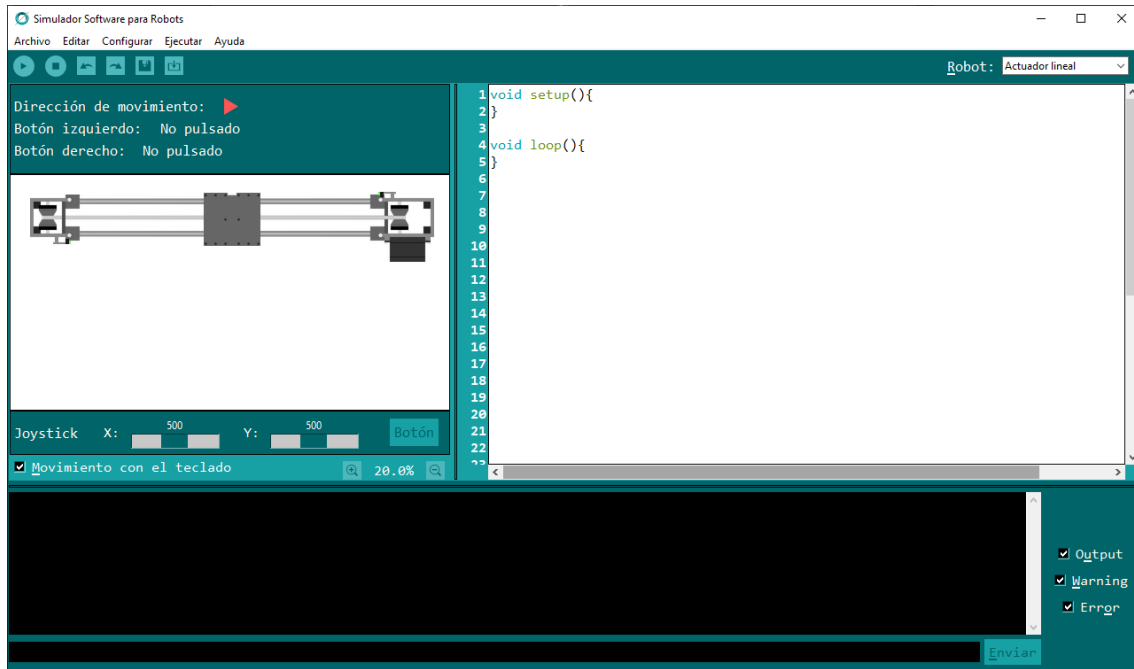


Simulador S4R



Práctica 4 – Simulador (v1.0 septiembre 2022)

Software para robots

Cristian González García

gonzalezcristian@uniovi.es

Simulador desarrollado por Diego Fernández Suárez y dirigido por Cristian González García

Índice

1.	Introducción	2
2.	Instalación	2
3.	Ejecución	4
4.	Manual de usuario	5
	Visión general del sistema	5
	Menú	5
	Barra de herramientas	8
	Representación gráfica de los robots	9
	Editor de código	14
	Consola de entrada/salida	15
	Otras características a tener en cuenta	16

1. Introducción

Este simulador fue desarrollado por el alumno Diego Fernández Suárez de la Universidad de Oviedo para la asignatura de Software para Robots del Grado en ingeniería Informática del Software a través de la realización de su trabajo final de grado (TFG). Este fue dirigido por Cristian González García (gonzalezcristian@uniovi.es).

El siguiente enlace contiene el TFG: <https://digibuo.uniovi.es/dspace/handle/10651/64148>

El manual es un extracto de dicho TFG, pero incluyendo mejoras y cambios realizados.

Actualmente, dicho simulador está en proceso de ampliación para un Proyecto de Innovación Docente en colaboración con los profesores Ana Gómez Pérez de la Universidad Rey Juan Carlos y Domingo Gómez Pérez de la Universidad de Cantabria.

2. Instalación

El sistema deberá instalarse en un ordenador que tenga como sistema operativo Windows. Sería preferible que fuese Windows 10, ya que sobre este se ha probado la correcta ejecución del sistema. No obstante, el sistema debería funcionar correctamente instalándolo en cualquier versión posterior de Windows (Windows 11 en adelante) y en cualquier versión de Windows que cuente con soporte técnico hoy en día.

Ahora, se mostrarán los pasos necesarios para instalar el sistema de forma correcta:

1. Descargar archivo comprimido (.zip) de <https://github.com/diegofs29/simulator-robotic-software/releases/download/v1.0.0/simulator-SR-win.zip>. Este contiene los archivos necesarios para el correcto funcionamiento del ejecutable, además del propio ejecutable (.exe). Al descargarlo, deberíamos tener un archivo cuyo nombre será del tipo “simulador-SR-win”.
2. Descomprimir el archivo (en el directorio que queramos almacenar el programa y sin falta de descomprimir en un nuevo directorio, el archivo comprimido ya contiene un directorio que contiene el sistema). Deberíamos tener una situación similar a la de la Figura 1.

