
Presentación →
**Proyecto
Heracross**



Índice de contenidos



01. Reintroducción

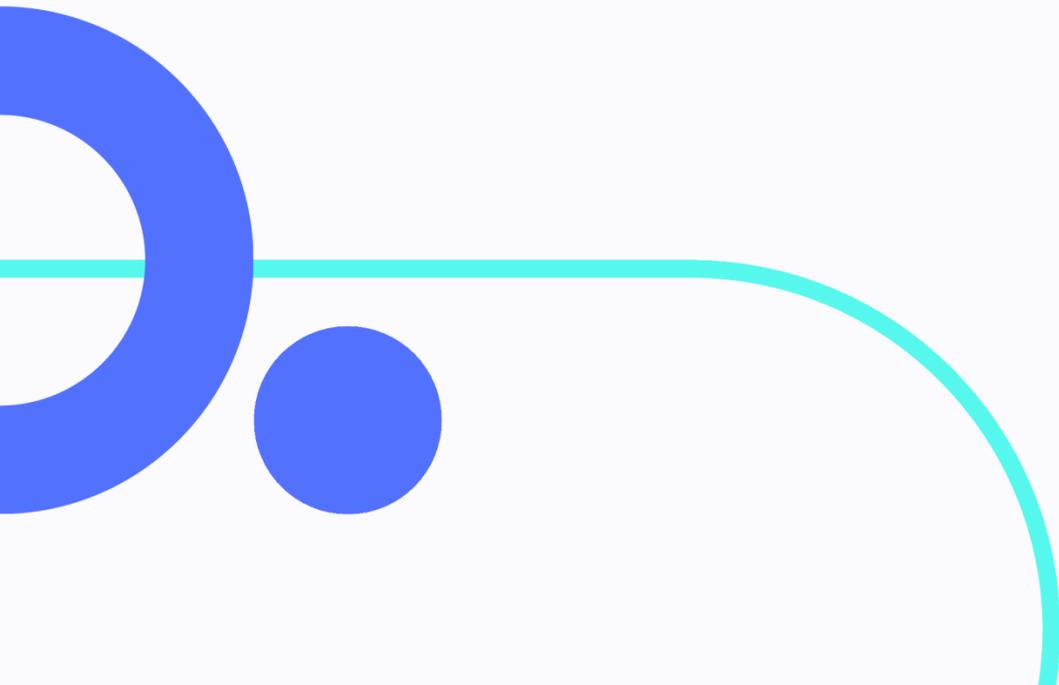
02. Carta Gantt

03. Manual de usuario

04. Pruebas del robot

05. Resultados

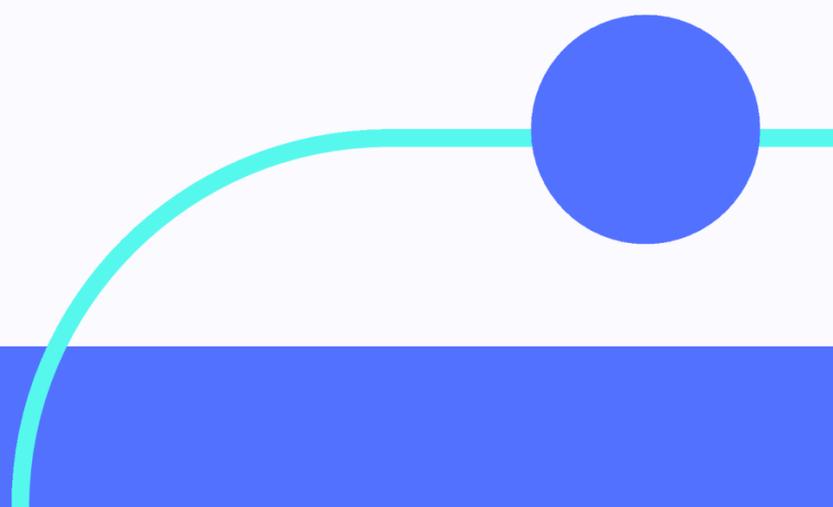
06. Conclusión





→ 01. Reintroducción ←

A continuación procederemos rápidamente a recapitular lo visto en la presentación anterior



Puntos tratados anteriormente



Objetivos y organización

Crear un robot que sea capaz de agarrar una pelota de pingpong y llevarlo a un lugar, aprendiendo como utilizar linux y python ev3.

Se repartieron los roles entre los integrantes del grupo



Planificación, estimación de costos y progreso

Se planificó utilizando una carta Gantt.

Logramos avanzar con el modelo del robot y con la garra



Analisis y diseño, arquitectura e interfaz gráfica

Requisitos funcionales y no funcionales además de la arquitecturas de como funciona nuestro sistema y el diseño de la interfaz gráfica.



