UNIVERSIDAD DE TARAPACÁ

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA

Ingeniería@ Computación e Informática

Informe Fase 2
“Clasificador de colores LEGO”

Alumnos: Ariel Ortega
Constanza Serrano
Carla Flores
Jerry Quiroga
Martín Chávez

Asignatura: Proyecto I

Profesor: Baris Klobertanz

DICIEMBRE – 2025
ARICA - CHILE

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
10/10/2025	1.0	Creación del informe.	Carla Flores Jerry Quiroga
15/10/2025	1.1	Avances en el ítem 2.	Carla Flores Jerry Quiroga
17/10/2025	1.2	Avances en los ítems 3 y 4.	Carla Flores Jerry Quiroga
20/11/2025	1.3	Avances en ítem 1 y correcciones en el 4.	Carla Flores Jerry Quiroga
26/11/2025	1.4	Correcciones del informe en general.	Carla Flores Jerry Quiroga
27/11/2025	1.5	Correcciones enfocadas en los ítems 5 y 4.	Carla Flores Jerry Quiroga
27/11/2025	1.6	Últimas correcciones.	Carla Flores Jerry Quiroga
12/11/2025	1.7	Se agregaron las secciones: “Índice de tablas”, “Índice de figuras”, “Análisis y Diseño” e “Implementación”.	Jerry Quiroga
19/12/2025	1.8	Se agregó la sección “Prueba de funcionamiento de sistema”.	Jerry Quiroga Carla Flores
26/12/2025	1.9	Se realizaron las correcciones sugeridas por el profesor.	Constanza Serrano Jerry Quiroga Carla Flores

Índice de Contenidos

Índice de tablas

Índice de figuras

1. Panorama General

1.1. Introducción

En este semestre, nos embarcamos en un proyecto de ingeniería y programación que ilustrará nuestra capacidad de trabajo en equipo y el enfoque colaborativo para lograr los objetivos de la asignatura. Para cumplir con este desafío, emplearemos el kit educativo LEGO Education SPIKE Prime para desarrollar un robot que clasifique bloques de colores y los posicione en sus respectivos lugares, todo esto a través de una interfaz programada por el usuario en Python.

A lo largo de esta presentación, detallaremos la estructura de nuestro diseño y los avances clave logrados. Se expondrá la distribución de roles, la metodología de diseño y las estrategias de resolución de problemas aplicadas. Finalmente, compartiremos nuestras percepciones iniciales y la investigación teórica que fundamenta nuestras decisiones técnicas.

1.2. Objetivos

1.2.1. Objetivo General

Construir y programar un robot de Spike Prime que sea capaz de reconocer colores y distribuirlos en sus respectivos lugares mediante una interfaz de Python.

1.2.2. Objetivos Específicos

- Experimentar con el set de LEGO Education SPIKE Prime para la creación del robot.
- Ensamblar un modelo con buena movilidad, estabilidad, con un componente capaz de clasificar colores mediante un sensor de color.
- Estudiar la librería de tkinter para diseñar una interfaz gráfica apta para el usuario.

- Documentar el avance a lo largo del proyecto y las soluciones técnicas implementadas mediante el registro en bitácoras semanales.
- Coordinar el trabajo colaborativo mediante la distribución de roles y la planificación temporal para asegurar el cumplimiento de los plazos del proyecto.
- Estudiar el entorno de Python en la Spike Prime, junto con librerías externas, donde se investigará e implementará la API de Pybricks para la conexión con Visual Studio.

1.3. Restricciones

- Se debe programar sólo en Python.
- Solo se debe utilizar la plataforma Redmine para los documentos y avance del proyecto.
- Se debe utilizar el Set de LEGO Education SPIKE Prime .
- Limitación de tiempo para dedicar al proyecto.
- Cantidad de integrantes limitada a sólo 5.

1.4. Entregables

Bitácoras: Reportes semanales de progreso donde el equipo (a través de un miembro designado) detalla las actividades completadas, los obstáculos superados o pendientes, y las sugerencias de mejora. Estos informes son herramientas clave para la toma de decisiones estratégicas, la distribución de responsabilidades y la identificación de puntos a discutir en las reuniones grupales.

Carta Gantt: Herramienta visual de planificación que organiza las tareas del proyecto en una línea de tiempo, especificando su duración y orden. Su propósito es simplificar la gestión de tiempos y recursos al permitir una visualización clara del avance y la secuencia de las actividades.

Informe de Formulación: Documento estratégico inicial que establece la estructura y el plan del equipo para cumplir los objetivos de la asignatura. Incluye la definición de roles, las metas específicas del equipo, las metodologías a emplear, las impresiones iniciales sobre el proceso de desarrollo y la documentación esencial recopilada.

Presentaciones: Exposiciones concisas que resumen los objetivos del proyecto, los desafíos resueltos, las soluciones implementadas y los logros destacados. También sirven para presentar la estructura del equipo y ofrecer una visión general del diseño o funcionamiento del robot.