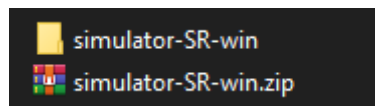


Figura 1 Directorio del Sistema – Descomprimido

3. Para ejecutar el programa principal accedemos a la carpeta descomprimida, en la que vemos lo mostrado en la Figura 2.

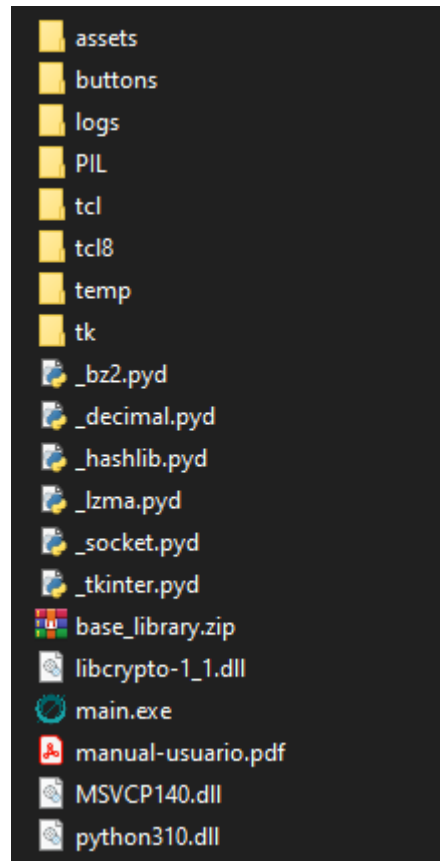


Figura 2 Directorio que contiene el sistema

El archivo que necesitamos ejecutar es el «main.exe», que está señalado en la Figura 2. Si lo ejecutamos, ya podremos ver el sistema en funcionamiento.

A continuación, se describe el resto del contenido de la carpeta, el cuál se aconseja no modificar o actualizar salvo petición y explicación expresa del profesor:

- Archivos de extensión “pyd” y “dll”: son bibliotecas del programa y en el archivo comprimido base_library.zip están los archivos de *bytecode* compilado.
- «manual-usuario.pdf»: es el manual de usuario que se ejecutará en el momento en que el usuario pulse sobre la opción “Manual de ayuda” del menú “Ayuda”.
- “robot_data.json”: contiene los datos necesarios para configurar los circuitos y los robots por defecto. Si se quiere cambiar los pines por defecto, se puede hacer en este archivo. No se recomienda hacer esto.
- Carpetas “assets” y “buttons”: contienen las imágenes png usadas para el *front-end* de la aplicación, en concreto “buttons” contiene los botones de la barra de herramientas y “assets” los gráficos de los robots y el icono de la aplicación.
- Carpeta «logs» y «temp»: se generan posteriormente. La primera almacena los logs que tendréis que enviar al profesor si este os los pide. La segunda contendrá el archivo de código compilado y las correspondientes las librerías usadas por el sistema.

3. Ejecución

Simplemente basta con entrar en la carpeta dónde se descomprimió el simulador previamente y ejecutar el archivo «main.exe» mostrado en la Figura 3.

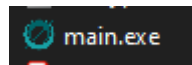


Figura 3 Main.exe

Una vez ejecutado, nos saldrán dos ventanas, la consola (que mostrará salida de Python si es que la hay, incluidos errores); y la ventana principal, que será la que nos permitirá utilizar toda la funcionalidad de la aplicación. Las ventanas se muestran en la Figura 4, la primera en la parte superior izquierda tapada por la segunda.

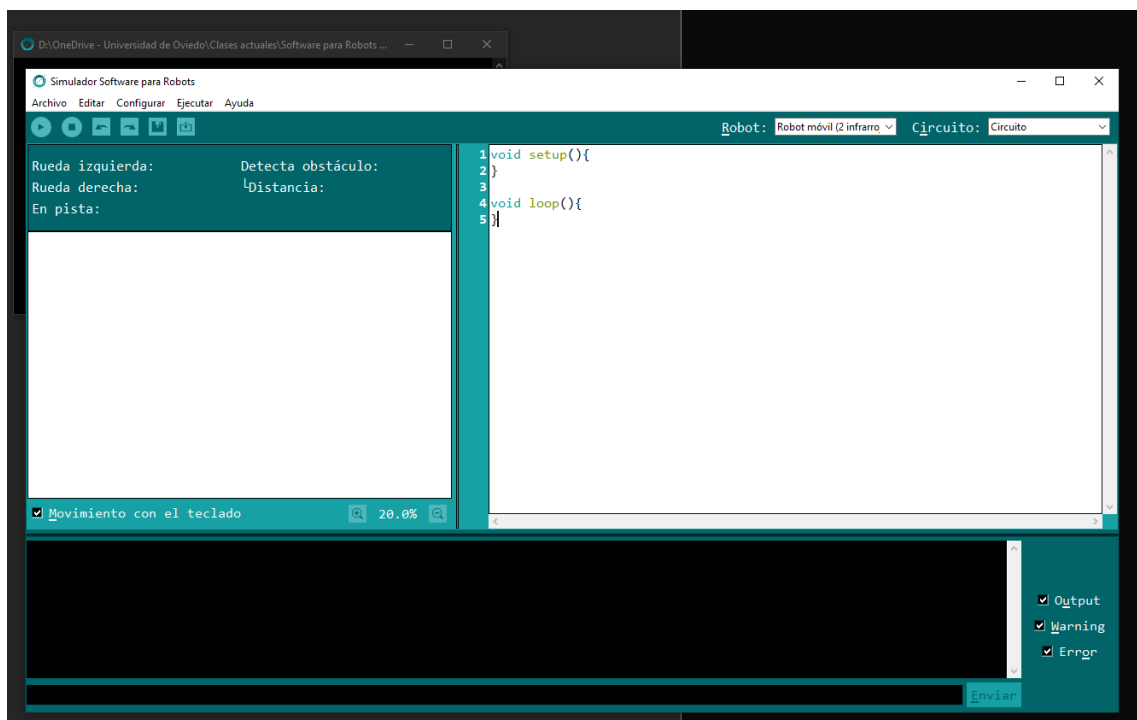


Figura 4 Simulador abierto

Para cerrar el sistema correctamente hay dos opciones:

- Cerrar desde el propio botón de cerrar de la ventana de la aplicación.
- Cerrar desde la opción dada en el menú Archivo de la aplicación.

