



Proyecto I

BRAZO ROBÓTICO DE AGARRE

Integrantes:

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

Profesor: Baris Klobertanz Quiroz

TABLA DE CONTENIDOS

- | | | | |
|-----------|-----------------------------------|-----------|-------------------|
| 01 | Introducción | 10 | Manual de Usuario |
| 02 | Objetivos del proyecto | 11 | Resultados |
| 03 | Organizacion del equipo | 12 | Conclusion |
| 04 | Planificacion del proyecto | | |
| 05 | Fundamento de los movimientos | | |
| 06 | Requerimientos del sistema | | |
| 07 | Diseño y Arquitectura del sistema | | |
| 08 | Implementacion | | |
| 09 | Validación y Prueba funcional | | |

PROBLEMA

La minería moderna exige la automatización de los procesos de carga y descarga con el fin de mitigar los riesgos críticos asociados al transporte y la manipulación manual de material en entornos peligrosos. Esta necesidad surge principalmente de la importancia de reducir la exposición humana y aumentar la seguridad operativa en dichas tareas.

En este contexto, el proyecto tiene como objetivo el desarrollo de un sistema robótico autónomo y preciso, capaz de recoger de manera confiable unidades de mineral representadas mediante bloques, e integrarse de forma eficiente con una cadena de transporte también autónoma. La fiabilidad del agarre, la precisión en los movimientos y la seguridad operativa constituyen los aspectos prioritarios del sistema propuesto.

OBJETIVO GENERAL

El proyecto tiene como objetivo principal el desarrollo e implementación de un brazo robótico de agarre fijo, programado para la transferencia eficiente, confiable y segura de unidades de material representadas mediante bloques. El sistema es operado mediante una interfaz de control manual y está diseñado para integrarse dentro de una cadena de transporte automatizada a escala experimental, contribuyendo a mejorar la coordinación operativa y el desempeño general de sistemas robóticos colaborativos.

OBJETIVOS ESPECÍFICOS

Los objetivos específicos del proyecto se estructuran en torno a tres ejes fundamentales:

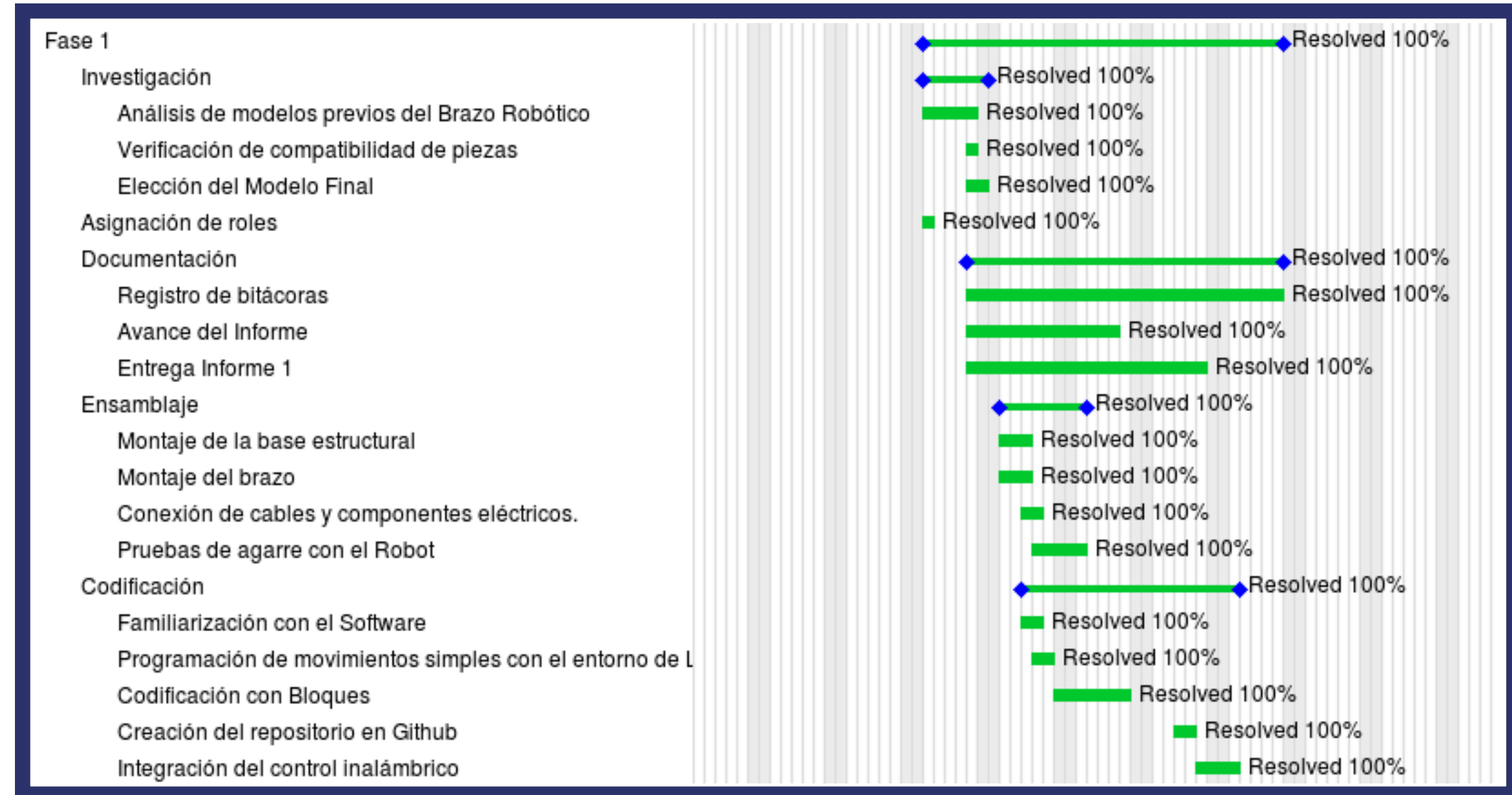
- **Diseño mecánico y estabilidad:** Analizar y seleccionar un diseño mecánico eficiente utilizando piezas LEGO, con el fin de garantizar la estabilidad estructural del brazo robótico y la precisión en los movimientos de sujeción y manipulación de carga.
- **Control y programación:** Desarrollar la lógica de control del sistema mediante la programación de secuencias de movimiento, comenzando con entornos visuales y migrando posteriormente a código Python, con el objetivo de permitir un control externo del brazo y considerar su futura coordinación con sistemas de transporte automatizados.
- **Documentación y gestión del conocimiento:** Documentar de manera sistemática cada etapa del desarrollo a través de informes técnicos y bitácoras semanales rotativas, asegurando una gestión ordenada del proyecto y promoviendo el aprendizaje colaborativo entre los integrantes del equipo.