2. Organización del Personal

La organización de un grupo es fundamental para llevar a cabo el desarrollo de un trabajo. Con este fin, el equipo realizó sesiones de planificación donde se debatieron las responsabilidades necesarias y las habilidades técnicas de los integrantes, para finalmente asignar los roles mediante mutuo acuerdo. Esta distribución busca aprovechar la disposición y el compromiso de cada miembro para lograr una planificación eficiente y alcanzar el objetivo del proyecto.

2.1. Descripción de los Roles

Jefe de proyecto: Representante del equipo, supervisa y organiza la planificación para el progreso del proyecto. Además, es el encargado de realizar reuniones diarias con el fin de escuchar y atender a su equipo de trabajo.

Programador: Encargado de la codificación y el funcionamiento del robot, trabaja en colaboración con el ensamblador.

Ensamblador: Encargado de ensamblaje y armado de piezas, además de trabajar revisando el funcionamiento junto al programador.

Documentador: Encargado de registrar el avance del proyecto, junto con la redacción de los informes.

2.2. Personal que Cumplirá los Roles

Tabla 1: *Roles asignados al personal*

Rol	Responsable
Jefe de proyecto	Carla Flores
Programador	Ariel Ortega
Ensamblador	Jerry Quiroga Martin Chavez Constanza Serrano
Documentador	Constanza Serrano Carla Flores Jerry Quiroga

2.3. Métodos de Comunicación

Los principales métodos de comunicación que utilizaremos son los siguientes:

Comunicación Presencial: Será la forma principal de comunicación durante el horario de clases. Se utilizará para la resolución inmediata de problemas. Instagram: Se utilizará para la mensajería, utilizando la función de grupos que tiene la plataforma. Discord: Será usado para las reuniones, mediante sus canales de texto y voz.

3. Planificación del Proyecto

3.1. Actividades

Tabla 2: *Actividades*

Nombre	Descripción	Responsable	Producto	Objetivo Específico
Preparar el proyecto.	Se buscó qué tipo de modelos podíamos hacer.	Todo el grupo.	3 modelos a escoger: Garra, Vehículo, Separador de colores.	O.E 1
Elección del modelo.	Se escogió el modelo entre 3.	Todo el grupo	Se escogió el modelo: Separador de colores.	O.E 1
Primeros avances de la construcción del modelo.	Se construyó el motor del modelo en base a una guía ilustrada.	Jerry Quiroga Martin Chavez	Confección del motor que hará funcionar el modelo.	O.E 2
Búsqueda de información sobre la codificación.	Se buscó vía Google y foros información acerca de la codificación necesaria para hacer funcionar el modelo.	Carla Flores Ariel Ortega	Comprension basica de la codificación para el modelo	O.E 6
Avances en la construcción del modelo.	Se avanzó en la construcción de la base del modelo	Constanza Serrano	Se completa la construcción del modelo	O.E 2

	siguiendo la misma guía.		quedando en un 80%.	
Organización	Se distribuyeron las tareas.	Carla flores	Asignación definitiva de tareas respecto al proyecto.	O.E 4
Finalización de la construcción del modelo.	Se finalizó con éxito la construcción del modelo en su totalidad.	Constanza Serrano Carla Flores	Finalización de la construcción del modelo.	O.E 2
Avances en la codificación (bloques).	Se hicieron las primeras pruebas con el modelo ya finalizado.	Ariel Ortega Jerry Quiroga	Hecho con bloques de codificación, se logró que el modelo funcionara y cumpliera su propósito correctamente.	O.E 6
Inicio de codificación (Python).	Se comenzó a “traducir” la codificación de bloques a python.	Ariel Ortega	Primeras pruebas con código python, sin éxito.	O.E 6
Avances en codificación (python).	Se investigó acerca de la codificación en Spike Prime.	Ariel Ortega	Mayor conocimiento acerca de la programación necesaria para hacer que el modelo cumpla su propósito.	O.E 6

Confección del primer informe de avance.	Se realiza el primer informe de avance.	Carla Flores Jerry Quiroga	Primer informe de avance.	O.E 5
Término de producción de bitácoras de la etapa 1.	Se realizaron bitácoras semanales detallando avances y problemas.	Constanza Serrano Carla Flores Martin chavez	Bitácoras.	O.E 5

