

El cuadro de azul que aparece en open Roberta, es un objeto sólido, el cual puede ser utilizado para ser detectado por el sensor ultrasónico.





Esta pandemia del COVID-19 nos permitió reinventarnos y conocer más recursos digitales para uso de la educación, en la búsqueda de lo mejor, la plataforma de Open Roberta ha sido una gran sorpresa, su facilidad de manejo, el entorno grafico amigable, la capacidad de interactuar con recursos del usuario y lo mejor, una plataforma gratuita al servicio de la educación.

Dr. Manuel Fernando Guzmán Muñoz

La robótica y programación son una de las pocas cosas del mundo en las que te sientas y puedes crear algo totalmente nuevo de la nada. Se han convertido en una herramienta muy fundamental para los estudiantes, desde niveles básicos hasta niveles media superior y superior. En la actualidad, existen diversos softwares para programar, construir y diseñar robots, que junto con el internet y las diferentes plataformas, los estudiantes pueden crear cosas extraordinarias con tan solo una computadora, y lo más importante, hacerlo desde casa y adaptándonos a las nuevas formas de trabajar.

Ing. Eduardo Peláez Morales

Recursos:

<https://lab.open-roberta.org/>

<https://robomatrix.org>