



Proyecto I

BRAZO ROBÓTICO DE AGARRE

FASE II

Integrantes:

- Benjamín Aguilera
- Franco Churata
- Carlos Cossio
- Bryan Palacios
- Joaquín Quezada

Profesor: Baris Klobertanz Quiroz

ÍNDICE



- 01** Introducción y Problemáticas
- 02** Objetivos del Proyecto
- 03** Planificación y Carta Gantt
- 04** Análisis y Diseño: Requerimientos y Arquitectura
- 05** Implementación: Fundamentos Físicos
- 06** Desarrollo del Software
- 07** Interfaz Gráfica de Usuario (GUI)
- 08** Resultados y Estado Actual
- 09** Conclusiones y Trabajo Futuro

INTRODUCCIÓN Y PROBLEMÁTICA

Contexto Industrial: La operación minera enfrenta riesgos críticos en las labores de transporte y carga manual de material en zonas peligrosas.

Problemática a Resolver: Necesidad urgente de reducir la exposición humana mediante la automatización de procesos de manipulación de mineral (bloques).

Solución Tecnológica: Implementación de un Brazo Robótico de Agarre independiente que gestiona la carga y la transfiere hacia un organizador de minerales.

Enfoque de la Fase 2: Validación técnica de la ingeniería del sistema (Física, Mecánica y Software) y refinamiento de la lógica de control para interactuar con el vehículo.

OBJETIVOS DEL PROYECTO

Objetivo General:

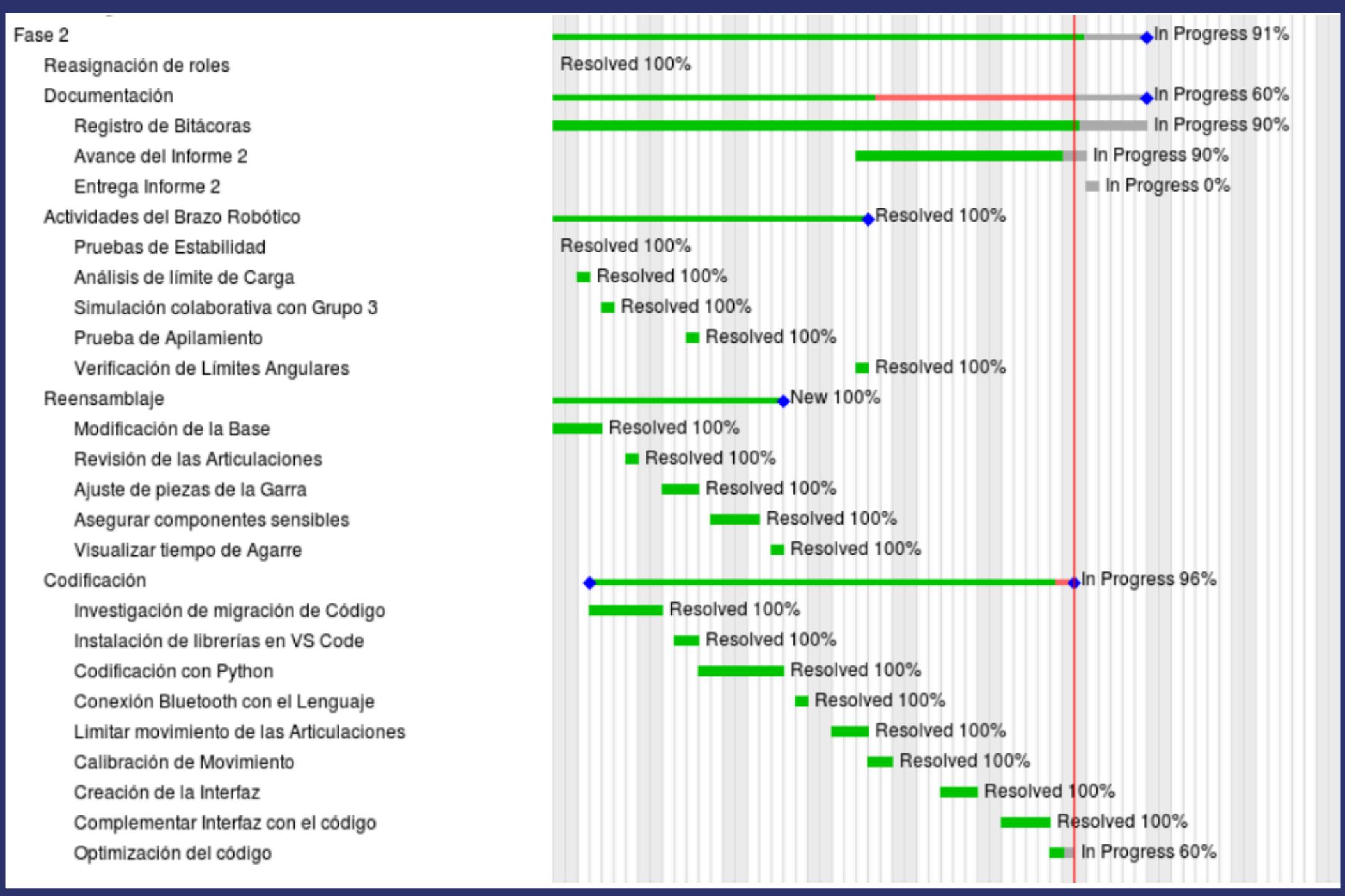
Desarrollar y programar un brazo robótico estacionario capaz de sujetar, levantar y depositar bloques en un vehículo robótico de transporte externo.