3.2 .Carta Gantt

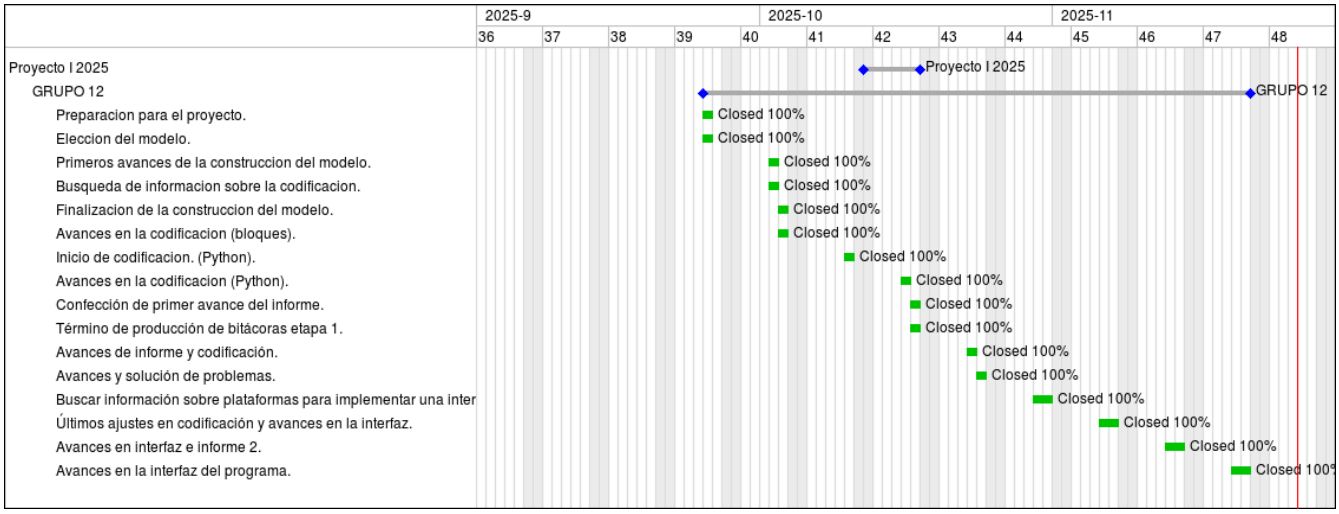

Figura 1: *Carta Gantt*

3.3. Gestión de riesgos

Se presenta a continuación una tabla que muestra un desglose de los problemas que se han presentado durante de la primera fase del proyecto. La tabla resume el impacto de cada desafío al clasificar el daño en cuatro niveles. Cada nivel está asociado con diferentes tipos de daño:

1. Daño catastrófico: Requiere medidas inmediatas, debido que puede provocar que el proyecto se pause totalmente o sufra retrasos muy significativos que obliguen a reiniciar etapas desde cero.

2. Daño Crítico: Requiere soluciones prioritarias, ya que tiene el potencial de retrasar el proyecto en varias de sus etapas consecutivas.

3. Daño circunstancial: Este riesgo debe resolverse puntualmente al momento de su aparición, dado que amenaza con retrasar el desarrollo de una etapa base del proyecto.

4. Daño menor: El riesgo no impacta en etapas principales ni en plazos de entrega; se trata de imprevistos de sencilla solución o problemas reiterativos que solo afectan la fluidez en las sesiones de trabajo.

Tabla 3: *Tabla de riesgos*

Riesgo	Probabilidad de ocurrencia	Nivel de impacto	Acción Remedial
Falta de documentación técnica	60%	2	Investigación en foros externos y migración a librerías con mejor soporte
Personal faltando al horario asignado de trabajo	70%	2	Implementación de trabajo asíncrono y reasignación de tareas críticas a miembros

			presentes.
Inestabilidad de versiones	30%	3	Congelamiento de la versión del firmware (evitar actualizaciones) una vez lograda la estabilidad operativa.
Dependencia del entorno (computadores)	10%	4	Designación de un computador único y dedicado para la ejecución del código en la presentación final.
Interferencia ambiental en el sensor	20%	4	Calibración de valores de luz al inicio del programa e instalación de barreras físicas (viseras) en el robot.
Integridad estructural y de montaje	10%	4	Revisión y etiquetado de cables antes de cada transporte; reforzamiento de barreras para contención de piezas.
Descoordinación por carga académica	80%	1	Realización de reuniones extraordinarias fuera del horario de clases para recuperar avances atrasados.

Desmontaje de batería para cargar	30%	4	Planificación de carga completa previa a las sesiones y verificación de la posición de cables post-carga.
Error en codificación	80%	1	Depuración colaborativa (pair programming) y pruebas unitarias por componente antes de la integración total.
Compatibilidad de interfaz gráfica	50%	2	Uso de Visual Studio Code vinculado a Pybricks para evitar limitaciones de las plataformas de bloques.
Dificultad en la conexión bluetooth	60%	3	Instalación de la API de Pybricks Dev que optimiza la conexión inalámbrica con el Hub SPIKE.
Dificultad en la conexión wifi	40%	3	Mantener cable USB disponible como método de conexión de respaldo (failsafe) inmediato.
Atraso en el cumplimiento de tareas	80%	1	Actualización constante de la Carta Gantt y priorización de tareas esenciales sobre estéticas.

4. Planificación de los Recursos

La planificación de recursos permite garantizar la disponibilidad de herramientas y componentes necesarios para el desarrollo del proyecto, abarcando recursos tanto físicos (hardware) como digitales (software). Esta etapa asegura que el robot pueda ser construido, programado y monitoreado de forma organizada, además de permitir la correcta gestión, documentación y coordinación de las actividades técnicas y administrativas del proyecto.

4.1. Hardware

- Set SPIKE™ Prime de LEGO® Education.
- LEGO® Education: Set de Expansión SPIKE™ Prime.
- Computador con el sistema operativo necesario para ser capaces de programar las instrucciones para el robot.

4.2. Software

- Sistema operativo windows, para programar las funciones del robot.
- Redmine, página para la organización del proyecto.
- Spike Prime, aplicación original del set de LEGO.
- Microsoft Office, para la documentación del proyecto, utilizando principalmente Word como herramienta.

