

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

**DEPARTAMENTO DE INGENIERÍA CIVIL EN COMPUTACIÓN E
INFORMÁTICA**



**Sorting Mining: Clasificador Remoto de
Minerales para la Optimización y Seguridad
en Minería.**

**Alumnos: Jose Quispe
Camilo Geraldo
Ignacio Cuevas
Maximiliano Burgos
Jose Quispe
Matias Sagredo**

Asignatura: Proyecto 1

**Profesor: Baris Nikolai Klobertanz
Quiroz**

Historial De Cambios

Fecha	Versión	Descripción	Autor(es)
27/09/2025	1.0	Concepción del Documento	Camilo Geraldo
03/10/2025	1.1	Bosquejo general del Informe y planteo de objetivos.	Camilo Geraldo Jose Quispe
08/10/2025	1.2	Corrección de la introducción y del objetivo general y específicos.	Jose Quispe
10/10/2025	1.3	Adición del prototipo en los entregables, cambio de nombres en el punto 2.2	Todos
13/10/2025	1.4	Corrección de la carta gantt, cambio en el nivel de gestión de riesgos y de posibles riesgos	Matias Sagredo Camilo Geraldo
14/10/2025	1.5	Estimación de costos de los recursos de hardware	Ignacio Cuevas
15/10/2025	1.6	Cambio en las actividades del punto 3.1 y eliminación de la columna responsable del punto 2.2	Jose Quispe Ignacio Cuevas
17/10/2025	1.7	Redacción de la conclusión	Jose Quispe
24/11/2025	1.8	Corrección general del documento	Camilo Geraldo Ignacio Cuevas Maximiliano Burgos Jose Quispe Matias Sagredo
10/12/2025	1.9	Inicio Fase 2 del informe	Camilo Geraldo Ignacio Cuevas Maximiliano Burgos Jose Quispe Matias Sagredo
12/12/25	2.0	Redacción de los puntos 5, 6 y 7	Camilo Geraldo Ignacio Cuevas Maximiliano Burgos Jose Quispe Matias Sagredo

Tabla de Contenidos

Historial De Cambios.....	1
Tabla de Contenidos.....	2
Índice de Tablas.....	3
Índice de figuras.....	3
1. Panel General.....	4
1.1. Introducción.....	4
1.2. Objetivos.....	5
1.2.1. Objetivo General.....	5
1.2.2. Objetivos Específicos.....	5
1.3. Restricciones.....	6
1.4. Entregables.....	6
2. Organización del Personal.....	7
2.1. Descripción de los Roles.....	7
2.2. Personal que Cumplirá los Roles.....	8
2.3. Métodos de Comunicación.....	8
3. Planificación del Proyecto.....	9
3.1. Actividades.....	9
3.2. Carta Gantt.....	11
3.3. Gestión de Riesgos.....	12
4. Planificación de los Recursos.....	13
4.1. Hardware.....	13
4.2. Software.....	13
4.3. Estimación de Costos.....	14
5. Análisis y Diseño.....	16
5.1. Especificación de Requerimientos.....	16
5.1.1. Requerimientos funcionales.....	16
5.1.2. Requerimientos no funcionales.....	16
5.2. Arquitectura de Software.....	17
5.2.1. Definición de Componentes.....	17
5.2.2. Diagrama de Arquitectura.....	17
5.3. Diseño inicial de la interfaz gráfica de usuario (GUI).....	18
6. Implementación:.....	19
6.1. Fundamentos de los movimientos:.....	19
6.2. Descripción del sistema.....	21
6.2.1. Cliente.....	21
6.2.2. Servidor.....	24
6.2.3. Interfaz gráfica de usuario (GUI).....	27
7. Resultados.....	29
7.1. Estado actual del proyecto.....	29
7.2. Problemas encontrados y solucionados.....	30
8. Prueba de funcionamiento.....	31
8.1 Descripción de la prueba de funcionamiento.....	31
8.2 Resultados observados para la prueba de funcionamiento.....	31
9. Conclusión.....	33

10. Referencias.....	34
-----------------------------	-----------

Índice de Tablas

Tabla N° 1. Designación de roles para los integrantes del equipo.....	8
Tabla N° 2. Establecimiento de actividades para el desarrollo del proyecto.....	10
Tabla N° 3. Tipos de daño.....	12
Tabla N° 4. Riesgos posibles en el desarrollo del proyecto.....	13
Tabla N° 5. Costos de hardware.....	14
Tabla N° 6. Estimación del costo del trabajador.....	15
Tabla N° 7. Total de costos.....	15

Índice de figuras

Figura n° 1: "Carta Gantt".....	11
Figura n°2: "Diagrama de arquitectura"	17
Figura n°3: "Wireframe GUI"	18
Figura n°4: "Modelo de Motor Principal"	19
Figura n°5: "Modelo de Motores Secundarios"	20
Figura n°6: "Clase GUI"	21
Figura n°7: "Implementación de los botones principales"	22
Figura n°8: "Implementación de solicitud de conexión cliente-servidor"	22
Figura n°9: "Clase BLEWorker"	23
Figura n°10: "Implementación de la conexión a Bluetooth"	23
Figura n°11: "Implementación de función para ejecutar comandos"	24
Figura n°12: "Interfaz gráfica del prototipo robótico"	25

1. Panel General

1.1. Introducción

La industria minera en Chile representa uno de los pilares económicos más importantes del país, aportando más del 10% del PIB y generando empleo directo para 280 mil personas y unos 900 mil puestos laborales en total [10]. Al ser un área tan importante para el crecimiento y progreso del país, es crucial la implementación de tecnologías actuales para la optimización de los procesos de extracción, refinamiento y transporte de minerales.