Objetivos Específicos (Avance Fase 2):

- Estabilidad Mecánica: Garantizar la precisión en el agarre para asegurar una transferencia limpia al vehículo sin caídas.
- Control de Software: Implementar lógica de control robusta en Python, migrando desde la programación por bloques.
- Coordinación del Sistema: Validar la capacidad del brazo para operar en el entorno colaborativo del proyecto.

PLANIFICACIÓN Y CARTA GANTT

Es un representación del cronograma del proyecto que muestra cómo se planificaron y ejecutaron las principales actividades de la Fase 2, incluyendo documentación, trabajo con el brazo robótico, reensamble y programación.



ANÁLISIS Y DISEÑO: REQUERIMIENTOS

Requerimientos Funcionales (Refinamiento)

- Control de Manipulación y Cinemática:** El sistema garantiza un ciclo completo de carga reversible (Apertura/Cierre simétricos) y una navegación precisa mediante ajuste dinámico de altura (Posición de Agarre) y rotación de 360° sobre el eje principal.
- Gestión de Estado e Inicialización:** Incluye rutinas de auto-calibración de motores al inicio (Home) y monitoreo constante del estado de conexión para asegurar la consistencia operativa.
- Interfaz de Mando:** El operador dispone de controles específicos mapeados para la ejecución unitaria de cada articulación y visualización de telemetría básica.

Estándares de Calidad y Métricas (RNF)

- Tiempo de Respuesta (Latencia):** <500 ms para asegurar control en tiempo real.
- Precisión y Fiabilidad:** Tolerancia de 0% de caídas de carga y desviación posicional máxima de 0.5 mm.
- Seguridad Estructural:** Limitación de velocidad a 45 RPM en actuadores para protección mecánica.