ORGANIZACIÓN DEL EQUIPO

Integrante	Rol	Función Principal
Carlos Cossio	Jefe de Proyecto / Documentador	Planificación del proyecto, organización del equipo, seguimiento de avances y apoyo transversal en programación y documentación.
Joaquin Quezada	Programación y Control	Desarrollo del código de control del brazo robótico en Python, implementación de la comunicación Cliente-Servidor vía Bluetooth y desarrollo de la interfaz gráfica de usuario.
Benjamín Aguilera	Diseño Mecánico	Diseño estructural del brazo robótico, selección de componentes LEGO y aseguramiento de la estabilidad mecánica.
Franco Churata Brayan Palacios	Ensamblador	Armado del brazo robótico y verificación experimental de su correcto funcionamiento mecánico y operativo.

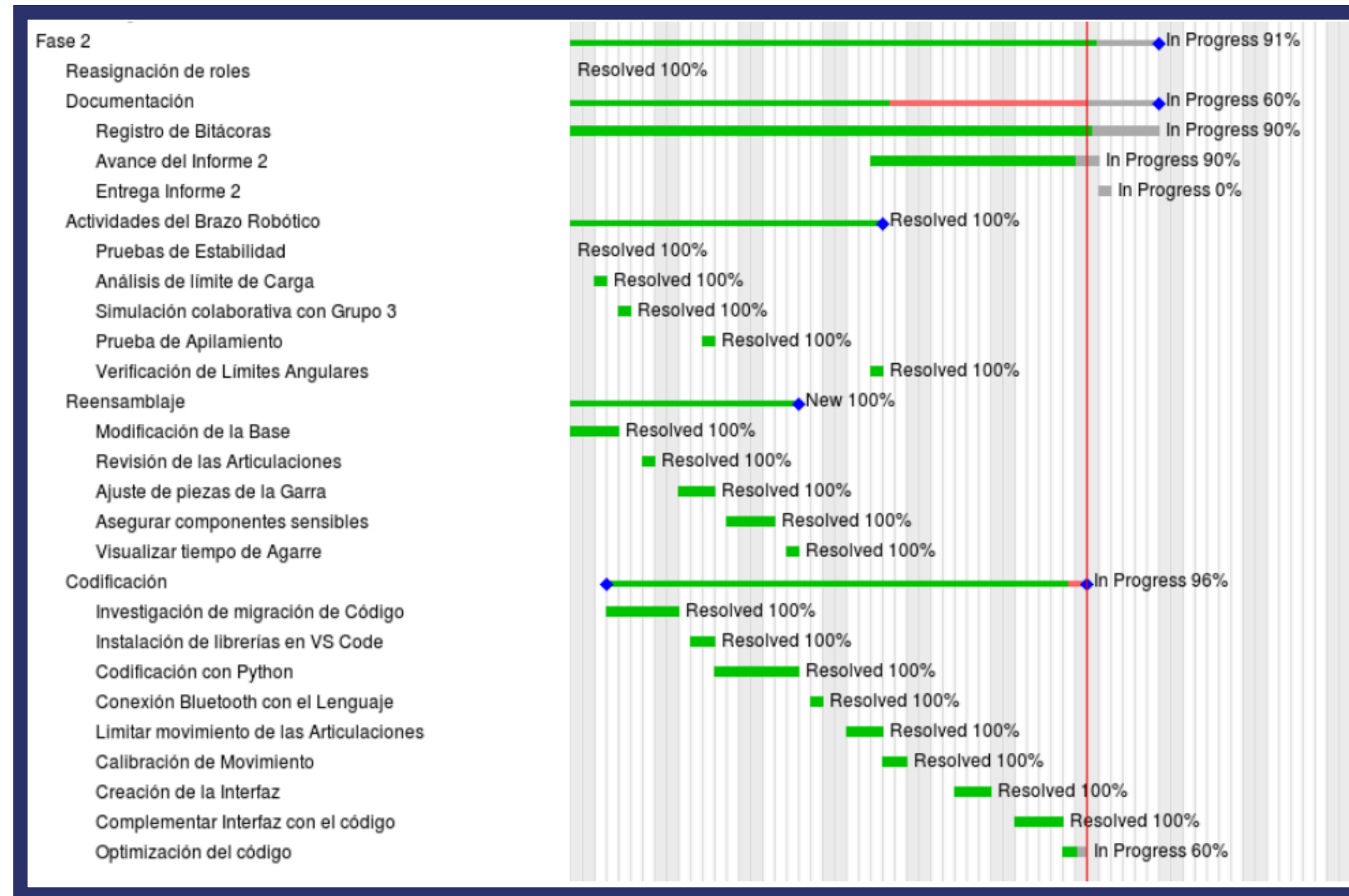
CARTA GANTT

La Fase 1 se centró en la cimentación estratégica del proyecto, definiendo el alcance, levantando requerimientos y planificando recursos. Se establecieron las bases organizacionales y cronogramas, coordinando el diseño mecánico inicial y la estructuración del desarrollo. Esto fue crucial para mitigar riesgos y asegurar una transición exitosa a la implementación técnica del brazo robótico.



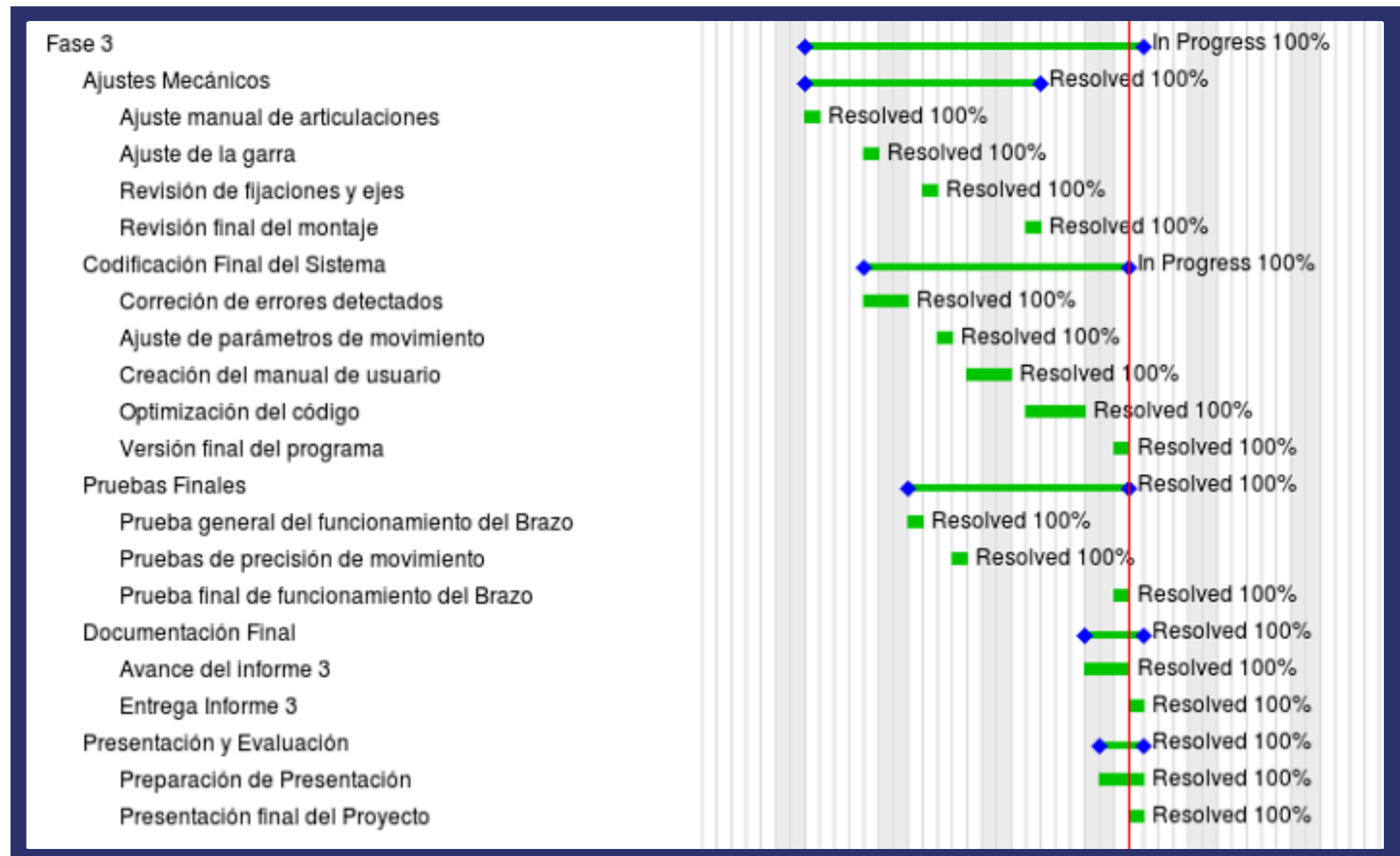
CARTA GANTT

La Fase 2 se centró en la construcción física del brazo y la migración de la lógica de control a código Python/MicroPython. Incluyó el reensamblaje estructural para mejorar el torque, la validación de la carga física y el establecimiento de la comunicación inalámbrica Cliente-Servidor, logrando la operatividad estable y cumpliendo con los requisitos de precisión y latencia.