4.3. Estimación costos

Costo Hardware

Tabla 4: *Costo Hardware*

Producto	Precio (CLP)	Cantidad
Set SPIKE™ Prime de LEGO® Education.	$ 383.196	1

LEGO® Education: Set de Expansión SPIKE™ Prime.	$ 150.375	1
Lenovo V14 G2 Ryzen 7 (x3)	$ 2.189.970 (729.990 c/u)	3
Total	$ 2.723.541	4

Costo Software

Tabla 5: *Costo Software*

Producto	Precio (CLP)
Licencia Microsoft Office	$ 14.000
Total:	$ 14.000

Costo de Trabajador

Tabla 6: *Costo de Trabajador*

Rol	Horas	Horas extra	Precio/Hora (CLP)
Jefe de Proyecto	20 horas	8 horas	$ 30.000
Programador	20 horas	6 horas	$ 27.000
Ensamblador	20 horas	0 horas	$ 27.000
Documentador	20 horas	12 horas	$ 25.000
Total	-	-	$ 2.882.000

Destacado:

- La contabilización de las horas trabajadas comienza a partir de la formación del grupo de trabajo (12/09/2025).
- Para la categorización de las horas de trabajo, se tuvo en cuenta el tiempo de trabajo en clases descritas en Intranet.
- Para la categorización de horas extras, se tuvo en cuenta las horas trabajadas fuera de horario de clase, independiente del lugar.
- Las 20 horas base por persona equivalen a la dedicación semanal estimada para cumplir las tareas mínimas del proyecto.
- Las tarifas por hora se basan en rangos reales de mercado para roles técnicos y de programación en Chile en 2025.
- Los costos por rol se calcularon como: (horas base + horas extra) × tarifa horaria. La suma de los subtotales da un costo total de $2.882.000 CLP.

Total de Costo:

Tabla 7: *Total de costo*

Costo de Hardware (CLP)	$ 2.723.541
Costo de Software (CLP)	$ 14.000
Costo de Empleados (CLP)	$ 2.882.000
Total	$ 5.619.541

5. Análisis y diseño

5.1. Especificación de requerimientos

5.1.1. Requerimientos funcionales

Los siguientes requerimientos funcionales describen las acciones que el sistema será capaz de realizar, considerando

la interacción entre el usuario, el sistema de control y el robot LEGO SPIKE Prime:

1. El robot debe identificar el color de cada bloque mediante un sensor de color integrado.

2. El robot debe recibir instrucciones desde una interfaz gráfica utilizada por el usuario para controlar su funcionamiento.

3. El robot debe clasificar cada bloque según su color, depositándolo en el compartimiento correspondiente.

4. El sistema debe permitir al usuario controlar el robot mediante una interfaz gráfica, la cual debe incluir cuatro botones de posición, un botón para abrir, un botón para cerrar y un botón de salida, desde los cuales se pueda dirigir manualmente el movimiento y las acciones del robot.

5. El robot debe coordinar el movimiento de sus motores para tomar el bloque desde el punto de lectura, transportarlo de manera controlada y depositarlo correctamente en el compartimiento asignado según su color.

5.1.2. Requerimientos no funcionales

Los siguientes atributos de calidad permiten evaluar el desempeño del sistema:

1. Disponibilidad: El robot debe permanecer operativo durante una sesión de trabajo continua de al menos 30 minutos, sin interrupciones causadas por batería insuficiente o desconexiones.
2. Rendimiento: El robot debe responder a las órdenes enviadas por la interfaz en menos de 1 segundo.
3. Usabilidad:Un usuario nuevo debe ser capaz de ejecutar las funciones básicas del sistema (mover el robot, abrir y cerrar el mecanismo, y finalizar la sesión) en menos de 5 minutos, sin necesidad de asistencia externa.

4. Robustez: El sistema debe manejar lecturas inválidas del sensor sin interrumpir la ejecución, aplicando reintentos o mensajes de alerta al usuario.

5.2. Arquitectura de Software

La arquitectura implementada corresponde a un modelo de control local con comunicación directa, donde:

- Interfaz de control: corresponde a la interfaz gráfica construida en Python, desde la cual el usuario envía instrucciones para controlar el movimiento y las acciones del robot.
-
- Robot: corresponde al LEGO SPIKE Prime programado mediante Pybricks, el cual recibe los comandos enviados desde la interfaz, ejecuta los movimientos, lee el sensor de color y realiza la acción correspondiente.

Flujo general de comunicación:

1. El usuario interactúa con la interfaz gráfica en Python.

2. La interfaz envía los comandos de control al robot LEGO SPIKE Prime mediante conexión Bluetooth.

3. El robot ejecuta el movimiento o acción solicitada.

4. El robot utiliza el sensor de color para identificar el bloque y continúa el proceso según la lógica definida.

Diagrama de arquitectura:

La siguiente figura muestra el diagrama correspondiente a la arquitectura del sistema, basado en un modelo de control local con comunicación directa entre la interfaz de usuario y el robot.