Por lo que se refiere como Minería 4.0 a la transición de la minería tradicional hacia un modelo digitalizado y automatizado, impulsado por las tecnologías emergentes como el internet, la inteligencia artificial, la robótica, el Big Data y la realidad aumentada.

Un ejemplo de esta nueva minería, es la empresa minera Antofagasta Minerals [7][8], la cual, en colaboración con otras instituciones, innova en tecnologías para la teleoperación remota de maquinaria pesada o que conlleva un cierto grado de riesgo para los trabajadores.

La implementación de estas nuevas tecnologías busca mejorar la productividad, reducir costos, optimizar procesos, aumentar los estándares de seguridad y mantener la competitividad de las industrias a nivel global. De los puntos presentados, el que impulsa el desarrollo del proyecto es la seguridad.

Mantener la seguridad de los trabajadores mineros ha sido un problema que ha existido desde el comienzo de esta noble labor, ya que conlleva la manipulación de elementos explosivos de gran potencia, el manejo de maquinaria de carga de alto tonelaje y procesos de tratado de minerales con sustancias químicas nocivas para la salud.

Desde este punto de vista, el proyecto tiene la finalidad de desarrollar e implementar un gemelo físico o maqueta robótica de la maquinaria empleada en la industria minera y automatizarla para que un usuario pueda operar dicho prototipo de forma remota, simulando así una etapa del proceso minero.

En este informe, se documentó el trabajo finalizado durante el segundo semestre de la asignatura Proyecto I. La propuesta al problema de la seguridad en la industria minera, consistió en implementar un prototipo robótico automatizado utilizando el kit LEGO Spike Prime; dicho prototipo simulará el proceso de clasificación de materiales en la industria minera. El prototipo deberá ser capaz de identificar diferentes tipos de materiales representados por piezas de colores y ordenarlos en contenedores designados de manera autónoma o manual.

Durante el desarrollo del proyecto, se abordaron distintas etapas del proceso de diseño e implementación, incluyendo la planificación inicial, el diseño mecánico del prototipo, la programación del sistema de control y la validación del funcionamiento a través de pruebas prácticas.

Cabe destacar que el proyecto se desarrolló bajo restricciones propias de un entorno académico, utilizando el kit LEGO Spike Prime como una plataforma experimental, de la cual no hay mucha documentación relacionada con el proyecto que se lleva a cabo, lo que implica trabajar en una escala reducida, con recursos limitados y centrado exclusivamente en la simulación del proceso minero de clasificación de minerales.

1.2. Objetivos

1.2.1. Objetivo General

Implementar un prototipo robótico utilizando el kit LEGO Spike Prime, que simule el proceso de clasificación de materiales en la industria minera, siendo capaz de identificar diferentes tipos de materiales representados por piezas de colores y ordenarlos en contenedores designados de manera autónoma o manual por los trabajadores mineros.

1.2.2. Objetivos Específicos

- Diseñar y ensamblar en 3 semanas una estructura de prototipo con LEGO Spike Prime que incluya mecanismos de entrada y salida de piezas.
- Programar el prototipo para identificar y clasificar piezas por color de forma autónoma a través del sensor de color del set LEGO Spike Prime en un lapso de 5 semanas.
- Diseñar una interfaz de usuario tkinter capaz de enviar comandos inalámbricos al prototipo, permitiendo al menos 4 acciones de control manual.
- Optimizar el prototipo robótico, realizando mejoras en software y estructura, hasta lograr al menos un 98% de precisión en pruebas controladas tanto del modo automático como manual.
- Elaborar en 2 meses un informe técnico de al menos 10 páginas y una presentación preliminar de mínimo 10 diapositivas que explican el diseño, funcionamiento y resultados del prototipo.

1.3. Restricciones

- o Sólo se debe utilizar la plataforma Redmine para la presentación de los documentos y avance del proyecto.
- o Se debe utilizar el Set de Lego Spike Prime.
- o Limitación de tiempo para dedicar al proyecto.
- o Cantidad de integrantes por equipo limitado a sólo 5.
- o Disponibilidad del prototipo para codificar y probar.
- o El prototipo debe ser capaz de clasificar los materiales de forma manual.

1.4. Entregables

Bitácoras: Son informes semanales que describen el avance del equipo en el proyecto, abarcando actividades realizadas, dificultades encontradas, recomendaciones para mejorar y acciones tomadas. Preparadas por un individuo designado, ofrecen un panorama exhaustivo para apoyar decisiones estratégicas, asignan responsabilidades y resaltan asuntos a tratar en grupo.

Carta Gantt: Representación visual de la programación del proyecto, mostrando en una línea de tiempo las tareas, su duración y secuencia, facilitando la gestión del tiempo y los recursos al visualizar la evolución de las actividades a lo largo del proyecto.

Informe de Formulación: Este documento detalla la organización y estrategia que se siguió para alcanzar los objetivos de la asignatura. Abordaremos la asignación de roles, las metas del equipo y las medidas que implementaremos para lograr el propósito académico. Además, compartiremos nuestras primeras impresiones durante el proceso de desarrollo y presentaremos la documentación relevante recopilada a lo largo del semestre.

