



UNIVERSIDAD DE TARAPACÁ  
*Universidad del Estado*

Ingeniería@  
Computación e Informática

# BRAZO ROBÓTICO



**Integrantes:** Eduardo Suaña  
Dylan Calderón  
Matías Agriano  
Benjamín Sucso

**Profesor:** Baris Klobertanz  
**Asignatura:** Proyecto I

# ÍNDICE

<b>01</b>	Introducción	<b>08</b>	Requerimientos funcionales
<b>02</b>	Problema	<b>09</b>	Requerimientos no funcionales
<b>03</b>	Objetivos	<b>10</b>	Arquitectura
<b>04</b>	Organización del equipo y roles	<b>11</b>	Implementación
<b>05</b>	Carta Gantt	<b>12</b>	Demostración
<b>06</b>	Gestión de riesgos	<b>13</b>	Manual de usuario
<b>07</b>	Fundamento de los movimientos	<b>14</b>	Conclusión

# INTRODUCCIÓN

En este proyecto, se construyó un brazo robótico utilizando el sistema LEGO Spike Prime para realizar tareas de movimiento y agarre. El objetivo fue crear un programa en Python que permitiera manejar el robot de forma inalámbrica desde un computador mediante Bluetooth. Gracias a esto, el usuario puede controlar con precisión el giro, la altura y la garra del brazo a través de una pantalla con botones fáciles de usar.



# PROBLEMÁTICA

**Riesgo Actual:** Alta exposición de trabajadores a condiciones peligrosas en minería subterránea.

**Foco del Conflicto:** Los procesos de carga y manipulación de material son los más críticos para la seguridad.

**Necesidad:** Desarrollo de una herramienta robótica teleoperada que aleje al humano del peligro, mejorando la seguridad y productividad.

# OBJETIVOS

## Objetivo general:

Diseñar y programar un brazo robótico teleoperado utilizando LEGO Spike Prime y MicroPython, capaz de manipular materiales para simular faenas mineras seguras.

## Objetivos específicos:

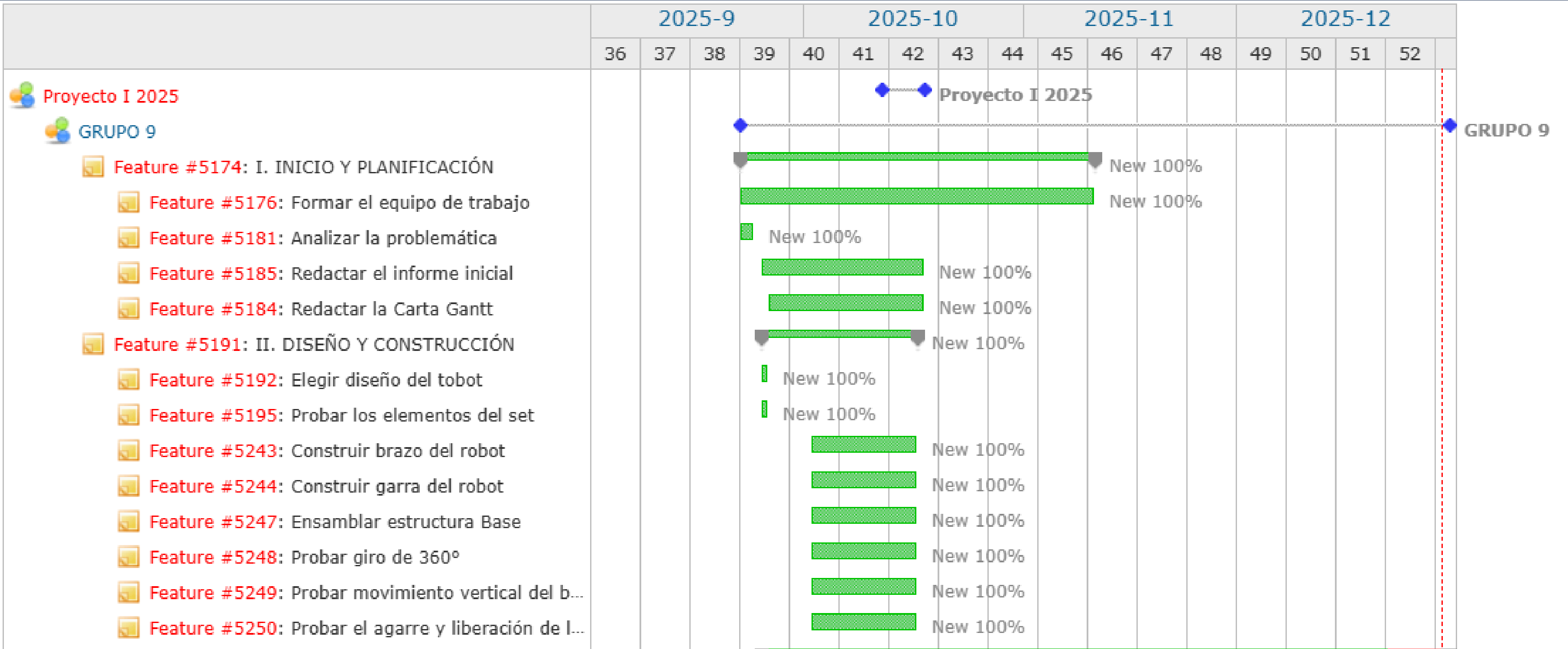
- Construir la estructura mecánica del brazo y la garra asegurando estabilidad.
- Programar en Pybricks la lógica de control para motores.
- Diseñar una interfaz gráfica que permita la teleoperación remota del robot.
- Validar el prototipo mediante pruebas de carga y precisión de movimiento.