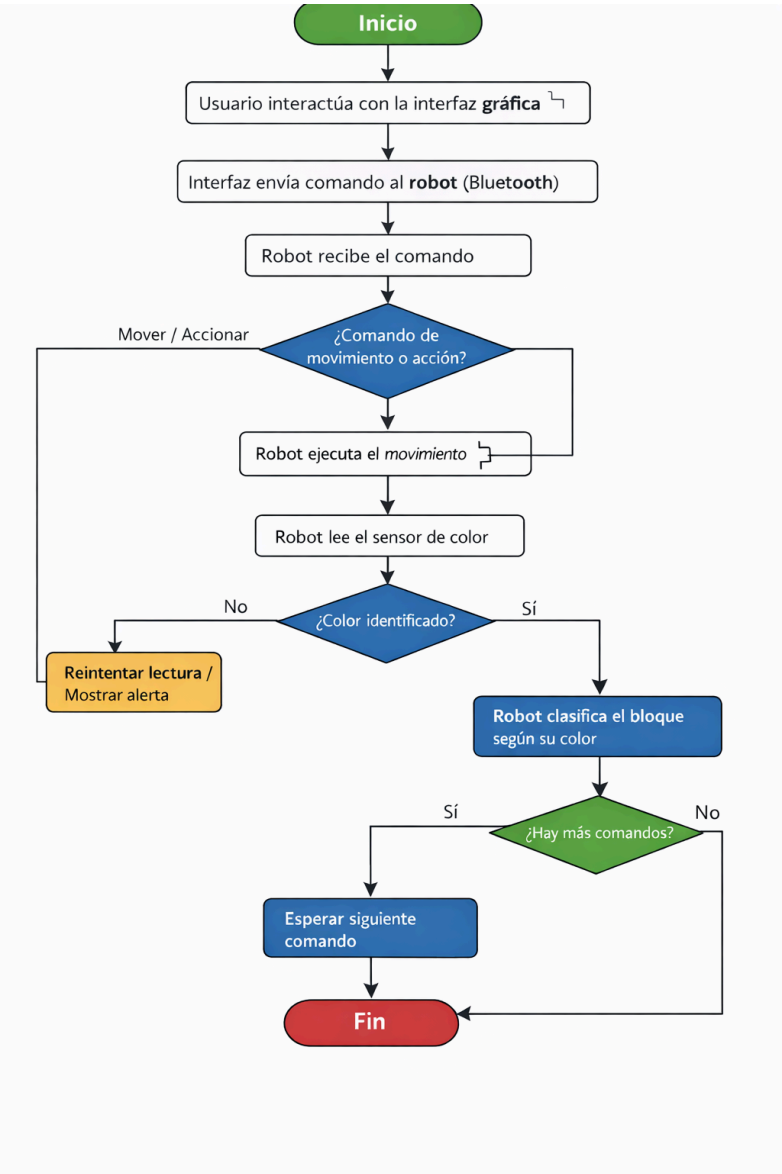

Figura 2: *Diagrama de arquitectura*

5.3. Diseño inicial de la interfaz gráfica de usuario (GUI)

El diseño preliminar de la interfaz se basó en un esquema simple que permite controlar el robot y visualizar su estado sin requerir conocimientos técnicos por parte del usuario. El wireframe de baja fidelidad incluye los siguientes elementos:

- 4 botones de posición.

- 1 botón iniciar según el color asignado.
- 1 botón para abrir la compuerta.
- 1 botón para cerrar la compuerta.
- 1 botón para salir

A continuación, se presenta el bosquejo inicial de la interfaz gráfica diseñada para el sistema:

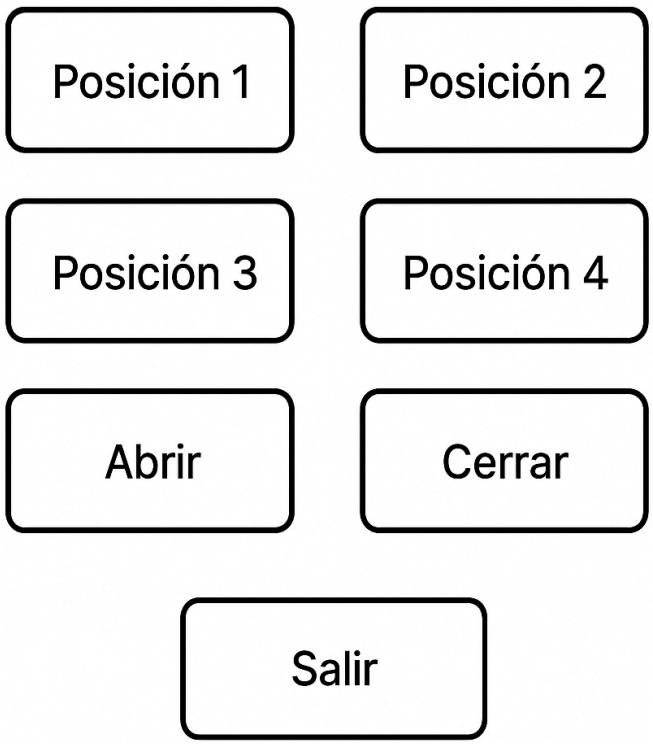

Figura 3: *Bosquejo de interfaz gráfica*

6. Implementación

6.1. Fundamentos de los movimientos

Para definir cómo debía moverse el robot, utilizamos principios básicos del movimiento rotacional. El motor es el que permite que el brazo gire hacia las distintas posiciones donde se deja cada color. Para cada posición, el robot debe girar un ángulo específico.