Presentaciones: Se detallan los objetivos del proyecto, los retos superados y las soluciones aplicadas. También se resaltan los éxitos obtenidos, la distribución del equipo y se ofrece una visión general del prototipo.

prototipo: Es el objetivo principal de este proyecto, es un prototipo robótico utilizando el kit LEGO Spike Prime, que simula el proceso de clasificación de materiales en la industria minera.

Informe Final: Este documento detalla todo el desarrollo del proyecto, incluyendo la metodología que utilizamos, resultados que se obtuvieron, desempeño del prototipo y conclusiones.

Código Fuente: Repositorio de GitHub que contendrá el código del cliente desarrollado para el control autónomo y manual del prototipo.

2. Organización del Personal

La organización en un grupo es esencial para el desarrollo de un trabajo, y para ello, en base a las necesidades para el proyecto se necesitará una distribución del trabajo adecuada para lograr el objetivo del proyecto.

2.1. Descripción de los Roles

La organización de los roles se definieron considerando las etapas más importantes del proyecto que son Diseño, Construcción, Programación, Documentación y Gestión. Además se tomó en cuenta las competencias individuales de cada uno de los integrantes del proyecto, asegurando una distribución equilibrada de responsabilidades y una cobertura de todas las áreas necesarias para el éxito del proyecto.

Jefe de proyecto: Representante del equipo, supervisa y organiza el progreso de cada rol y del proyecto en general.

Ensamblador: Encargado del montaje y el armado de las piezas, monitorea el cumplimiento de las funcionalidades del prototipo, en conjunto con el programador.

Programador: Encargado del área de la codificación y funcionamiento del prototipo, en colaboración del ensamblador.

Documentador: Encargado de tener un registro del avance del proyecto, junto con la redacción de los informes y de la Carta Gantt.

Diseñador: Encargado de la creación del logotipo, la estética y la presentación del proyecto, complementando al ensamblador en cuanto el diseño del prototipo.

2.2. Personal que Cumplirá los Roles

Rol	Responsable
Jefe de proyecto	Maximiliano Burgos
Ensamblador	Matias Sagredo
Diseñador	Jose Quispe
Programador	Camilo Geraldo
Documentador	Ignacio Cuevas

Tabla N°1. Designación de roles para los integrantes del equipo.

2.3. Métodos de Comunicación

Los medios de comunicación que se usaron para el desarrollo del informe fueron los siguientes:

WhatsApp, se utilizó para la mensajería, haciendo uso de los grupos que ofrece la plataforma.

Discord, fue empleado como servicio de reuniones, aprovechando sus canales de texto y voz.

3. Planificación del Proyecto

3.1. Actividades

Nombre	Descripción	Responsables	Resultado
Realizar la introducción y exploración de las funcionalidades del set Lego Spike Prime	Se realizó una introducción del set Lego Spike y sus funcionalidades	Todo el grupo.	Comprensión del uso del HUB, motores y sensores
Planificar la estructura base del proyecto	Planificación de roles y asignación.	Camilo Geraldo	Definición del nombre del proyecto y asignación de roles
Redactar las bitacora semanalmente como registro del avance del proyecto	Se realizaron bitácoras semanales detallando el avance del proyecto.	Matias Sagredo	Bitácoras que sirven como registro de lo que se ha realizado durante la semana.
Diseñar el prototipo basándose en guías	Se busca un modelo de prueba del prototipo, siguiendo una guía básica.	Todo el grupo.	Se establece la forma que tendrá el prototipo
Ensamblar el prototipo	Armado del prototipo empleando una guía.	Matias Sagredo Maximiliano Burgos	Concepción base del modelo del prototipo.
Programar el código en automático del clasificador	Comienzo del código para hacer funcionar al prototipo	Camilo Geraldo Maximiliano Burgos	Se logra el control automático mediante programación por bloques
Redactar el avance de la primera parte del informe	Se realiza el primer informe de avance.	Todo el grupo.	Tener un primer registro del avance del proyecto.
Avanzar con las diapositivas presentación	A base del informe, se realizará la presentación.	Todo el grupo.	Primera presentación.

Probar y optimizar el clasificador con diferentes piezas	Se realizarán pruebas para ver la eficiencia del clasificador	Jose Quispe Camilo Geraldo Ignacio Cuevas	Detectar posibles fallas de programación o estructurales o casos particulares.
Diseñar una interfaz de usuario para el manejo manual del prototipo	Diseñar una interfaz sencilla y atractiva para el manejo manual del prototipo.	Jose Quispe Maximiliano Burgos Camilo Geraldo	La funcionalidad del prototipo pasa de ser automática a manual.
Enlazar la interfaz de usuario con el prototipo.	Conectar la interfaz de usuario con el código del prototipo.	Camilo Geraldo Jose Quispe	Establecer la función manual para que el usuario opere el prototipo de forma remota.
Probar y optimizar la funcionalidad manual para la detección de fallas.	Probar si la interfaz se sincroniza correctamente con el prototipo.	Matias Sagredo Jose Quispe	Detectar posibles fallas en la sincronización con el prototipo y establecer los límites de conexión
Redactar el avance de la segunda parte del informe	Realizar el segundo informe ya finalizando el proyecto.	Todo el grupo	Recopilar los últimos datos
Presentación final de proyecto	Presentar el prototipo terminado.	Todo el grupo	

Tabla N°2. Establecimiento de actividades para el desarrollo del proyecto.