# ESTRUCTURA ORGANIZACIONAL

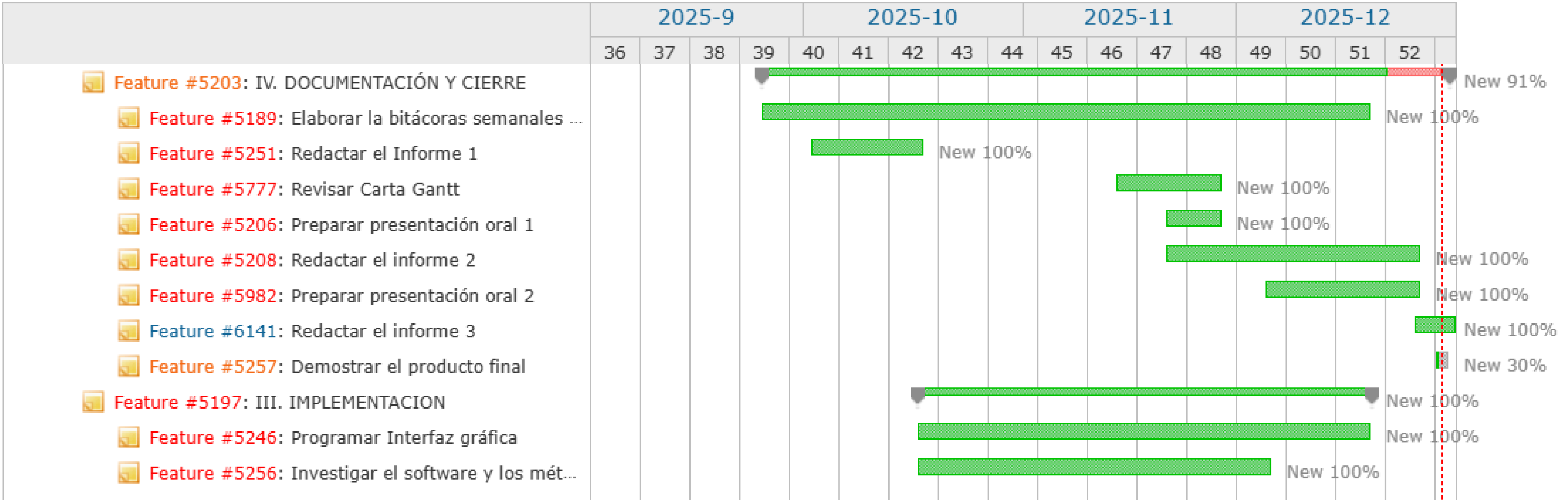
Rol	Integrante	Descripción
Jefe de Proyecto	Dylan Calderón	Coordinación general
Programador	Benjamín Sucso	Desarrollador de código
Diseñador	Matías Agriano	Diseño físico y virtual
Documentador	Eduardo Suaña	Supervisión de los entregables
Constructor	Dylan Calderón	Encargado del montaje físico