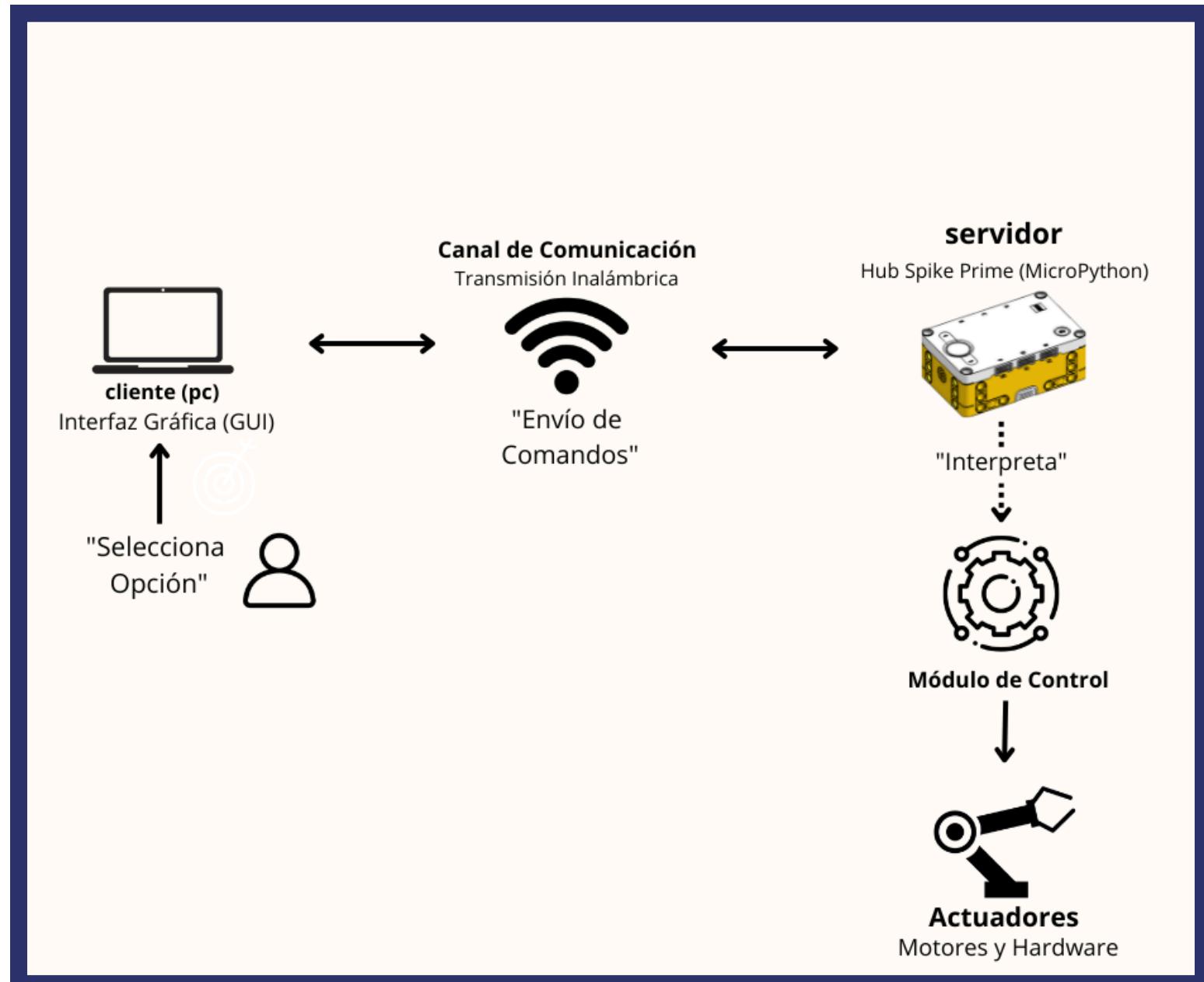
ANÁLISIS Y DISEÑO: ARQUITECTURA

Arquitectura Distribuida (Cliente-Servidor)

- El diseño implementa un patrón Cliente-Servidor que desacopla la capa de presentación (PC) de la lógica de control crítica (Hub SPIKE Prime). Esta separación garantiza la modularidad y facilita el mantenimiento evolutivo del código.
- El Servidor (en el robot) procesa la lógica física, mientras que el Cliente gestiona la interacción humana, asegurando que el procesamiento pesado no dependa de la interfaz gráfica.

Protocolo de Comunicación

- **Transmisión de Datos:** Se utiliza un canal inalámbrico vía Bluetooth, operando mediante el intercambio de paquetes de bytes estructurados .
- **Bidireccionalidad:** El sistema soporta flujo de doble vía: envío de comandos de control (Cliente → Servidor) y recepción de confirmaciones de estado (Servidor → Cliente), eliminando la dependencia de infraestructura de red externa (WiFi).



IMPLEMENTACIÓN: FUNDAMENTOS FÍSICOS

Esta sección justifica la configuración mecánica seleccionada para el brazo robótico, aplicando conceptos básicos de física vistos en la asignatura Introducción a la Física.

El análisis considera las capacidades reales de torque y velocidad de los motores LEGO SPIKE Prime, así como las limitaciones estructurales del material utilizado.

Velocidad Angular – Rendimiento del Sistema

Movimiento de rotación completa de la garra (360°).

Tiempo máximo requerido: 3 segundos.

Velocidad angular requerida: $120^\circ/\text{s}$ ($\approx 20 \text{ RPM}$).