CARTA GANTT

La Fase 3 se enfoca en la consolidación y cierre del proyecto. Las actividades principales incluyeron la optimización del sistema (ajustes de precisión y corrección de errores), el control de calidad mediante pruebas integrales de estabilidad, la finalización de la documentación técnica y la interfaz gráfica, y la preparación para la demostración práctica que valida el cumplimiento de los requisitos.



Riesgos	Nivel de Impacto	Acción Remedial
Incompatibilidad con mando externo.	1	Buscar alternativas dentro del software Spike Prime o mediante Python.
Pérdida del robot o desarme entre clases.	2	Guardar el robot en un lugar seguro y registrar su estado al finalizar cada sesión.
Falta de tiempo y carga académica.	3	Coordinar mejor los horarios y distribuir tareas entre los miembros.
Errores de codificación.	3	Solicitar piezas de reemplazo o usar piezas del kit de expansión.
Errores de codificación.	4	Revisar código en equipo y comparación ejemplos de Spike Prime.

GESTIÓN DE RIESGOS

En esta sección se realiza para identificar, analizar y planificar respuestas a eventos que podrían afectar negativamente los objetivos del proyecto. A continuación, se detallan los riesgos identificados, el nivel de su impacto y las acciones para solucionar esos tipos de eventos.

El Nivel de Impacto se definió utilizando una escala del 1 al 5, donde un valor más alto representa una amenaza más grave para el cumplimiento de los objetivos del proyecto:

- **1 (Bajo):** riesgo que tiene una solución sencilla o con un impacto mínimo en el proyecto.
- **3 (Medio):** riesgo que requiere tiempo y recursos adicionales, afectando el tiempo de trabajo.
- **5 (Alto):** riesgo que amenaza la viabilidad o el alcance del proyecto.

FUNDAMENTOS DE LOS MOVIMIENTOS

Esta sección justifica la configuración mecánica del brazo robótico mediante el uso de principios básicos de física, evaluando la velocidad y capacidad de carga de los motores LEGO SPIKE Prime.

Velocidad Angular – Rendimiento del Sistema

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

Tiempo máximo requerido: 3 segundos.

Velocidad angular requerida: 120°/s ≈ 20 RPM

Principio / fórmula utilizada:

Velocidad angular

$$\omega = \frac{\Delta\theta}{\Delta t} \quad \Bigg| \quad \omega = \frac{360^\circ}{3s} = 120^\circ/s \approx 20RPM$$

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

Principio / fórmula utilizada:

Trabajo mecánico contra la gravedad.

$$W = m \cdot g \cdot h \quad \Bigg| \quad W = 0.003 \cdot 9.8 \cdot 0.10 = 2.94mJ$$

Conclusión:

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

IDENTIFICACIÓN DE USUARIO Y CLIENTE

Usuario del sistema

Rol: Operador del sistema robótico.

Descripción: Persona encargada de operar y supervisar el brazo robótico mediante una interfaz de control (GUI o mando), asegurando la correcta ejecución de las tareas de carga y descarga simuladas durante pruebas y demostraciones.

Cliente del sistema

Rol: Entidad solicitante de la solución.

Descripción: En el contexto académico, corresponde al profesor a cargo del proyecto, quien define los objetivos, supervisa el desarrollo del prototipo y evalúa los resultados obtenidos. En un contexto industrial, el cliente sería la empresa interesada en automatizar sus procesos.

REQUERIMIENTOS FUNCIONALES

RF-01 – Control del sistema:

El usuario, a través del cliente de control externo, debe poder enviar comandos al sistema de control del brazo robótico para su operación manual.

RF-02 – Manipulación de la garra:

El cliente de control debe permitir al usuario ejecutar la apertura y cierre de la garra del brazo robótico para la manipulación de bloques.

RF-03 – Movimiento vertical del brazo:

El cliente de control debe permitir al usuario controlar la elevación y el descenso del brazo robótico, manteniendo estabilidad durante la operación.

RF-04 – Rotación del brazo:

El sistema del brazo robótico debe ejecutar movimientos de rotación controlada en respuesta a los comandos enviados desde el cliente.

RF-05 – Comunicación Cliente-Servidor:

El cliente de control debe establecer comunicación con el servidor implementado en el Hub SPIKE Prime para el envío y ejecución de comandos.

RF-06 – Estado de conexión:

El cliente de control debe informar al usuario el estado de la conexión con el brazo robótico.

REQUERIMIENTOS NO FUNCIONALES

RNF-01 – Confiabilidad

Conexión Bluetooth estable durante la operación del sistema.

Métrica: Sin pérdida de conexión durante la prueba funcional.

RNF-02 – Rendimiento (Latencia)

Respuesta oportuna del sistema a los comandos de control.

Métrica: Movimientos ejecutados sin retardos perceptibles.