**CARTA GANTT**







# GESTION DE RIESGOS

Riesgo identificado	Nivel de impacto(bajo / medio / alto)	Medida preventiva / Solución	Estado y mitigación aplicada
Abandono de personal.	Alto	Redefinir roles, ofrecer ayuda mutua para tratar de prevenir.	Ocurrido: Se debieron reasignar tareas y un integrante asumió dos roles para continuar con el proyecto.
Mal funcionamiento o desempeño del robot.	Alto	Volver a la fase de diseño teniendo en cuenta las complicaciones.	Mitigado: Se realizó un análisis físico y estructural previo para garantizar la estabilidad del brazo antes de las pruebas.
Limitación de piezas del Set de LEGO.	Medio	Planificar bien el diseño antes de construir y adaptar el modelo según las piezas disponibles o solicitar piezas adicionales.	Ocurrido: Durante la construcción faltaron piezas clave.
Cambios de actividades en horarios destinados al proyecto.	Bajo	Acordar reuniones y horas extraordinarias.	Ocurrido: Se pidió sala para poder seguir avanzando con nuestro proyecto para no quedarnos atrás.

# FUNDAMENTO DE LOS MOVIMIENTOS

**Objetivo del Análisis:** Confirmar que la potencia de los motores es suficiente para elevar la carga de la mina (20 cm) de forma segura y sin interrupciones.

## Datos:

Masa de carga(m): 0,15 kg

Gravedad (g): 9,8 m/s<sup>2</sup>

Distancia(h): 0,2 m

## Cálculo del peso:

$$P = m \cdot g \longrightarrow 0,15 \text{ kg} \cdot 9,8 \text{ m/s}^2 = 1,47 \text{ N}$$

## **Cálculo de Trabajo Mecánico:**

$$W = P \cdot h \longrightarrow 1,47 \text{ N} \cdot 0,20 \text{ m} = 0,294 \text{ J}$$

## **Resultado de validación:**

El requerimiento de 0,294 Joules asegura que el brazo puede realizar la tarea de elevación sin forzar los motores, garantizando un funcionamiento estable y duradero.



# REQUERIMIENTOS FUNCIONALES

## • RF1 MOVIMIENTO ROTATIVO

El brazo robótico debe ser capaz de rotar su base en 360 grados.

## • RF2 MOVIMIENTO FLEXIBLE

El brazo robótico debe poder articular el codo para alcanzar un objeto.

## • RF3 MANIPULACIÓN DE CARGA


La garra debe abrirse y cerrarse para sujetar firmemente un bloque de LEGO estándar sin dejarlo caer durante el traslado.

## • RF4 TELEOPERACIÓN

El sistema debe permitir el control manual de todos los motores a través de la interfaz gráfica.

## • RF5 PARADA DE EMERGENCIA

La interfaz debe contar con una función para detener todos los motores inmediatamente en caso de una situación de riesgo.





# REQUERIMIENTOS NO FUNCIONALES

## • USABILIDAD

La interfaz gráfica debe permitir que un usuario sin experiencia previa logre conectar el robot y realizar un movimiento básico en menos de 30 segundos. Los controles deben estar etiquetados claramente en español.

## • RENDIMIENTO


El tiempo de respuesta entre la pulsación de un botón en la interfaz (Cliente) y la reacción del motor del robot (Servidor) debe ser inferior a 200 milisegundos para garantizar una teleoperación fluida.

## • SEGURIDAD

En caso de pérdida de conexión Bluetooth, el robot debe detener todos sus motores automáticamente en un lapso no mayor a 1 segundo para evitar daños.

## • DISPONIBILIDAD

El sistema debe ser capaz de mantener la conexión activa y operativa durante al menos 10 minutos continuos, tiempo estimado para una demostración estándar.





# ARQUITECTURA DE SOFTWARE

01

Modelo de trabajo: El sistema funciona como un equipo de dos partes: un Cliente (el computador) y un Servidor (el robot LEGO).