No es recomendable cerrar la aplicación cerrando la consola o usando el administrador de tareas, es mejor cualquiera de las opciones propuestas anteriormente. Por ello, es mejor dejar estas opciones como las últimas en caso de que fallen las anteriores.

4. Manual de usuario

En este apartado se desarrollará el manual de usuario, que está incluido dentro de la aplicación como manual de ayuda para los usuarios

Visión general del sistema

Al abrir la aplicación nos encontraremos con la ventana que contiene toda la funcionalidad del sistema.

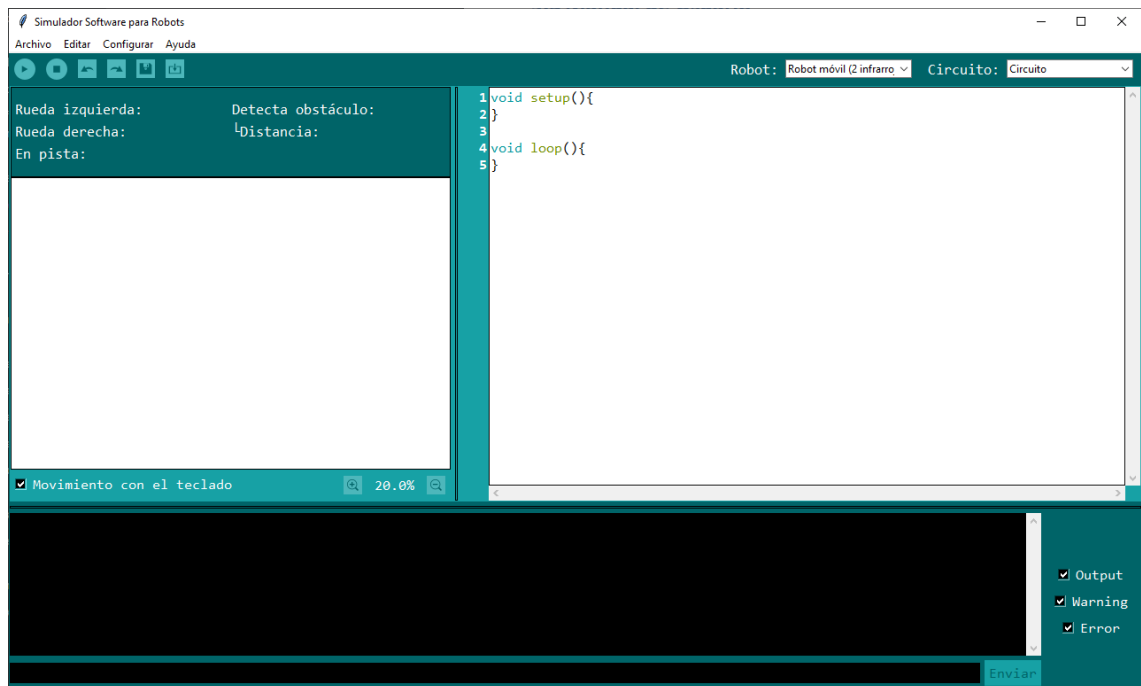


Figura 4.1 Pantalla principal del sistema

En ella podemos ver varios elementos:

- Arriba del todo encontramos el menú de la aplicación.
- Debajo justo del menú, encontramos la barra de herramientas, que contiene todas las herramientas básicas para ejecución, edición y apertura de sketches.
- Después, la interfaz se divide en tres componentes principales:
 - Arriba a la izquierda se encuentra la representación gráfica de los robots.
 - Arriba a la derecha se encuentra el editor de código.
 - Abajo se encuentra la consola de entrada/salida.

A continuación, se explicará las funcionalidades que se podrán encontrar en cada componente mencionado.

Menú

Comenzaremos describiendo la funcionalidad del menú de la aplicación. En la Figura 4 se pueden ver las opciones ofrecidas por dicho menú.

Figura 4 Menú de la aplicación

La primera opción es archivo. Si pulsamos sobre ella, veremos que surge el menú desplegable mostrado en la Figura 2.

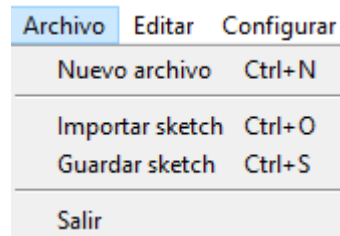


Figura 2 Menú archivo

En este menú podemos ver cuatro opciones:

- Nuevo archivo: nos permitirá crear la estructura de un nuevo sketch. Antes de crear el nuevo archivo nos saldrá un diálogo como el de la Figura 3. Si queremos crear el archivo escogeremos sí, en otro caso basta con escoger no o cerrar el diálogo.