Conclusión:

El motor SPIKE Prime supera este valor, cumpliendo el rendimiento exigido y permitiendo un movimiento estable y eficiente.

Trabajo Mecánico – Capacidad de Carga

Objeto manipulado: bloque LEGO ligero (3 g).

Altura de elevación: 10 cm.

Trabajo mecánico requerido: 0.00294 J o 2.94 mJ.

Conclusión:

El esfuerzo requerido es mínimo; el motor puede levantar la carga sin dificultad.

DESARROLLO DEL SOFTWARE

Estructura Cliente-Servidor (Python & MicroPython)

Cliente (PC): Una interfaz específica desarrollada en Python que funciona como un centro de control, encargada de capturar y gestionar las órdenes emitidas por el operador.

El Hub Spike Prime es un programa en MicroPython que recibe y ejecuta comandos. Interpreta las instrucciones y mueve los motores conectados para realizar tareas físicas con precisión.

Canal de Comunicación: Enlace inalámbrico bidireccional mediante Bluetooth para el envío de paquetes de datos en tiempo real con un alcance máximo y una velocidad de transmisión óptima.

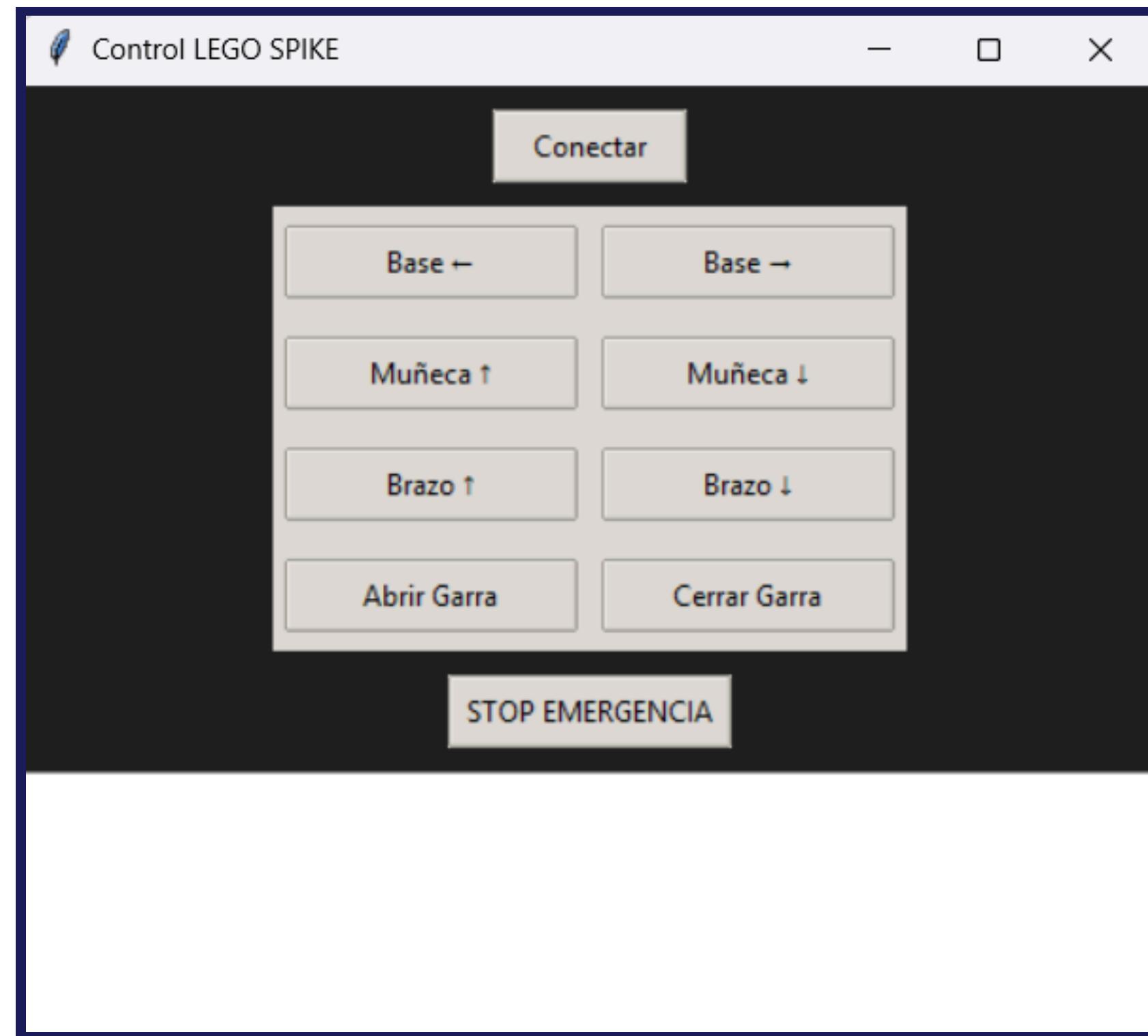
Módulos Críticos del Sistema

Módulo de Recepción: Decodifica las instrucciones entrantes para evitar errores de interpretación, asegurando que cada comando sea procesado correctamente.

Control de Actuadores: Gestiona las rutinas de elevación a alturas específicas, rotación de 360° con precisión de un grado y sujeción con precisión angular de 0.5 grados.

Reporte de Estado: Envía telemetría básica, incluyendo el estado de los motores (velocidad, temperatura y consumo de energía), de vuelta a la interfaz cada 5 segundos.

INTERFAZ GRÁFICA DE USUARIO (GUI)



RESULTADOS Y ESTADO ACTUAL

1. Validación Integral del Backend y Mecánica

- **Operatividad Mecánica:** El sistema ha alcanzado un estado funcional estable. Las pruebas confirman la ejecución correcta de las maniobras de sujeción, elevación, descenso y rotación de 360° (Cumplimiento de RF1, RF2, RF3, RF4, RF5, RF9 y RF10).