3.2. Carta Gantt

Para ayudar a mantener una planificación temporal clara y ordenada se ha elaborado la siguiente Carta Gantt. Esta nos permite gestionar el cronograma de actividades, mostrando todas las tareas programadas, la duración y el progreso alcanzado en cada una de ellas, haciendo más fácil el seguimiento y la gestión del avance a lo largo del semestre.

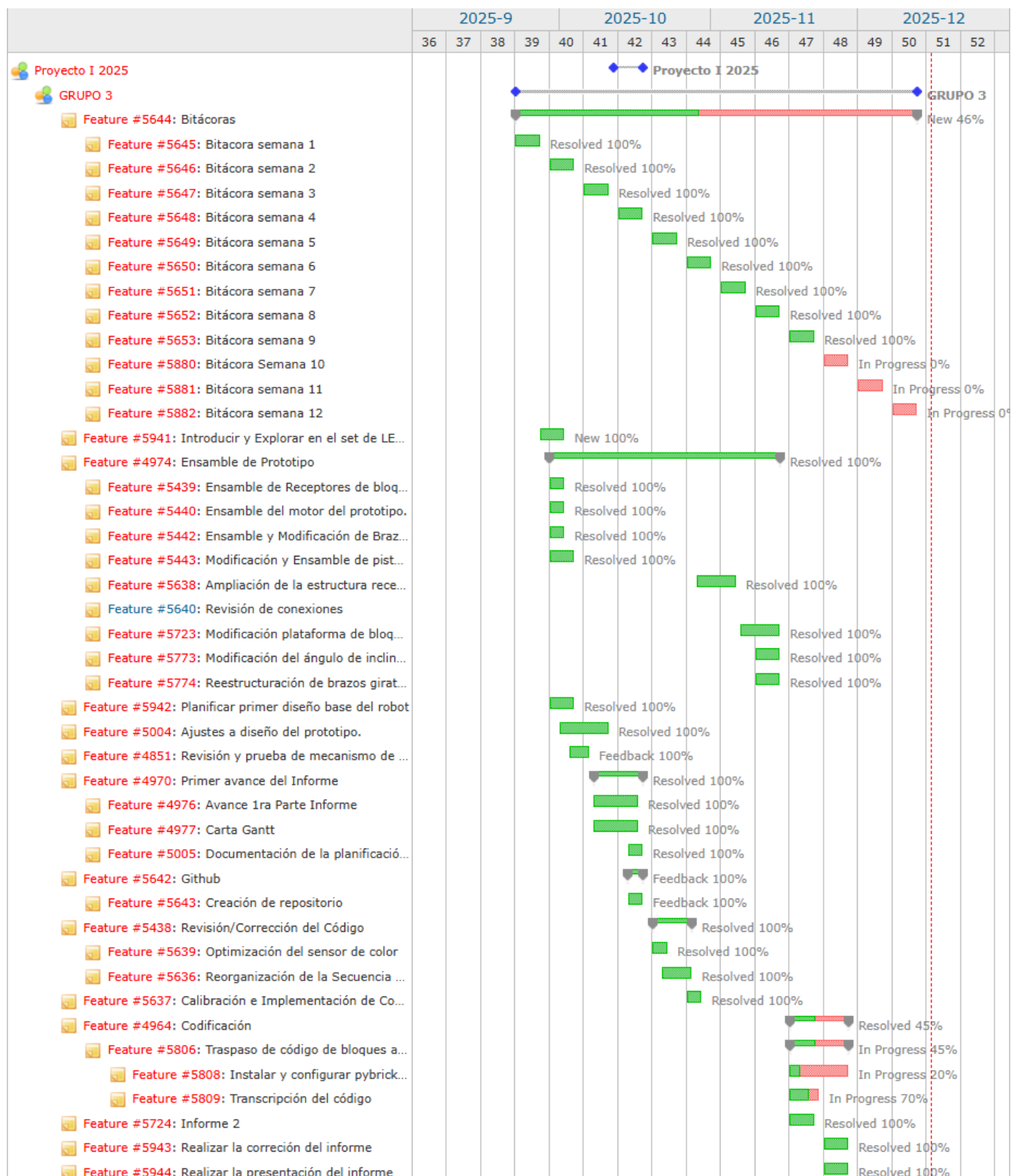


Figura nº 1: "Carta Gantt"

3.3. Gestión de Riesgos

Se presenta a continuación una tabla que exhibe un desglose de los problemas que se han presentado a lo largo del proyecto. Esta tabla resume el impacto de cada desafío al clasificar el daño en cinco niveles distintos. Cada nivel está asociado con diferentes tipos de daño:

Daño	Descripción
Recurrente	Riesgo no significativo, pero es reiterativo, retrasa en las sesiones de trabajo, pero no en etapas.
Irrelevante	Riesgo de no mayor importancia, es un detalle imprevisto que no necesita mucha atención y se puede resolver en cualquier momento.
Circunstancial	Riesgo que se debe resolver en el momento, debido a que puede retrasar el desarrollo de una etapa base del proyecto.
Crítico	Se deben tomar medidas necesarias para resolver el riesgo, debido a que puede provocar que el proyecto se retrase en varias etapas.
Catastrófico	Las medidas a tomar en el caso son de forma inmediata, puede provocar que el proyecto se detenga o retrase significativamente, teniendo que volver a empezar desde cero.