02

El cerebro: Es donde está la pantalla de control (GUI). Aquí el usuario da las órdenes usando botones creados con la herramienta Tkinter.

03

El ejecutor: Recibe las órdenes y mueve los motores. Funciona con un programa llamado MicroPython que lee los comandos enviados desde el PC.

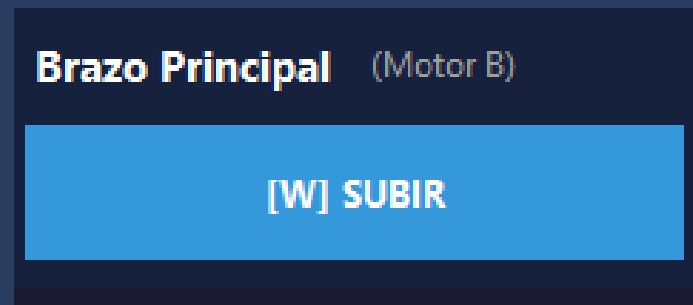
04

La conexión: Se comunican sin cables a través de Bluetooth (BLE), lo que permite manejar el brazo a distancia de forma rápida.



# IMPLEMENTACIÓN

Interfaz de Usuario (GUI)

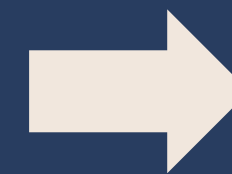


```
# interfaz.py (Cliente)
def start_move(self, event):
    # Detecta tecla 'w' y envía comando
    if event.char == 'w':
        self.controller.send(b'w')
```

Captura de evento 'W' y envío **asíncrono**



Transmisión de 1 Byte  
(eficiencia) vía librería bleak.



```
# main.py (En el Hub)
# Bucle infinito de escucha
if select.select([sys.stdin], [], [], 0)[0]:
    char = sys.stdin.read(1)
    if char == "w":
        # Acciona motor B (Brazo)
        motor_brazo.run(50)
```

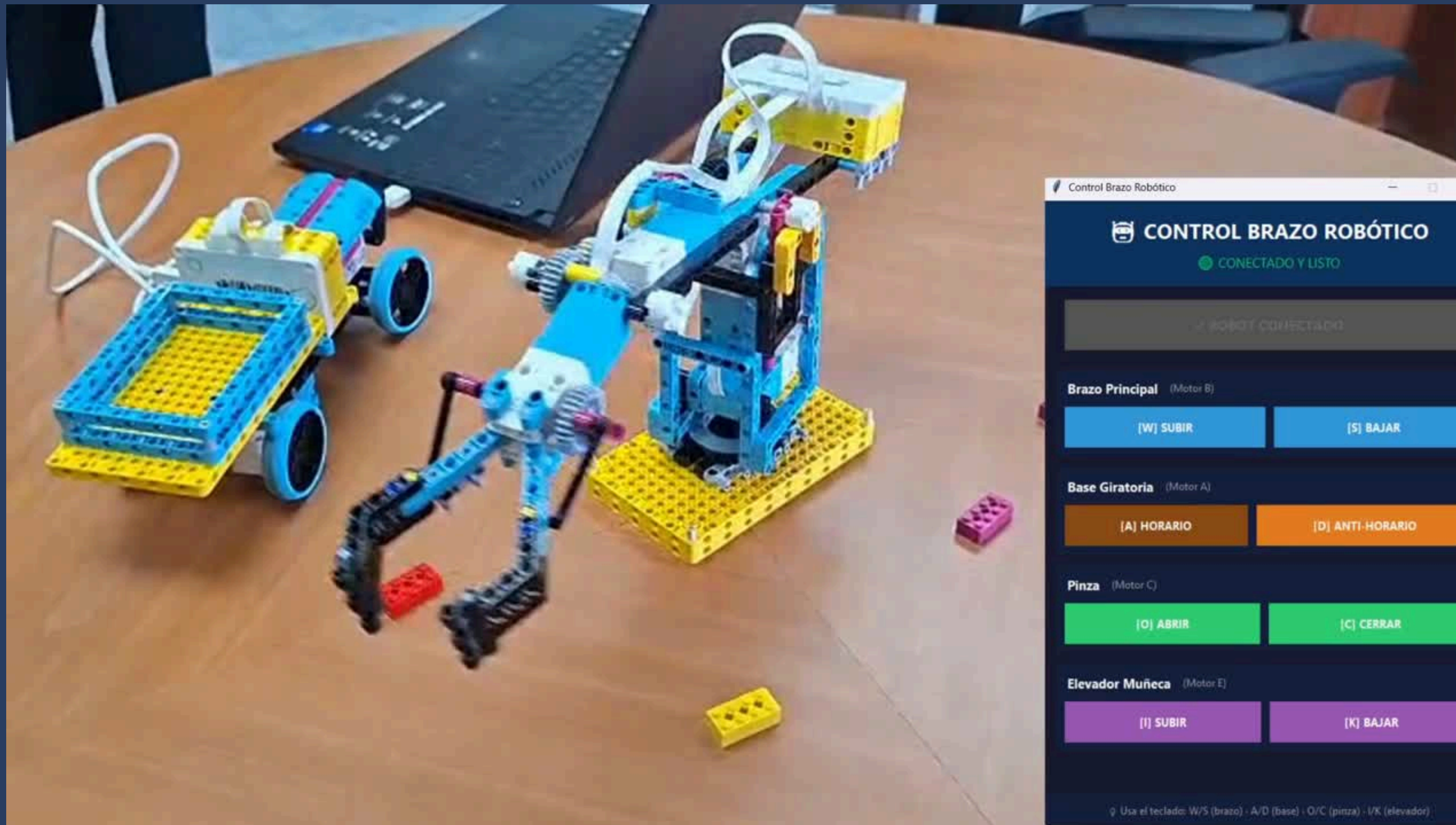
Decodificación y actuación física (<200ms)