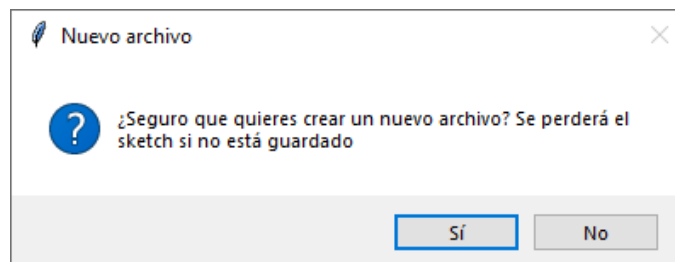


Figura 3 Diálogo de confirmación – Crear nuevo archivo

- Importar sketch: nos permitirá abrir un sketch que queramos editar. Emplea el diálogo típico de apertura de archivos de otras aplicaciones parecidas.
- Guardar sketch: nos permitirá guardar un sketch que hallamos creado/editado. Usa el diálogo clásico de guardado, con lo que podemos sobrescribir el archivo o guardar en el archivo que nosotros escojamos.
- Salir: permite salir de la aplicación, al igual que el botón cerrar de la ventana, solo que, en este caso, lanza un diálogo de confirmación como el de la Figura 4.

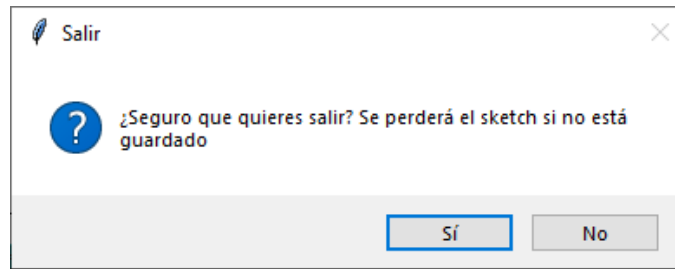


Figura 4 Diálogo de confirmación – Salir de la aplicación

La segunda opción es el menú editar, si pulsamos sobre ella, veremos el menú mostrado en la Figura 5.

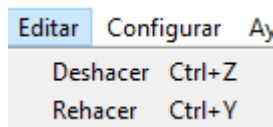


Figura 5 Menú editar

En este menú vemos dos opciones: deshacer y rehacer. Dichas opciones implican el que se deshaga o rehaga acciones del editor de código, sean editar, borrar, añadir o cualquier otra acción que tenga que ver con el sketch. Esto implica incluso que se puede deshacer el crear archivo o importar uno nuevo.

La siguiente opción es el menú de configuración, que contiene una sola opción, Configurar pines, cuyo mnemónico es «ctrl + p». Si accedemos a dicha opción, abriremos ventana de la Figura 6.

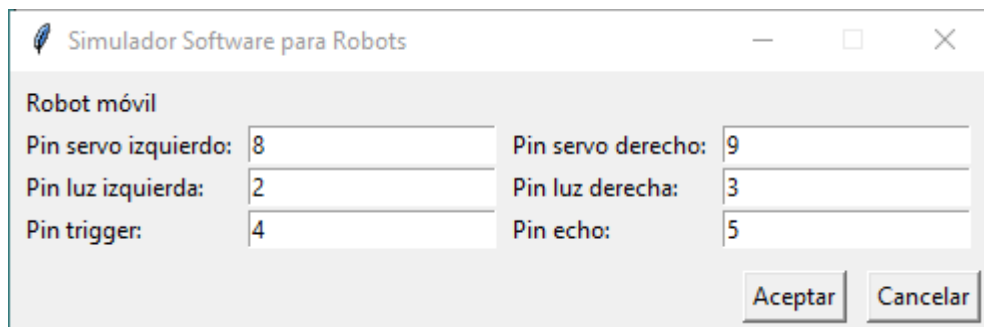


Figura 6 Ventana de configuración de pines del robot

La ventana que se muestra en la Figura 6 varía según el robot que esté elegido en la aplicación. La ventana mostrada en la figura anterior es para el robot móvil con 2 sensores de infrarrojos. Mientras que las figuras siguientes muestran las otras dos opciones, la primera el robot móvil de 4 infrarrojos (Figura 7) y la segunda el actuador lineal (Figura 8).

Figura 7 Ventana de configuración - Robot móvil de 4 sensores infrarrojos

Figura 8 Ventana de configuración – Actuador lineal

Como podemos ver, si el pin es analógico, como en el caso de las ordenadas del joystick, se muestra el número de pin que correspondería en número. Esto no impide que se pueda escribir el pin con la usual forma A0-A5, que, de hecho, es recomendable que se haga así para evitar confusiones.

En la cuarta opción vemos el menú ejecutar, que se puede ver en la Figura 9. Da cuatro opciones: Ejecutar, que comienza la simulación, Detener, que la detiene, Ampliar, que amplía la imagen de la simulación, y Reducir, que la reduce.

Ejecutar	Ayuda
Ejecutar	F5
Detener	Ctrl+F5
Ampliar	Ctrl++
Reducir	Ctrl+-

Figura 9 Menú ejecutar

Con respecto a la quinta y última opción, es el menú de ayuda, que incorpora el «acerca de» de la aplicación y este manual de usuario.