Tabla N°3. Tipos de daño.

Riesgo	Nivel de Impacto	Acción Remedial	Amonestación
Descarga de la batería del lego spike prime	5	Utilizar cargador y detener su uso hasta que la carga llegue al mínimo del 30%	No hay culpable directo
Error en la codificación	5	Corregir errores sintácticos y lógicos en lo posible, de no serlo investigar una solución o explorar otro enfoque.	Comunicarlo con el encargado responsable
descoordinación con los horarios de trabajos en conjunto	4	Organizar mejor los horarios disponibles del personal, para que así se puedan realizar..	No hay culpable directo
Personal faltando al horario asignado de trabajo	3	Adelantar tareas para evitar perjudicar al resto de integrantes y así terminar a tiempo el encargo.	Comunicarlo al responsable verbalmente
Dificultades con la conexión wifi	3	Esperar 10 minutos por si se logra volver a conectar automáticamente; si no, cambiar la conexión a una privada o directa.	No hay culpable directo
Atraso en el cumplimiento de tareas	3	Comunicar al equipo, y utilizar las horas extras disponibles, para solucionarlo.	Comunicarlo al responsable verbalmente
Ausencia de piezas	2	Solicitar las faltantes al administrador de piezas.	No hay culpable directo
Desempeño deficiente del prototipo	2	Modificar la estructura del prototipo siguiendo guías en línea o adaptando un nuevo diseño a lo requerido.	Conversar directamente con el encargado
Falla de registro en el redmine	1	Comunicar al administrador de la página para encontrar una solución.	No hay un culpable directo

Tabla N°4. Riesgos posibles en el desarrollo del proyecto.

4. Planificación de los Recursos

En esta sección se verán los recursos que fueron necesarios para concebir el proyecto.

4.1. Hardware

- Set de piezas Lego Spike Prime.
- HUB Lego Spike Prime.
- Computadores con sistema operativo compatible para la conexión del HUB.
- SPIKE Prime Set de expansión(v2).

4.2. Software

- Sistema operativo Windows.
- Página Redmine para la gestión del proyecto.
- Página Lego Education Spike para la programación en bloques y Micro Python.

4.3. Estimación de Costos

Estimar costos para el desarrollo de un proyecto es fundamental ya que permite tomar decisiones estratégicas informadas, lo que permite identificar y controlar gastos innecesarios, en esta sección se contemplan todos los costos que fueron necesarios para llevar a cabo el proyecto.

Costo de Hardware:

Los precios fueron estimados a partir de precios reales en tiendas online.

Set Lego Spike Prime [\[1\]](#) es un kit de lego de la linea spike prime el cual es una excelente herramienta educativa para aprender robótica, ingeniería y programación de manera divertida e interactiva

Set de expansión Lego Spike Prime [\[2\]](#) es una expansión del Set Lego Spike Prime [\[1\]](#) lo que conlleva a una excelente opción para aquellos que desean ir más allá con sus proyectos de robótica y programación, permitiendo una experiencia de aprendizaje aún más rica y variada.

En cuanto a equipo computacional, tanto Lenovo LOQ 15IAX9 [\[3\]](#), Lenovo V14 G2 ALC [\[4\]](#) como Dell AMD Ryzen 5 3450U [\[5\]](#) son las marcas de laptops que fueron necesarias para llevar a cabo el proyecto

Producto	Cantidad	Precio (CLP)
Set Lego Spike Prime [1]	1	\$ 399.950
Set de expansión Lego Spike Prime [2]	1	\$ 156.950
Lenovo LOQ 15IAX9 [3]	1	\$ 699.990
Lenovo V14 G2 ALC [4]	3	\$ 699.990
Dell AMD Ryzen 5 3450U [5]	1	\$ 349.990
Total:	7	\$ 2.306.870

Tabla N°5. Costos de hardware.

Costo de Software:

El proyecto no requirió de ningún tipo de software de pago.

Producto	Precio
Total :	\$ 0

Costo de Trabajador:

El salario promedio fue sacado de la página computrabajo [\[6\]](#); la manera para calcular los salarios es horas x precio/hora + horas extra x precio/hora x 1.5

Rol	Horas	Horas Extra	Precio / Hora	Total (CLP)
Jefe de proyecto	48 horas	5 horas	\$ 8.000	\$ 444.000
Programador	48 horas	5 horas	\$ 7.500	\$416.250
Ensamblador	48 horas	3 horas	\$ 7.000	\$367.500
Diseñador	48 horas	3 horas	\$ 7.000	\$367.500
Documentador	48 horas	4 horas	\$ 7.600	\$410.400
Total :	-	-	-	\$2.005.650

Tabla N°6. Estimación del costo del trabajador.

Destacado:

- La contabilización de las horas trabajadas comienza a partir de la formación del grupo de trabajo 26-09-2025.
- Para la categorización de las horas de trabajo, se tuvo en cuenta el tiempo de trabajo en clases.
- Para la categorización de las horas extras, se tuvo en cuenta el tiempo en las que se trabajó fuera del horario de clase.