RNF-03 – Precisión y Estabilidad

Ejecución de movimientos suaves y controlados.

Métrica: Ausencia de movimientos bruscos o caída de la carga.

RNF-04 – Usabilidad

Interfaz de control comprensible y fácil de operar.

Métrica: Operación exitosa tras una breve explicación inicial.

RNF-05 – Mantenibilidad

Código estructurado para facilitar ajustes y mejoras.

Métrica: Modificación de parámetros sin alterar la lógica principal.

RNF-06 – Seguridad Operativa

Operación segura durante la manipulación de la carga.

Métrica: Manipulación completa sin incidentes durante la prueba.

ARQUITECTURA DEL SISTEMA

El sistema utiliza una arquitectura Cliente-Servidor donde un computador actúa como cliente, enviando comandos desde una interfaz gráfica (GUI). El Hub LEGO Spike Prime funciona como servidor, procesando estas instrucciones mediante MicroPython para controlar con precisión los motores del brazo.

La comunicación se realiza vía Bluetooth, permitiendo el control inalámbrico en tiempo real de funciones como apertura, elevación y rotación. Este diseño elimina la necesidad de interacción física directa, cumpliendo con el objetivo crítico de mitigar riesgos humanos en entornos peligrosos como la minería



CLIENTE
(COMPUTADOR)