El cálculo general que usamos fue:

$$\theta = \omega \cdot t$$

Esto nos permitió estimar cuánto tiempo tardaría el brazo en llegar a cada punto según la velocidad del motor. Por ejemplo, si necesitábamos girar 90° y el motor gira a 360° por segundo, entonces el tiempo aproximado era de 0.25 segundos.

De esta forma se pudo definir tiempos y ángulos que fueran constantes y que permitieran que el robot se moviera de manera estable.

6.2. Descripción del sistema

6.2.1. Cliente

El cliente corresponde a la interfaz hecha en Python. Contiene lo siguiente:

- 4 botones de posición.
- 1 botón para iniciar según el color asignado.
- 1 botón para abrir la compuerta.
- 1 botón para cerrar la compuerta.
- 1 botón para salir.

Cada botón envía una instrucción al robot indicando qué acción debe realizar.

URL de GitHub:
https://github.com/sbmartin05/Proyectos1

Capturas del código.

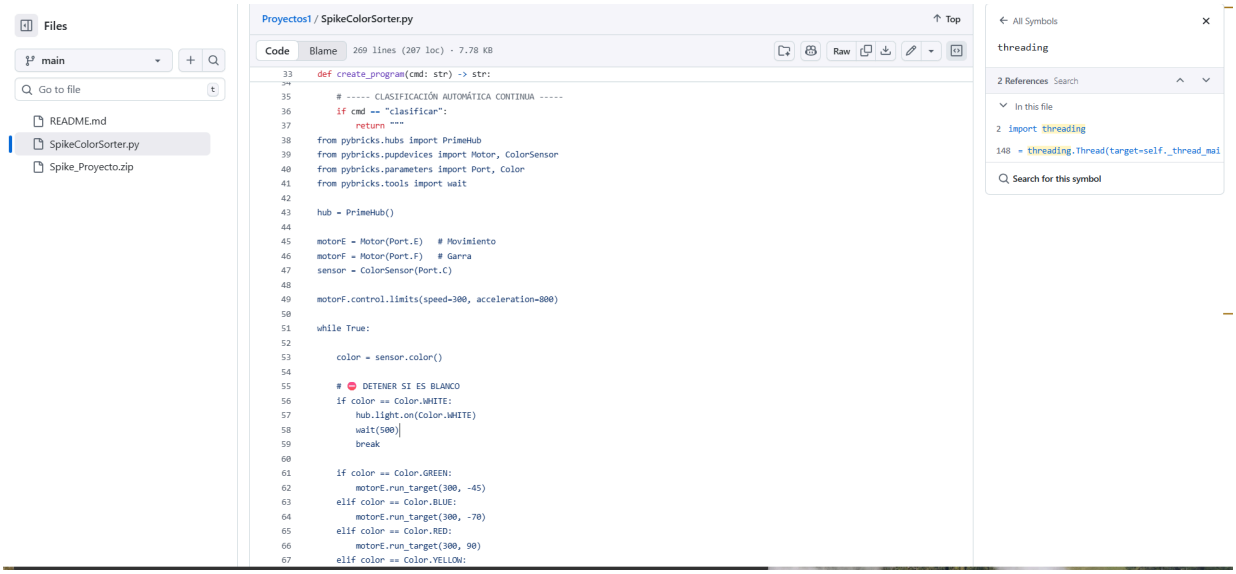

Figura 4: Primer captura de código.