Barra de herramientas

La barra de herramientas de la aplicación se encuentra en la parte superior de la aplicación, justo debajo del menú de esta. La Figura 10 y la Figura 11 muestran las variaciones de dicha barra.



Figura 10 Barra de herramientas – Actuador lineal



Figura 11 Barra de herramientas – Robot móvil

En la primera figura, se ve la barra de herramientas del actuador lineal y en la segunda de ambos robots móviles. La única diferencia entre ambas es que, en el caso del robot móvil, se añade una opción para poder cambiar el circuito que recorrerá el robot.

Como se puede ver en la Figura 11, si pasamos por encima de los botones de la izquierda, se mostrará a la derecha de estos la opción que se va a ejecutar.

Las opciones de este menú son las siguientes:

- Ejecutar: ejecuta el sketch escrito en la parte del editor de código.
- Detener: detiene la ejecución del sketch si este se está ejecutando.
- Deshacer: como en editar, deshace las acciones que se han realizado a la hora de editar el código.
- Rehacer: rehace las acciones que se han realizado a la hora de editar el código.
- Guardar: guarda el sketch de la misma forma que la opción del menú archivo.
- Importar: importa el sketch de la misma forma que la opción del menú archivo.
- Robot: permite cambiar el tipo de robot. Además, cambia la interfaz gráfica según el robot seleccionado.
- Circuito (solo robots móviles): permite cambiar el circuito que recorrerá el robot móvil.

Representación gráfica de los robots

Debajo de la barra de herramientas de la aplicación, hacia la izquierda, se encuentra la parte de la representación gráfica de los robots. Su apariencia es la que muestra en la Figura 12 y Figura 13.

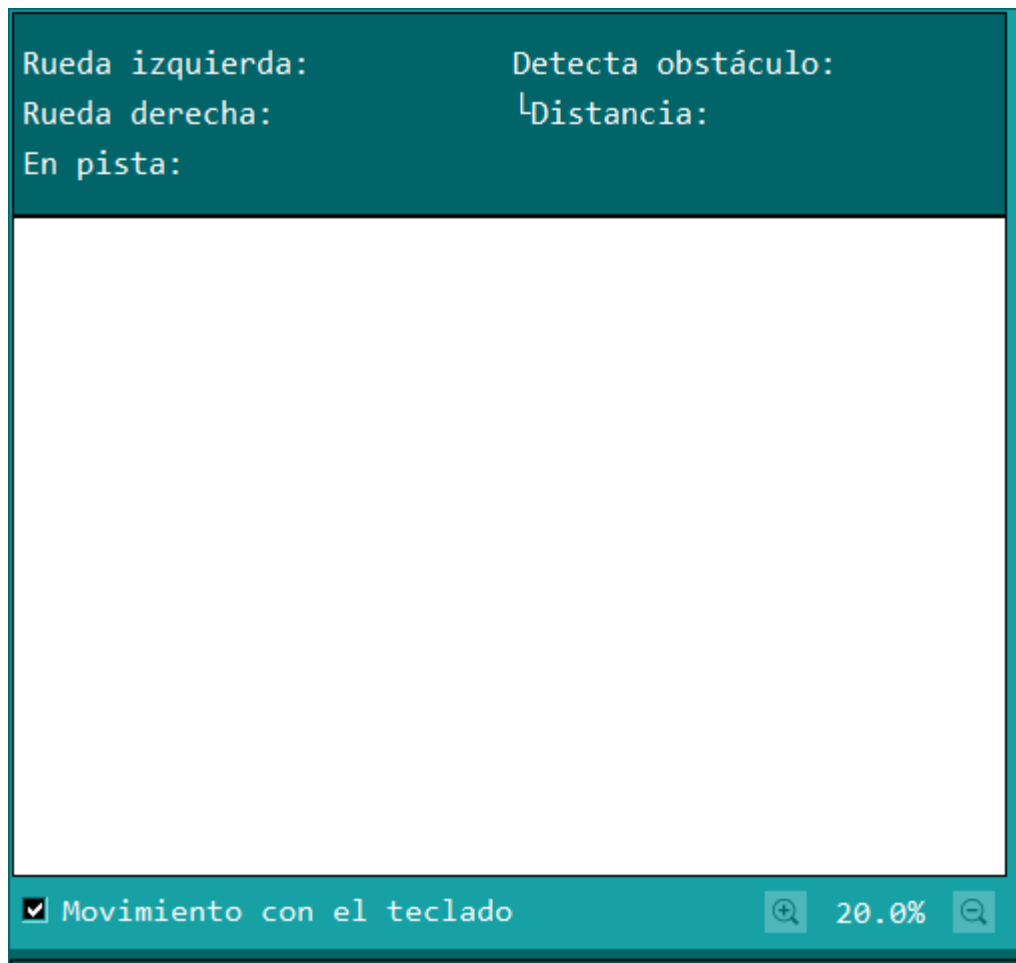


Figura 12 Representación gráfica – Robot móvil – Sin ejecutar

Como vemos en la Figura 12, la representación gráfica cuenta con un HUD que muestra durante la ejecución los datos de que realiza cada componente del robot. Debajo está un recuadro en blanco, que será donde se muestre la simulación del robot. Por último, y abajo del todo, se ve una *checkbox* para activar o desactivar el movimiento por teclado. Esto es **IMPORTANTE**. Para ejecutar el código (automáticamente), se debe desactivar esta casilla. Si no, el robot estará quieto, a menos que pulsemos las teclas de movimiento WASD. Esto es útil para posicionar el robot en una posición determinada de forma manual.