Total de Costo:

Costo Hardware	\$2.306.870
Costo Software	\$ 0
Costo Empleados	\$ 2.005.650
Total :	\$ 4.321.520

Tabla N°7.Total de costos.

5. Análisis y Diseño

5.1. Especificación de Requerimientos

El proyecto tiene como objetivo principal salvaguardar la integridad física de los trabajadores del sector minero, a través del diseño y construcción de un prototipo robótico de clasificación de materiales. Se considera a dos actores principales: el usuario final que es el trabajador minero que será representado por un miembro del equipo y el cliente que es la empresa minera que solicita el proyecto, rol que será ejercido por el profesor a cargo del proyecto.

5.1.1. Requerimientos funcionales

Para poder medir el éxito del proyecto, se han definido los siguientes requerimientos funcionales, los cuales consideran una acción específica por cada ítem para asegurar su precisión y verificabilidad:

- **RF1 - Identificación de materiales:** El prototipo debe identificar cuatro tipos de materiales representados por piezas de LEGO de los siguientes colores: rojo, azul, amarillo y verde.
- **RF2 - Clasificación física:** El prototipo debe ser capaz de mover un bloque de color identificado hacia su cuadrante o contenedor correspondiente.
- **RF3 - Modo de operación automático:** El sistema debe contar con una función autónoma que ejecute cíclicamente la identificación (RF1) y la clasificación (RF2) de las piezas sin intervención del operador.
- **RF4 - Modo de control manual:** El prototipo debe permitir la ejecución de las acciones de clasificación (RF2) mediante comandos enviados por un operador desde la Interfaz Gráfica de Usuario (GUI).
- **RF5 - Visualización de datos:** La interfaz gráfica debe mostrar al operador el color detectado por el sensor del prototipo.
- **RF6 - Eliminación de Materiales:** El sistema debe permitir al operador eliminar cualquier material no deseado desde la interfaz gráfica.
- **RF7 - Control de contenedores:** El prototipo debe permitir el giro de los contenedores de depósito en ángulos de 0° y 180° para la recepción de piezas.

5.1.2. Requerimientos no funcionales

Disponibilidad: El prototipo debe garantizar una disponibilidad operativa del 100% durante el horario de la demostración. Durante operaciones mineras, debe operar por encima del 99.5% permitiendo realizar mantenimiento y cambio de las baterías.

Robustez: El prototipo debe manejar de forma elegante situaciones inesperadas como es el caso de detectar piezas de material que no puede clasificarse, moviendo la pieza a una zona de rechazo sin interferir en el proceso general de clasificación de las piezas.

Rendimiento: El tiempo de latencia entre el envío de un comando manual desde la interfaz de usuario y el inicio de la ejecución física del comando por el prototipo no debe superar los 500 milisegundos. El ciclo completo de clasificación no debe superar los 2 segundos.

Usabilidad: La interfaz gráfica de usuario debe ser diseñada bajo principios de diseño universal, debiendo obtener un puntaje mayor a 70 en el cuestionario SUS (System Usability Scale) y al menos el 80% de los usuarios o trabajadores deben completar la tarea de clasificación sin asistencia.

Seguridad: La interfaz gráfica debe contar con un botón virtual de parada de emergencia que al ser activado corte instantáneamente la potencia a todos los actuadores.

5.2. Arquitectura de Software

La arquitectura del sistema se basa en un modelo **Cliente-Servidor**, diseñado para permitir el control remoto y la supervisión del proceso de clasificación de minerales. Esta estructura separa la gestión del usuario de la ejecución física de las tareas en el prototipo.

5.2.1. Definición de Componentes

Bajo este enfoque, la solución se divide en dos entidades principales:

1. Cliente (Estación de Control): Representa toda la lógica que reside en el computador del operador. Para asegurar la mantenibilidad del código, el cliente se subdivide en dos módulos independientes:
 - Módulo de Interfaz (GUI): Encargado de la visualización de datos y la recepción de comandos del usuario (operador/jefe de área).
 - Módulo de Comunicación: Responsable de traducir las acciones de la interfaz en protocolos entendibles para el robot y gestionar la conexión vía Bluetooth.
2. Servidor (Unidad de Ejecución): Constituido por el LEGO Spike Prime HUB y su firmware. Actúa como el servidor de la arquitectura, ya que recibe las peticiones del cliente y gestiona directamente los recursos físicos (sensores y motores) para ejecutar las tareas de identificación y movimiento.

5.2.2. Diagrama de Arquitectura

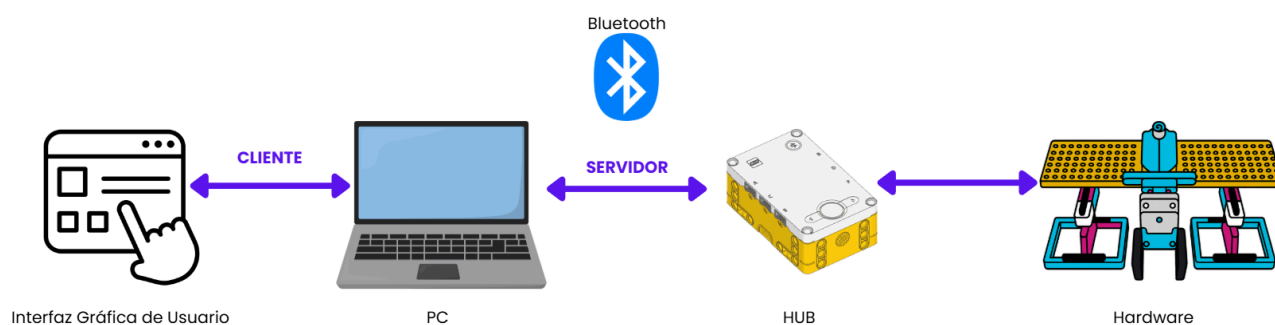


Figura n°2: “Diagrama de arquitectura”

La figura n°2 ilustra el flujo de información desde la interfaz gráfica (Cliente) hasta el Hardware, pasando por el servidor y el Firmware del HUB LEGO.