HUB




The image features a dark blue background with stylized, light brown clouds in the top-left and bottom-right corners. The word "DEMOSTRACIÓN" is centered in a bold, white, sans-serif font.

# DEMOSTRACIÓN



Control Brazo Robótico

 **CONTROL BRAZO ROBÓTICO**

● CONECTADO Y LISTO

ROBOT CONECTADO

**Brazo Principal** (Motor B)

[W] SUBIR

[S] BAJAR

**Base Giratoria** (Motor A)

[A] HORARIO

[D] ANTI-HORARIO

**Pinza** (Motor C)

[O] ABRIR

[C] CERRAR

**Elevador Muñeca** (Motor E)

[I] SUBIR

[K] BAJAR

🔗 Usa el teclado: W/S (brazo) - A/D (base) - O/C (pinza) - I/K (elevador)





# MANUAL DE USUARIO

Pasos para la Operación:

Conexión: Encender el Hub LEGO Spike Prime y presionar el botón "Conectar" en la aplicación (esperar indicador de estado en verde).

Control de Ejes: Utilizar los botones de dirección para mover la Base (Sentido horario/antihorario) y el Brazo Principal (Subir/Bajar).

Manipulación de Carga: Controlar la Muñeca para posicionar la garra y usar los botones de Abrir/Cerrar para sujetar el material minero.

Atajos de Teclado: Para mayor fluidez, se pueden usar las teclas preconfiguradas (ej: flechas dirección) sin necesidad de usar el ratón.



# CONCLUSIONES

Se cumplió el objetivo de diseñar y construir un brazo robótico teleoperado mediante Python y LEGO Spike Prime, logrando movimientos precisos de rotación, elevación y agarre para entornos mineros. El hito principal fue la implementación de programación asíncrona, técnica indispensable que permitió la comunicación Bluetooth constante sin bloquear la capacidad de respuesta de la interfaz gráfica (GUI). El equipo demostró resiliencia al resolver problemas reales de hardware (falta de piezas) y humanos (redistribución de roles), asegurando la entrega final a pesar de los cambios internos.

Propuestas de mejoras:

Para futuras versiones del proyecto, se sugiere añadir movilidad al brazo robótico mediante ruedas, lo que permitiría desplazarlo dentro de distintos entornos de trabajo. Además, se podrían automatizar ciertos movimientos o secuencias, de modo que se puedan ejecutar con un solo botón, aumentando la eficiencia y facilitando su operación en tareas repetitivas.



MUCHAS  
**GRACIAS**