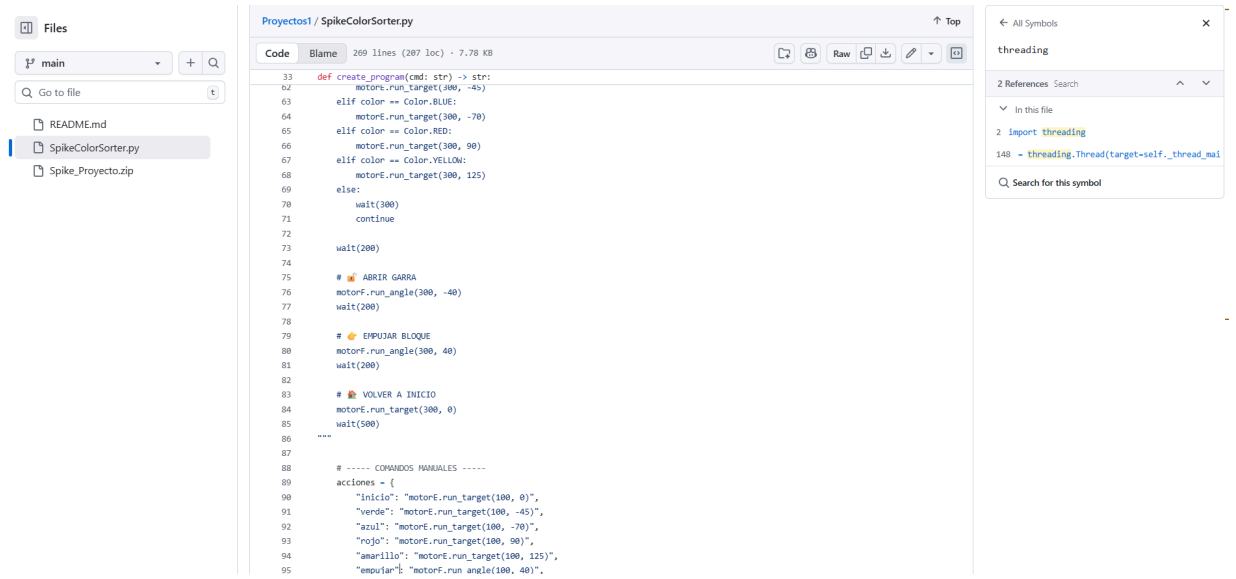

Figura 5: Segunda captura de código.

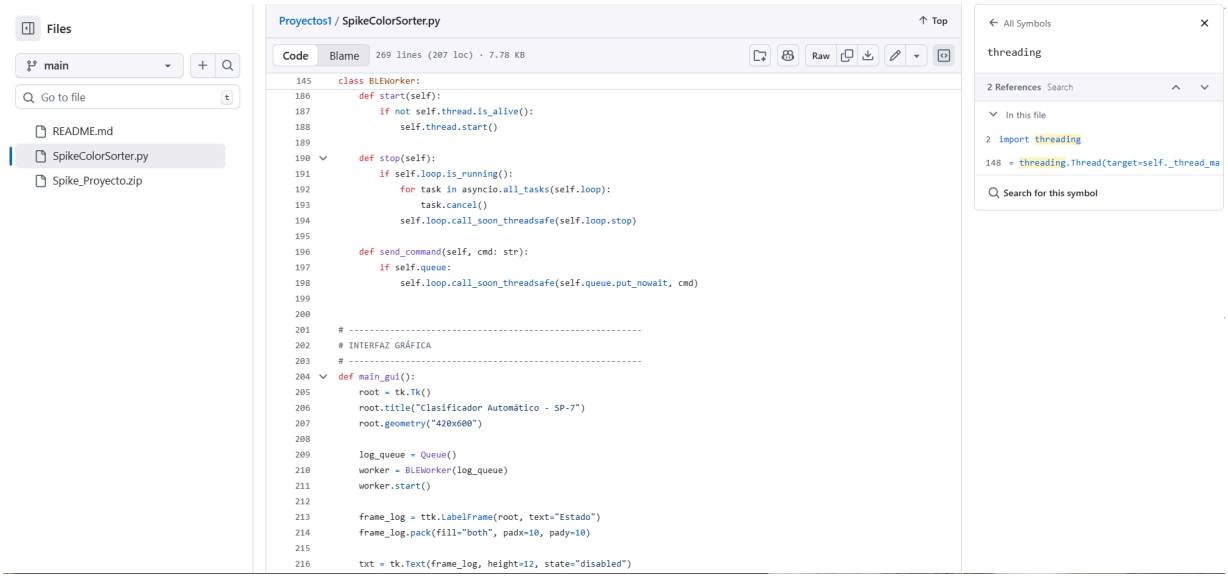

Figura 6: Tercer captura de código.

6.2.2.Servidor

El servidor es el controlador principal de SPIKE Prime y actua como el centro de operaciones del robot , El robot se programa en python, El controlador principal funciona con el software de Pybricks y es el encargado de ejecutar los movimientos y leer el sensor de color. Recibe las instrucciones enviadas desde la interfaz y las convierte en acciones concretas: mover el motor, abrir, cerrar o posicionarse en algún punto.

URL de GitHub:
https://github.com/sbmartin05/Proyectos1

6.2.3. Interfaz gráfica de usuario (GUI)

La interfaz está pensada para ser simple ya que solo contiene los botones necesarios para mover el robot y controlar el mecanismo. Cada botón está asociado a una función que envía la instrucción al dispositivo.

El objetivo fue que cualquier usuario pudiera entenderla sin dificultad y poder usarla sin necesidad de conocer o entender el código.