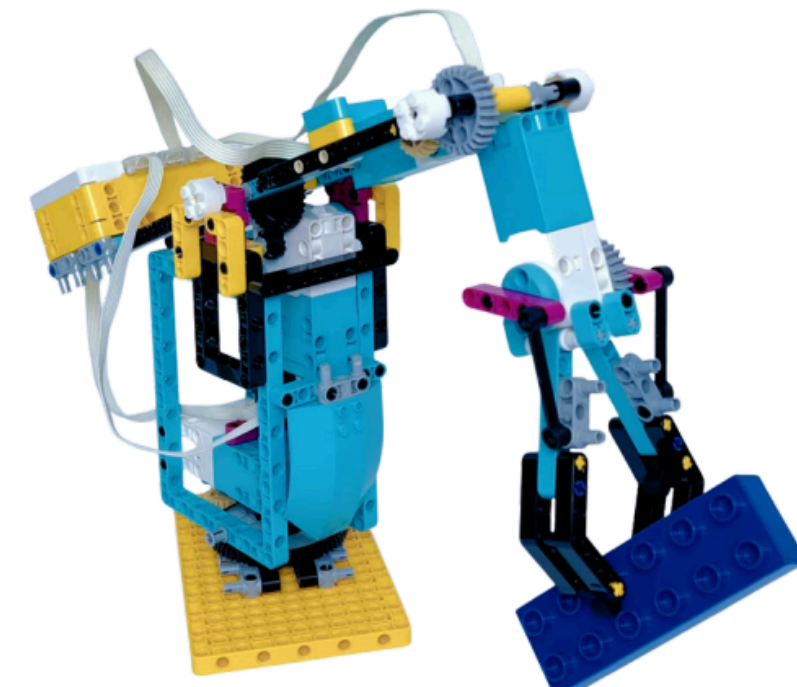
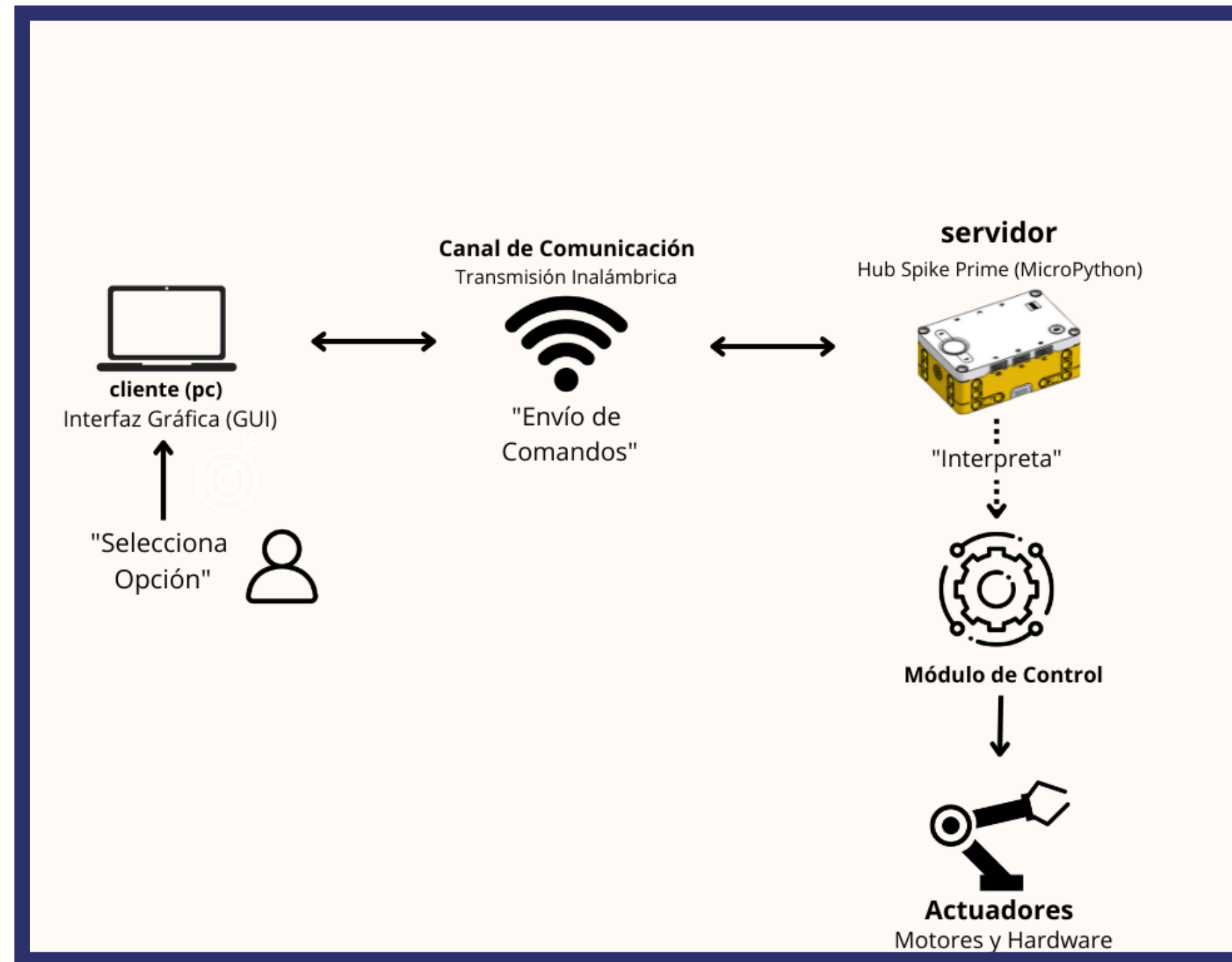