- **Estabilidad de Carga:** Se validó que el brazo manipula los bloques y realiza el traslado sin incidentes mecánicos ni caídas, cumpliendo con los estándares de seguridad definidos.
- **Lógica de Control:** Los algoritmos de control de motores (Servidor) interpretan correctamente las instrucciones, ejecutando los movimientos con la precisión angular requerida.

RESULTADOS Y ESTADO ACTUAL

2. Estado de la Interfaz de Usuario (GUI)

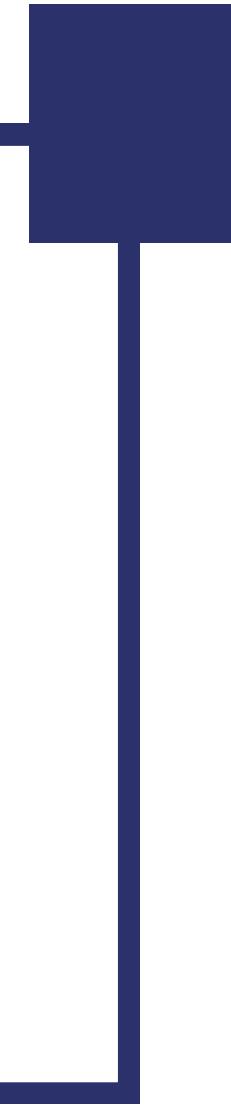
La etapa de diseño visual (wireframing) y el mapeo lógico de los botones se encuentran finalizados.

- **Implementación Visual:** La interfaz gráfica de usuario (GUI) ya ha sido desarrollada e implementada, permitiendo el control manual del brazo robótico mediante una ventana gráfica dedicada. La validación operativa del sistema se realiza a través de esta interfaz, la cual centraliza las acciones de control y mejora la interacción con el usuario.

CONCLUSIONES Y TRABAJO FUTURO

La fase II del proyecto cumplió con los objetivos fundamentales de construcción y control del sistema, logrando un brazo robótico mecánicamente estable y funcional, capaz de manipular bloques de forma segura mediante acciones de agarre, transporte y liberación. Se validó la física del movimiento y la integración de los motores con el Hub SPIKE Prime, demostrando una estructura robusta y un diseño mecánico adecuado.

Durante el desarrollo se presentaron desafíos asociados a la comunicación inalámbrica y la disponibilidad de componentes, los cuales fueron resueltos mediante ajustes en el código y el uso del kit de expansión. Como trabajo futuro, se plantea como prioridad el perfeccionamiento de la interfaz gráfica de usuario (GUI) y la automatización de secuencias de operación, con el fin de mejorar la usabilidad y eficiencia del sistema.



MUCHAS GRACIAS

Proyecto I