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

La interfaz de usuario presentada a continuación corresponde al wireframe de baja fidelidad concebido inicialmente para el manejo inalámbrico del prototipo robótico.

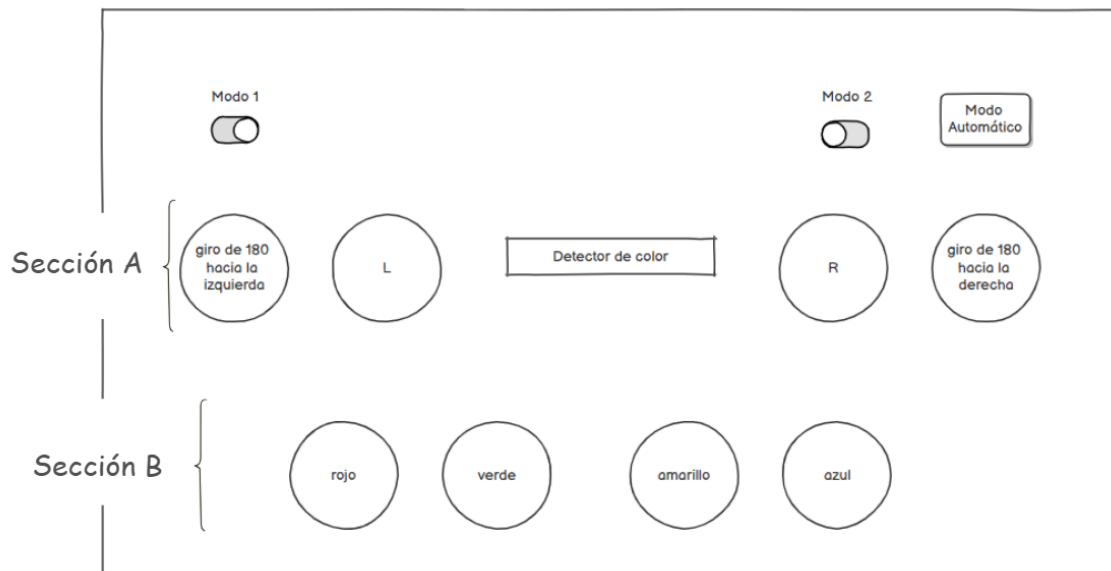


Figura n°3: “Wireframe GUI”

- **Detector de color:** Display o pantalla que muestra en tiempo real el color detectado por el sensor de color.
- **Botón L:** Acciona el brazo clasificador hacia la izquierda.
- **Botón R:** Acciona el brazo clasificador hacia la derecha.
- **Botones de giro 180°:** Mueven los contenedores 180° grados, uno hacia la izquierda y otro a la derecha.
- **Botones de Color:** Clasifica automáticamente el bloque de acuerdo al color.
- **Modo 1:**
 - Desactivado: Habilita la sección A de botones y deshabilita la sección B.
 - Activado: Habilita la sección B de botones y deshabilita la sección A.
- **Modo 2:**
 - Desactivado: Habilita la sección de botones de acuerdo al estado del modo 1 y deshabilita el botón Modo Automático.
 - Activado: Habilita el botón Modo Automático y deshabilita el resto de botones.

6. Implementación:

La etapa de implementación del proyecto se centró en llevar el diseño conceptual a solución funcional, integrando tanto el prototipo físico como el sistema de control por software. Para ello, se realizó el ensamblaje completo de la estructura con el kit LEGO education SPIKE Prime, incorporando motores, sensores y el HUB como unidad de control.

Paralelamente, se desarrolló el programa en python a través de pybricks que permite la operación automática del clasificador y su posterior control manual mediante interfaz gráfica. Durante el desarrollo del proyecto se llevaron a cabo múltiples pruebas para ajustar el correcto funcionamiento del sistema, especialmente en la sincronización entre la lectura del sensor de color y el accionamiento del mecanismo de empuje. Estas pruebas permitieron detectar y corregir errores tanto estructurales como de programación, logrando finalmente un sistema estable y operativo.

6.1. Fundamentos de los movimientos:

Para justificar la configuración seleccionada del robot clasificador se analiza una de las acciones más relevantes del sistema, que corresponde al giro de los motores para desviar las piezas hacia los distintos contenedores.

Basándonos en las especificaciones técnicas de los motores [\[11\]\[12\]](#), el prototipo utiliza un motor angular grande y dos motores angulares medianos.

Ambos motores cuentan con una velocidad máxima de 135 RPM (Revoluciones Por Minuto).

Además cabe mencionar que las piezas de prueba empleadas corresponden a los bloques Brick 2x4 w/cross hole de colores, las cuales miden 1,6 cm × 3,2 cm × 0,96 cm, teniendo un peso de 2,32 gramos aproximadamente.