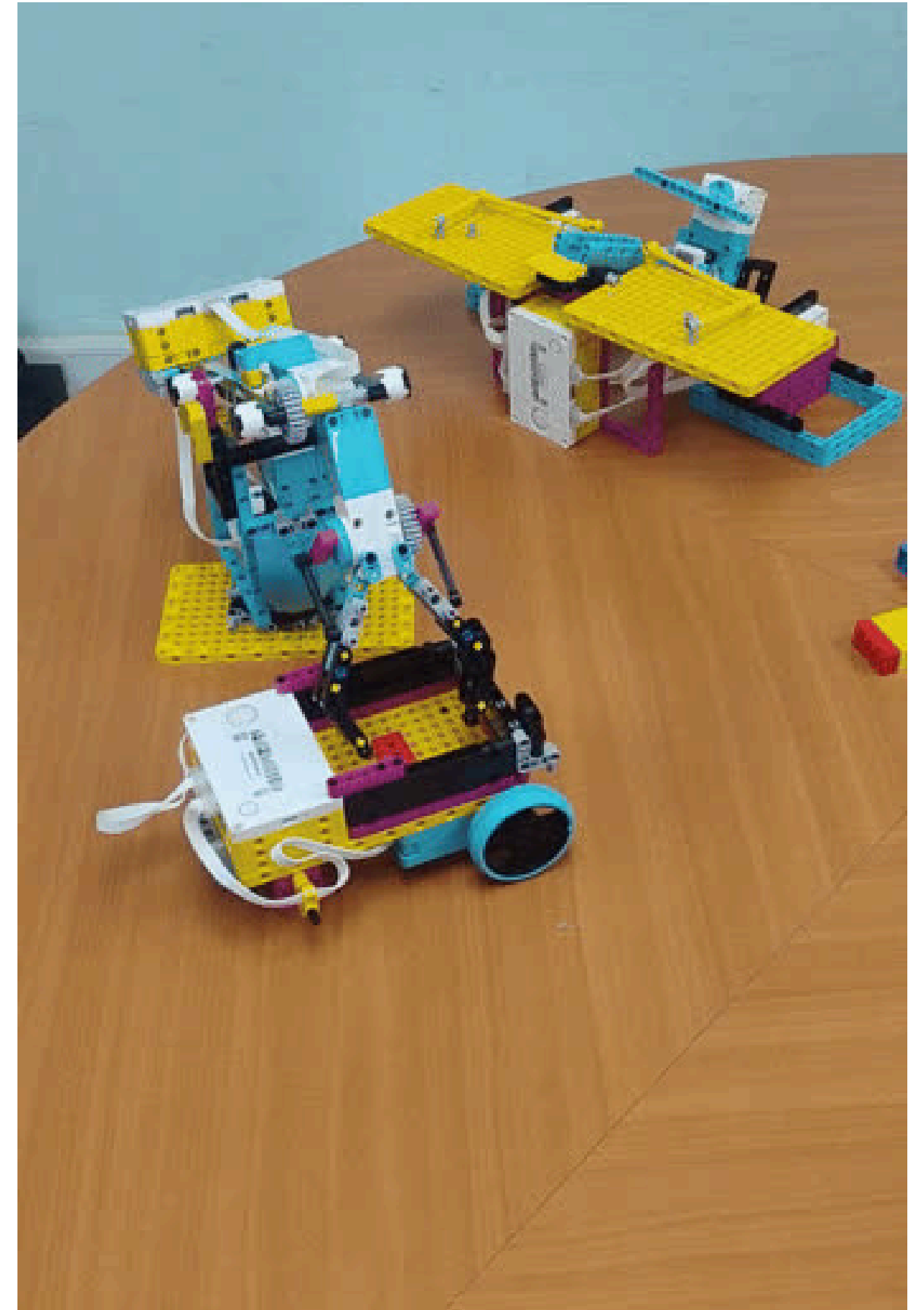