Implementación y descripción de los programas

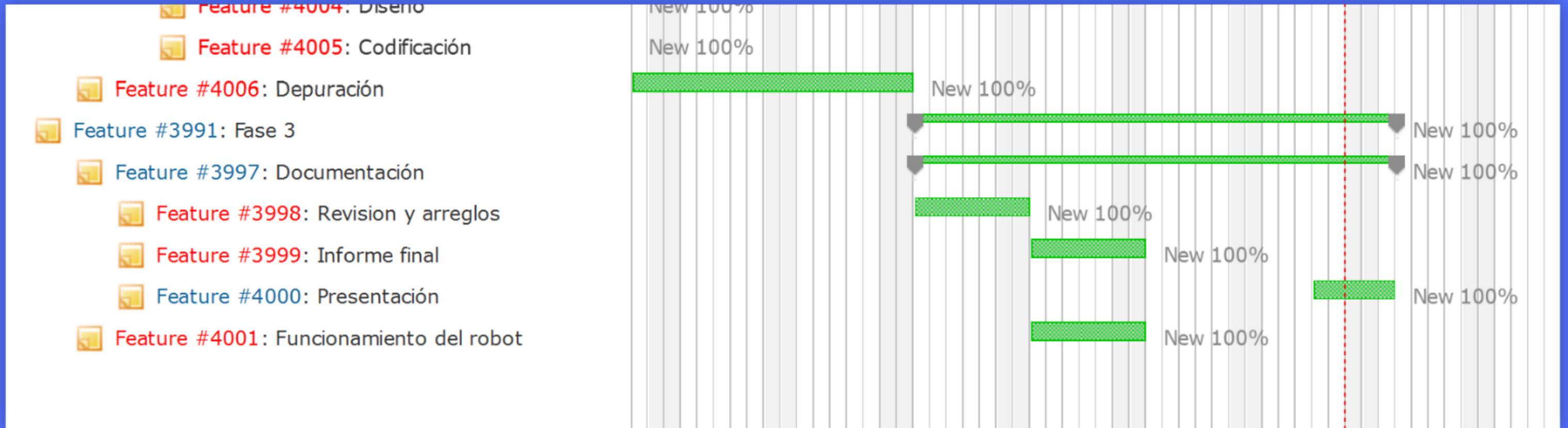
Como funcionan los movimientos del robot y el levantamiento de la pelota de pingpong y los programas del servidor y de la interfaz grafica.



Problemas, soluciones y resultados

Logramos resolver distintos problemas y también concluimos la construcción y funcionamiento del robot con éxito.

02. Carta Gantt ←



03. Manual de usuario

➔ Acciones del robot

- 1 Movimiento rectilíneo: El robot es capaz de desplazarse hacia adelante o atrás.
- 2 Rotación: El robot es capaz de rotar en ambas direcciones.
- 3 Operación de la garra: El robot está equipado con una garra que se puede abrir, cerrar, subir o bajar.

➔ Requerimientos mínimos

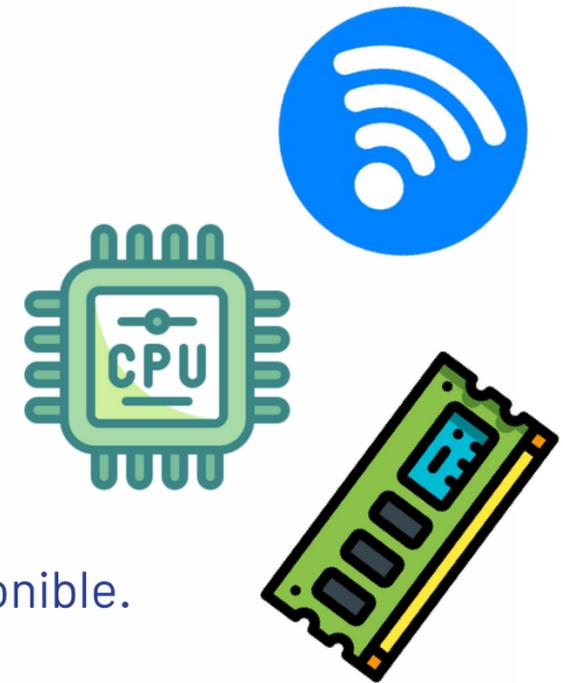
SOFTWARE

- Sistema operativo: Windows o Linux
- Lenguaje de programación: Python 3.11 o superior
- IDE: Visual Studio Code
- Extensiones: Lego Mindstorm EV3 para Visual Studio Code
- Librerías: Pybricks, Sockets.



HARDWARE

- Conexión: Wi-fi, adaptador Wi-fi USB
- Ordenador con, al menos:
 - Procesador: 2 GHz
 - Memoria RAM: 1GB
 - Espacio de almacenamiento disponible.