Captura de GUI implementada

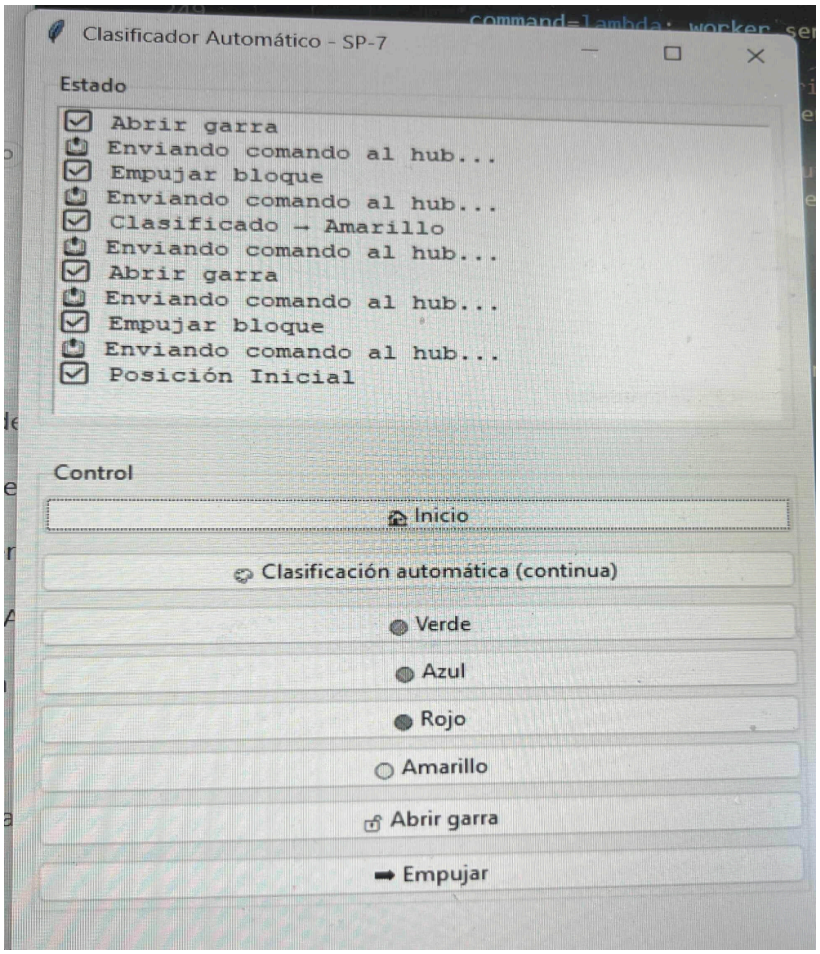

Figura 7: Captura de GUI implementada.

7. Resultados

7.1. Estado actual del proyecto

Hasta el momento, el sistema cumple con los requerimientos funcionales definidos para el proyecto:

- RF1 (Identificación del color de los bloques): Cumplido.

- El robot es capaz de identificar el color de los bloques mediante el sensor integrado.

- RF2 (Recepción de instrucciones desde la interfaz): Cumplido.
- Se ha establecido comunicación entre la interfaz gráfica en Python y el robot LEGO SPIKE Prime, permitiendo el envío de comandos.

- RF3 (Clasificación de bloques según su color): Cumplido.
- El robot clasifica los bloques de acuerdo con el color detectado, depositándolos en el compartimiento correspondiente.

- RF4 (Control manual del robot mediante la interfaz): Cumplido.
- El usuario puede controlar manualmente el robot utilizando los botones de movimiento, abrir, cerrar y salida.

- RF5 (Movimiento coordinado de motores): Cumplido.
- El robot ejecuta de manera correcta los movimientos necesarios para trasladar y clasificar los bloques.

Actualmente, el sistema se encuentra funcional y operativo.

Como trabajo pendiente, solo resta mejorar la interfaz gráfica, principalmente en aspectos visuales y de usabilidad, sin afectar el funcionamiento del sistema.

7.2. Problemas encontrados y solucionados

Durante el desarrollo surgieron varios problemas:

1. La conexión entre Python y SPIKE Prime no funcionaba bien.

Se solucionó instalando Pybricks y usando una versión más estable del firmware.

2. El motor no llegaba siempre al mismo punto.

   Se ajustaron los ángulos y se probaron bien antes de usarlo.

3. Los cables estorbaban en algunas ocasiones los movimientos del robot.

   Se solucionó poniendo sujetadores de cable en posiciones específicas que permitían el movimiento correcto del robot.

9. Conclusión

El proyecto demostró que, mediante el kit educativo LEGO Education SPIKE Prime y el lenguaje de programación Python, es posible diseñar un prototipo funcional para el reconocimiento y clasificación de colores. Aunque la primera transición desde programación por bloques a código Python no fue exitosa, el problema fue resuelto mediante investigación y depuración del código, lo que permitió validar el correcto funcionamiento del robot. La distribución de roles, la comunicación constante y el uso de la plataforma de gestión de proyectos Redmine, junto con pruebas incrementales del software controlador, fueron claves para avanzar de forma organizada y colaborativa. Como trabajo futuro, se propone optimizar el código, incorporar más pruebas automatizadas y mejorar la precisión del reconocimiento de colores en distintos escenarios. En resumen, el proyecto permitió integrar hardware, software y coordinación en equipo para cumplir el objetivo inicial de la asignatura.

9. Referencias

LEGO. (s. f.). Set básico LEGO SPIKE Prime. https://surl.li/fpvege

LEGO. (s. f.). Set de expansión LEGO SPIKE Prime. https://surl.li/ensdyc

OpcStore. (s. f.). Notebook V14 G2 ALC Ryzen 7. https://surl.lt/gchmtv

Computrabajo. (2025). Salarios de programador en 2025. https://n9.cl/wqp5nu

Computrabajo. (2025). Salarios de desarrollador programador en 2025. https://n9.cl/l9c8qt

Chiletrabajos. (2025). Sueldo promedio de programador junior: $715.336 (noviembre 2025). https://n9.cl/33lou