La parte derecha de esta barra inferior es la ampliación del dibujo, con el botón izquierdo se acerca y con el derecho se aleja. Se puede realizar lo mismo con el botón *scroll* del ratón, hacia delante amplía, hacia atrás aleja. La ampliación recomendada en una pantalla de 1920x1080 es 20% con el programa en pantalla completa. De esta manera se podrá ver todo lo que haga el robot.



Figura 13 Representación gráfica – Actuador lineal – Sin ejecutar

Como vemos en la Figura 13, cambia ligeramente el HUD y se añade una nueva barra. El resto de funcionalidad es la misma que en la Figura 12.

La nueva barra que se ha añadido es la que permite simular el joystick. El botón del joystick funciona pulsando el botón. La dirección del joystick funciona según la Figura 14.

0, 0	512, 0	1023, 0
0, 512	512, 512	1023, 512
0, 1023	512, 1023	1023, 1023

Figura 14 Movimiento del joystick

A continuación, se muestran tres figuras (Figura 15, Figura 16 y Figura 17) mostrando cada robot en modo ejecución.

En la Figura 15 podemos ver como se vería el robot móvil de 2 sensores de luz en su representación gráfica. La posición de los sensores de luz se puede ver mediante los círculos rojos. Dichos círculos se colorean de azul si el sensor detecta la pista. Si esto sucede, el HUD mostrará (en el lado correspondiente) si el sensor está en pista o no.

Respecto al HUD, también mostrará en qué dirección se mueve cada rueda. El color de la flecha mostrará la velocidad del servo (azul – rápido, amarillo – velocidad media, rojo – lento o parado).

El HUD también mostrará si el sensor de ultrasonidos detecta un obstáculo, y si lo detecta mostrará la distancia hacia el obstáculo, que no equivalente a la real.

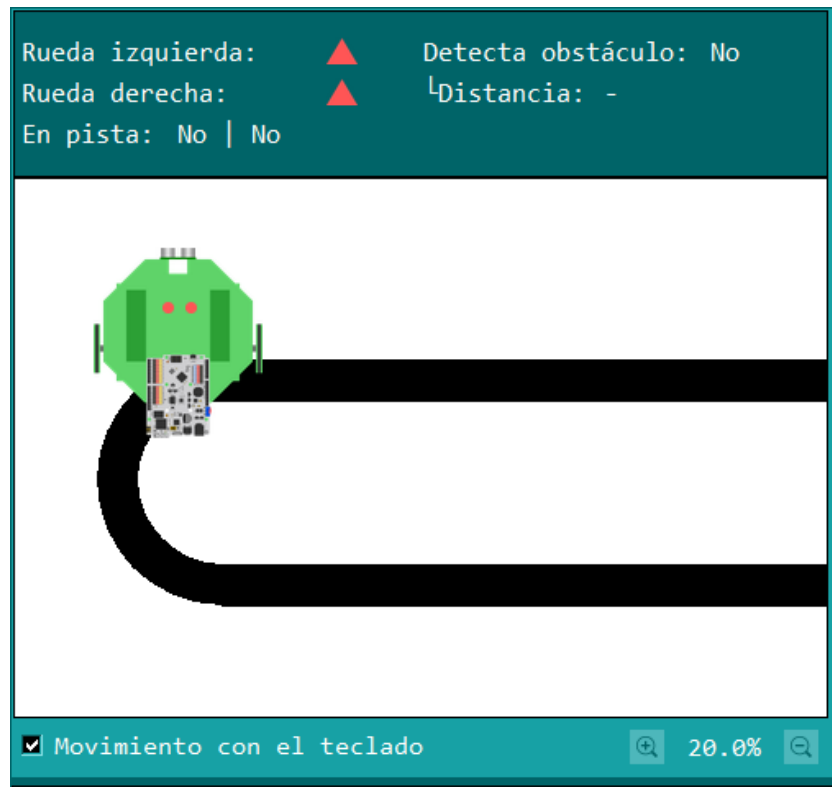


Figura 15 Representación gráfica – Robot móvil 2 – Ejecutando

En la Figura 15Figura 16 se ve que lo único que cambia en cuanto a la representación gráfica del robot móvil de 4 sensores es el número de círculos (sensores), y lo que detectan, mostrado en el HUD.