03. Manual de usuario

➔ Procedimientos

Instalación

1. Insertar el adaptador WI-FI USB en el puerto del EV3 Brick.
2. En VSCode, cree carpeta del proyecto con los archivos: "EV3server.py" e "Interfaz.py".
3. Conectar el ordenador y el EV3 Brick a la misma red WI-FI y luego establecer la conexión en VSCode.
4. Verificar los valores de la variable "ev3_ip" en Interfaz.py, este valor debe igualar al de la esquina superior izquierda del menú principal del EV3 Brick.
5. Actualizar la variable "port" en ambos códigos por uno disponible. Para esto, use el terminal de Visual Studio Code, con el comando:

```
Get-NetTCPConnection | Where-Object { $_.State -eq 'Listen' },
```

 busque un puerto en el que diga "Listening" y "0.0.0.0", el valor de "port" debe ser el puerto asociado.
6. Ejecutar ambos códigos.

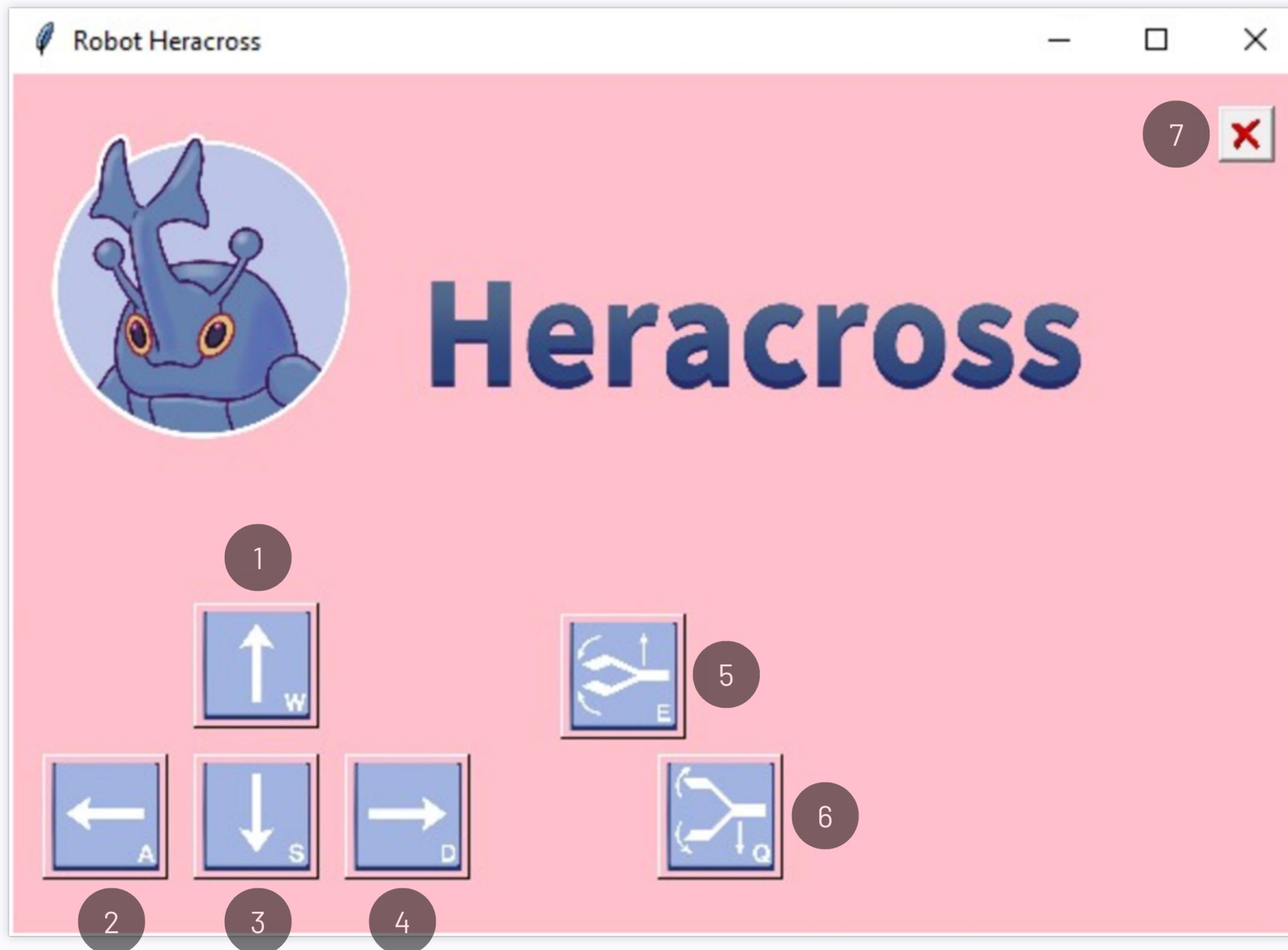
Obtención del software

- Descargar e instalar Python
<https://www.python.org/downloads/>
- Descargar e instalar VSCode
<https://code.visualstudio.com/download>
- Buscar en extensiones de VSCode: LEGO Mindstorm EV3.
- Ejecutar "pip install pybricks" en el terminal de VSCode una vez conectado al robot (para Pybricks).
- Ejecutar "pip install sockets" en el terminal de VSCode una vez conectado al robot (para el Sockets).
- Ejecutar "pip install tk" en el terminal de VSCode una vez conectado al robot (para Tkinter).