DIAGRAMA DE FLUJO



DEMO

El procedimiento comienza con el robot ubicado en una posición inicial. A partir de una orden enviada desde la interfaz, el sistema ejecuta una secuencia básica de operación, que incluye la apertura de la garra, el desplazamiento del brazo hacia el objeto, la sujeción del bloque y su traslado hasta el punto de descarga definido. Finalmente, el brazo libera la carga y completa el ciclo operativo.



REPOSITORIO

 **Proyecto-1** PublicPinWatch0

 main

 1 Branch

 1 Tag

 Add file

 Code

 **joaquinquezadaflores-cell**

se elimina Manual de usuario.txt por ser redundante con README.md

b0ab4ca · now

 25 Commits

 build/main	Se agrega ventana de seleccion de dispositivo y se actualiza ...	26 minutes ago
 dist	Se agrega ventana de seleccion de dispositivo y se actualiza ...	26 minutes ago
 README.md	Update usage instructions in README.md	13 minutes ago
 main.py	Se agrega ventana de seleccion de dispositivo y se actualiza ...	26 minutes ago
 main.spec	agregado el ejecutable	yesterday
 pseudocodigo	Correct motor control logic and variable names	last month

 README

Brazo Robótico de Agarre

Descripción del proyecto

RELEASE

Brazo Robótico de Agarre 1.0

Latest

Compare



joaquinquezadaflores-cell released this now



1.0



6831bc5

Descripción

Este programa capaz de mover una garra robótica construida con el set lego spike prime

OS

compatibilidad comprobada con Windows 11
se desconoce compatibilidad con otro OS

Requerimientos

- Windows 11
- set LEGO Spike Prime + paquete de expansión
- Python 3
- Pybricks
- Pybricksdev

Assets 3



Control.para.Brazo.Robotico.de.agarre.exe

sha256:72ad50d5455907d56069cbe5d4c56...



14.5 MB

6 min



Source code (zip)

8 min



Source code (tar.gz)

8 min



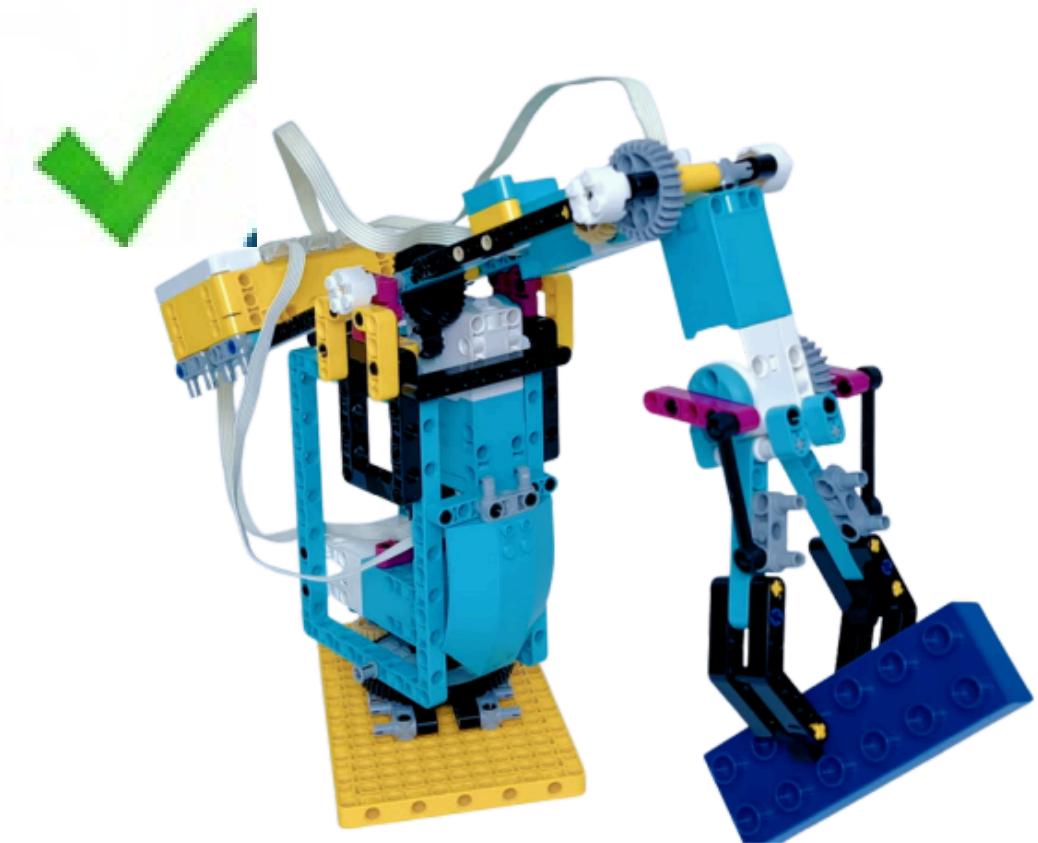
MANUAL DE USUARIO

Contenidos del README.md

- Descripción
- Autores
- Tecnologías Usadas
- Requisitos
- Instalación
- Release
- Uso básico