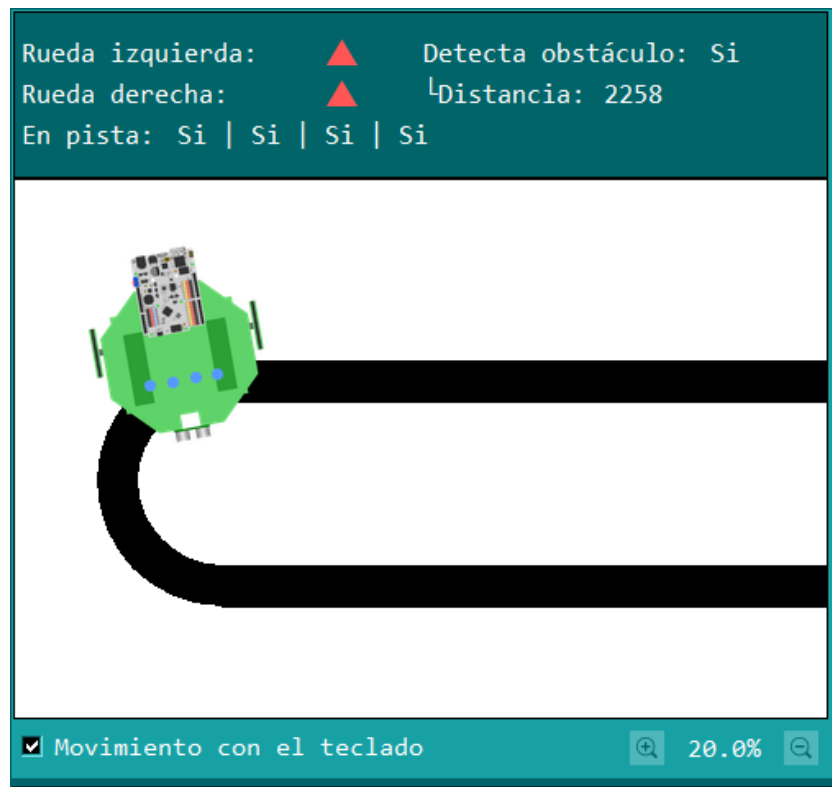


Figura 16 Representación gráfica – Robot móvil 4 – Ejecutando

La Figura 17 muestra el actuador lineal. En esta, vemos en el robot los dos sensores de final de carro que hacen de tope a cada extremo del actuador lineal. Si el bloque del actuador toca un botón, la parte que colisiona con el bloque se vuelve roja, como se muestra en la Figura 18. Si no, estará verde como en la Figura 17.

En el caso del HUD, se muestra en qué dirección se mueve el servo y su velocidad de la misma forma que en los robots móviles. También se muestra si los botones están siendo pulsados o no.

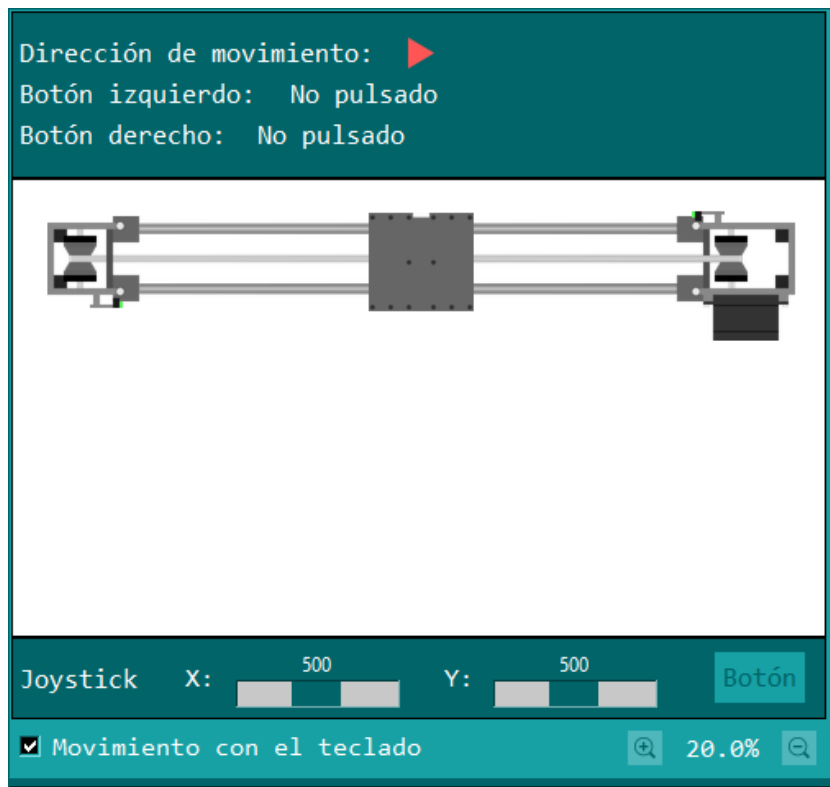


Figura 17 Representación gráfica – Actuador lineal - Ejecutando

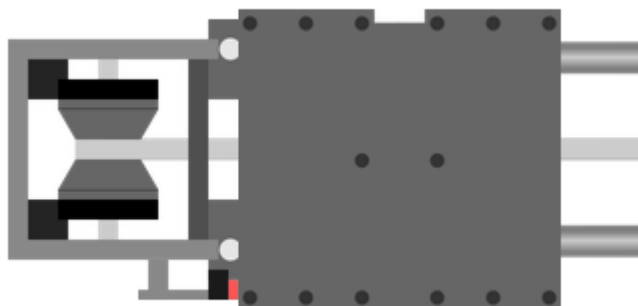


Figura 18 Actuador lineal - El botón tope está pulsado

Editor de código

A la derecha de la representación gráfica está el editor de código. La Figura 19 muestra cómo se estructura.