03. Manual de usuario

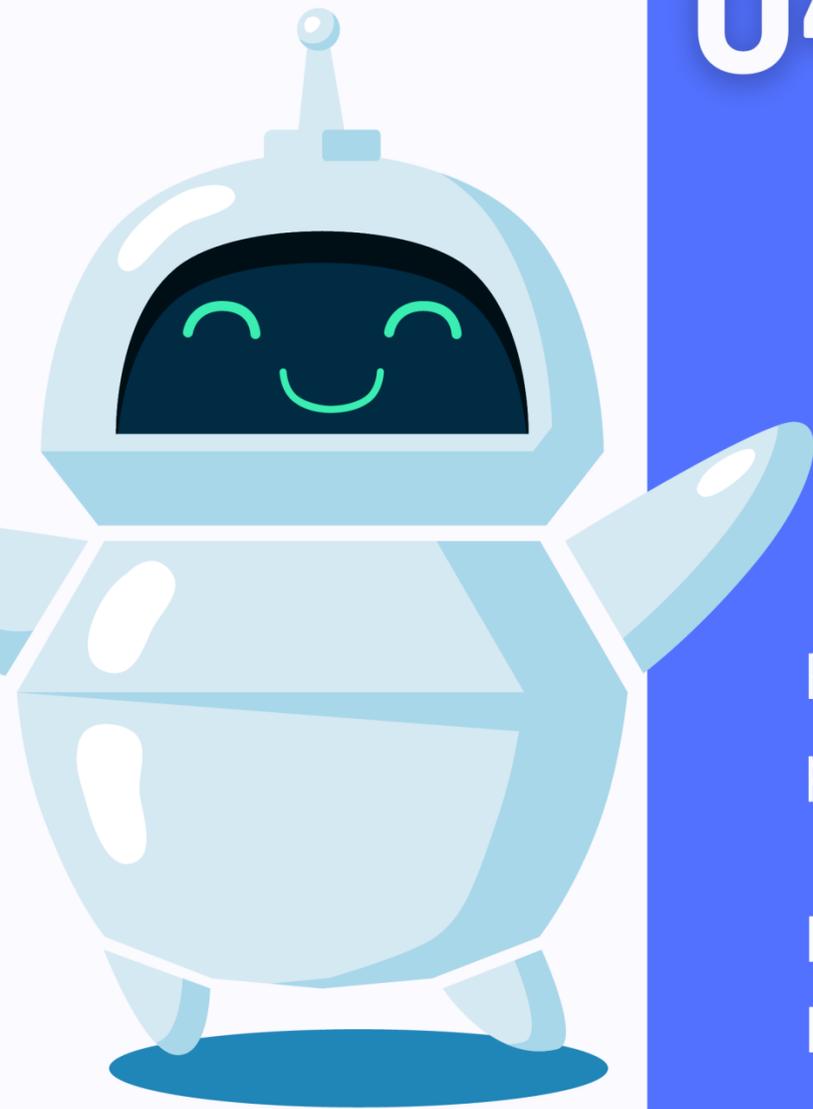
➔ Uso del software

- 1 Botón de movimiento hacia adelante
- 2 Botón de rotación a la izquierda
- 3 Botón de movimiento hacia atrás
- 4 Botón de rotación a la derecha



- 7 Botón para cerrar el programa
- 5 Botón que cierra y luego levanta la garra
- 6 Botón que baja y luego abre la garra

04. Pruebas realizadas



Programar y subir los archivos necesarios para la interfaz gráfica y permitir el avance del robot.

Movimientos del robot (adelante, hacia atrás y giros).

Movimientos de la garra: abrir, cerrar, levantar y dejar la pelota de ping pong.

Determinar que el robot funcione mediante una interfaz gráfica.



05. Resultados

Conexión entre el cliente y el robot mediante Wi-fi local.

Los movimientos logrados mediante la programación hecha en el IDE Visual Studio Code y Python.

Movimientos del robot en la interfaz gráfica usando Tkinter en Python.

06. Conclusión



Durante el desarrollo de este proyecto se cumplieron los objetivos planteados. En conclusión, este proyecto fue una experiencia en la cual pudimos aplicar nuestros conocimientos que en el futuro nos ayudará en a trabajar en un entorno profesional de manera formal y eficiente.