RESULTADOS OBTENIDOS

- **Funcionalidad Completa:** El sistema cumplió con el 100% de los requerimientos de movimiento, incluyendo apertura, cierre, elevación y rotación de 360°.
- **Estabilidad Mecánica:** El brazo robótico demostró un comportamiento estable y consistente, logrando una manipulación segura de los bloques durante todo el ciclo.
- **Control Inalámbrico Eficiente:** Se validó la arquitectura Cliente-Servidor vía Bluetooth, permitiendo una operación fluida desde la interfaz gráfica (GUI).
- **Mitigación de Riesgos:** Los hallazgos confirman la viabilidad del prototipo para reducir la intervención humana directa, alineándose con los estándares de seguridad industrial.
- **Éxito de Programación:** Se implementó correctamente la lógica en Python y MicroPython, superando los desafíos de comunicación y precisión iniciales.



LIMITACIONES ACTUALES

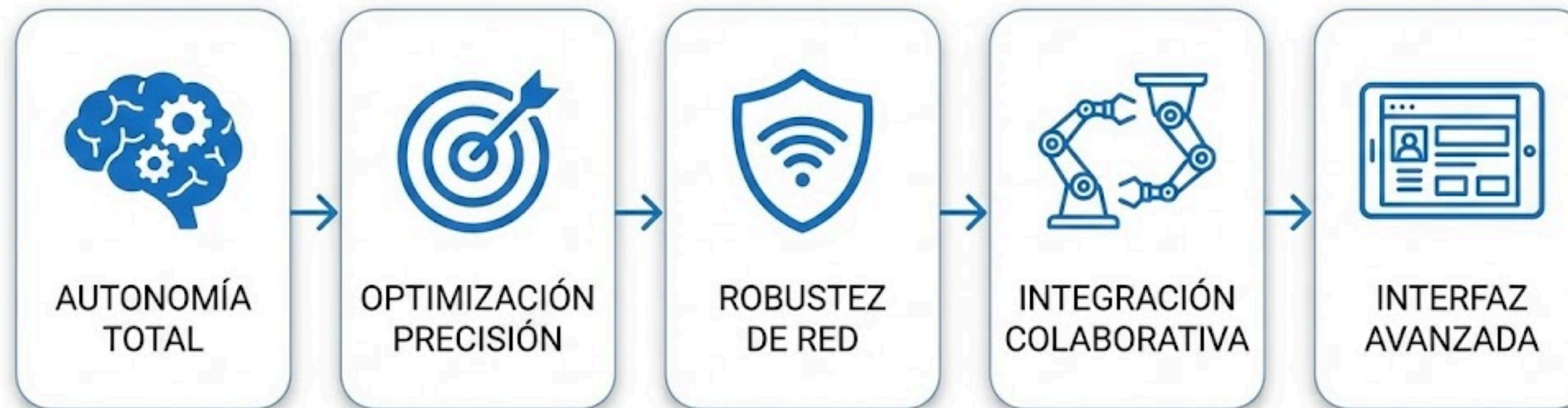
Limitaciones Actuales del Sistema

- **Limitada Autonomía (Control Manual Exclusivo):** El sistema opera únicamente mediante una interfaz de usuario (GUI) o un mando externo. La implementación de una secuencia de autonomía total aún está pendiente.
- **Dependencia de Recursos Externos (Procesamiento Gráfico):** El Hub presenta limitaciones de hardware que obligan a ejecutar el procesamiento gráfico y la interfaz en un dispositivo externo.
- **Necesidad de Calibración Fina (Precisión de Movimiento):** Se han observado ligeras desviaciones en los ángulos de rotación. Es indispensable realizar una calibración más precisa de los parámetros de los motores para corregir estas imprecisiones.



MEJORAS FUTURAS

- Autonomía Total: implementar secuencias de operación automática para eliminar la dependencia del control manual.
- Optimización de Precisión: calibrar algoritmos de movimiento para lograr desplazamientos más suaves y exactos.
- Robustez de Red: desarrollar protocolos de tolerancia a fallos para mitigar la latencia del Bluetooth.
- Interfaz Avanzada: finalizar la validación de la GUI para una experiencia de usuario más intuitiva y profesional.



CONCLUSIÓN

En conclusión, el proyecto ha logrado validar con éxito la integración de un brazo robótico estable y funcional, capaz de ejecutar ciclos completos de sujeción y rotación de 360° con una precisión posicional de 0.5 mm. A pesar de los desafíos iniciales en la conectividad Bluetooth y la escasez de piezas, la implementación de una arquitectura Cliente-Servidor en Python permitió superar las limitaciones del hardware, garantizando un control en tiempo real con una latencia inferior a 500 ms. Este desarrollo no solo cumple con los requerimientos funcionales establecidos, sino que sienta las bases para la futura integración sistémica con el vehículo autónomo, consolidando una solución robusta para la automatización de la cadena de transporte minera.



MUCHAS GRACIAS

Proyecto I