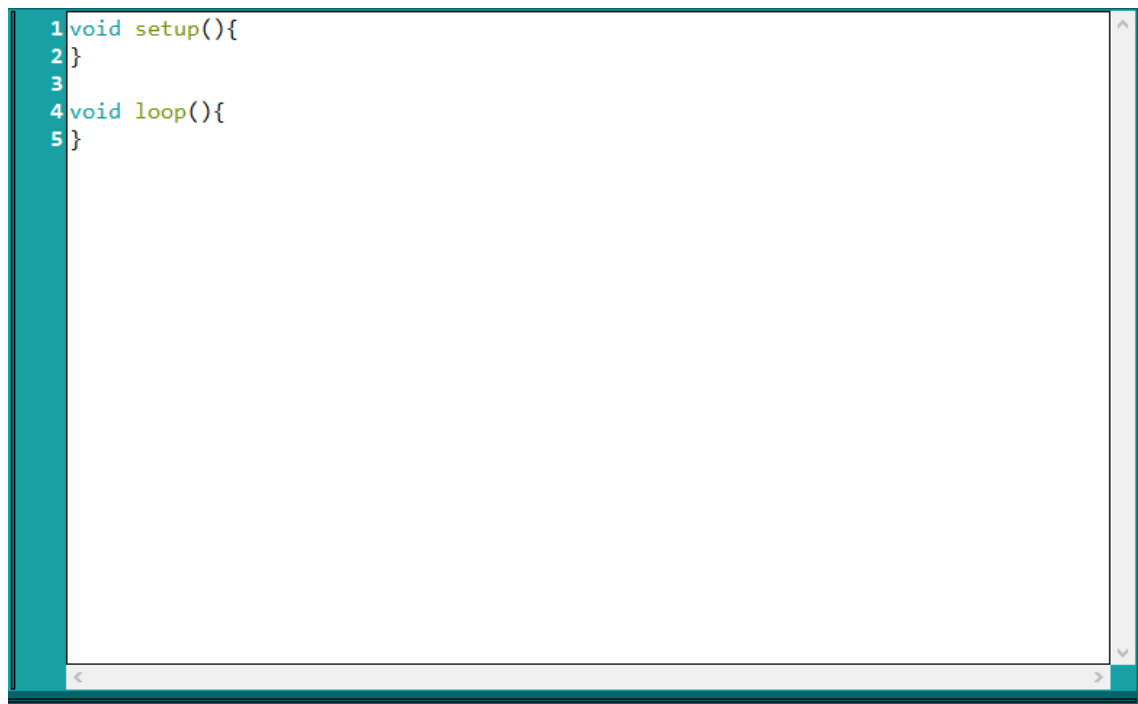


Figura 19 Editor de código

El editor tiene coloreado sintáctico, el mismo que Arduino IDE versión 1.X. Además, a la izquierda se muestra el número de línea correspondiente para facilitar la búsqueda de errores y advertencias.

Se puede navegar a través del código usando las barras de *scroll* lateral y vertical. Se volverán activas en caso de ser necesarias. Se pueden usar utilizando la rueda del ratón. El editor no autocompleta el código.

Consola de entrada/salida

Debajo de los dos últimos componentes se sitúa la consola, que se muestra en la Figura 20.

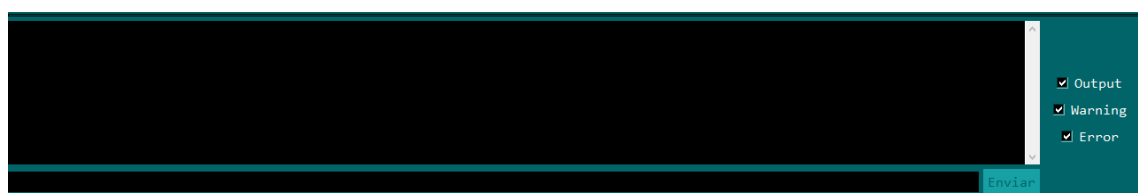


Figura 20 Consola

En la parte superior izquierda tenemos la consola como tal. Mostrará la salida del sketch, con un código de colores en el cual rojo es un error, amarillo una advertencia y blanco un mensaje normal. Estos mensajes se pueden filtrar con los *checkboxes* de la derecha (se pueden usar a la vez varios).

En la parte inferior hay un campo de texto que es el que se debe usar si se quiere hacer una entrada por consola. La entrada se envía usando el botón enviar.

Otras características a tener en cuenta

La aplicación se bloquea a todo tipo de interacción si se mete en un bucle o un *delay*, esta situación se termina en el momento que se sale del bucle, salvo que se vuelva infinito, o este el *delay* en funcionamiento. El arreglo de este problema ya está siendo desarrollado, pero mientras tanto, si hace falta cambiar algo en la interfaz gráfica cuando se bloquea, algunas veces funciona al realizar la interacción con la interfaz bloqueada.

Muchos errores de ejecución pueden ser resultado de una mala configuración de los pines, por lo que es recomendable revisarlos siempre que se vaya a ejecutar el programa. Si se quiere cambiar los pines por defecto del programa, se puede en el archivo “robot_data.json”, aunque no es muy recomendable tocar dicho archivo.

Cuando el botón ejecutar o parar se vuelve amarillo es que se está ejecutando alguna tarea. En el momento que se vuelve al color normal (turquesa), dicha tarea se terminó de realizar.

Si se modifica el código para recargar, hay que darle al botón detener y después al ejecutar.

La palabra reservada «Define» no funciona y no tiene soporte, pues es una mala práctica el utilizarla.