Movimiento Rotación de Clasificador:

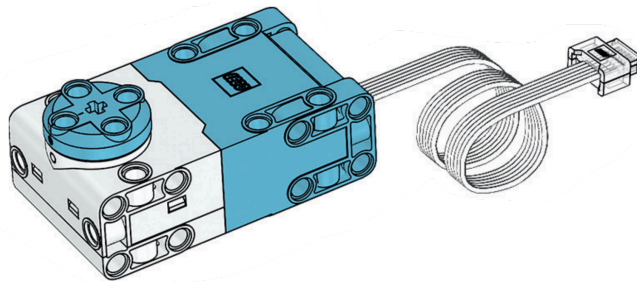


Figura n°4: "Modelo de Motor Principal"

El movimiento de clasificación principal se realiza empleando el motor angular grande, el cual debe rotar en un ángulo de 180° a -180° grados para mover las piezas a los contenedores correspondientes.

El bloque se encuentra a una altura aproximada de 8 cm, la barra del clasificador debe golpearlo y generar una velocidad de tal forma que la pieza realice una caída parabólica en un rango de 4 cm a 7 cm. Para ello calculamos el tiempo de caída empleando la altura.

$$t = \sqrt{\frac{2h}{g}} = \sqrt{\frac{0.16}{9.8}} \approx \sqrt{0.01633} \approx 0.1278 \text{ s}$$

Para que alcance la distancia de 4 cm la velocidad después del impacto debe ser:

$$v_b = \frac{x_{min}}{t} \approx \frac{0.04}{0.1278} \approx 0.313 \text{ m/s}$$

Para que alcance la distancia de 7 cm la velocidad después de impacto debe ser:

$$v_b = \frac{x_{max}}{t} \approx \frac{0.07}{0.1278} \approx 0.547 \text{ m/s}$$

Empleando choque elástico podemos calcular la velocidad del momento antes del impacto empleando la fórmula:

$$v_b = \frac{2M_{eff}}{M_{eff} + m} v_i$$

Donde v_i es la velocidad de la barra antes del contacto.

Donde M_{eff} es la masa efectiva de la barra y M su masa, lo que es equivalente a:

$$M_{eff} = \frac{M}{3} = \frac{0.0035}{3} \approx 0.0011667 \text{ kg}$$

La masa del bloque es de $m = 2.32 \text{ g} = 0.00232 \text{ kg}$

Obteniendo la siguiente fórmula:

$$v_b \approx 0.669 \cdot v_i$$

La cual podemos despejar y hallar la velocidad de la barra para cada distancia:

$$v_i^{min} \approx \frac{0.313}{0.669} \approx 0.468 \text{ m/s}$$

$$v_i^{max} \approx \frac{0.547}{0.669} \approx 0.818 \text{ m/s}$$

Por lo que la velocidad de la barra debe ser debe estar en el rango

$$0.468 \leq v_i \leq 0.818$$

Actualmente, la velocidad del motor del clasificador es de 600 *grados/segundos* lo que equivale a 100 RPM o una velocidad aproximada de 0.534 m/s que se encuentra dentro del rango óptimo de velocidad de impacto.

Movimiento Rotación de Contenedores:

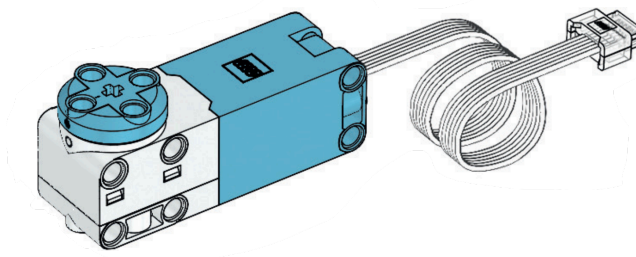


Figura n°5: “Modelo de Motores Secundarios”

El movimiento de los motores de los contenedores se basa en posiciones que se encuentran en 0° y 180° para capturar las piezas que el clasificador va dejando caer.

Considerando la velocidad máxima del motor de $\omega_{max} = 135 \text{ RPM} = 14.137 \text{ rad/s}$

y la velocidad angular máxima que puede lograr con un torque de 3.5 Nm

$$\alpha = \frac{\tau}{I} = \frac{3.5}{5.763 \times 10^{-5}} \approx 60750 \text{ rad/s}^2$$

Se obtiene la velocidad más óptima del motor:

$$v_{max} = \omega_{max} * r = 14.137 * 0.44 \approx 0.622 \text{ m/s}$$

Actualmente se aplica una velocidad 600 *grados/segundos* lo que equivale a una velocidad

de 0.534 m/s y un tiempo de giro de 0.3 s . A pesar de no ser el tiempo más óptimo, no sobreexige al motor dado que debe cargar con propia estructura y la de los bloques en los contenedores.

6.2. Descripción del sistema

El sistema que fue desarrollado para este proyecto está compuesto por tres principales componentes: Cliente, Servidor y Prototipo robótico, los cuales interactúan entre sí para dar lugar a la clasificación automática y manual de materiales.

El proceso comienza cuando el usuario interactúa con la interfaz gráfica (Cliente), lo cual envía una orden en específico. Esta solicitud es procesada por el servidor, que traduce esta orden en instrucciones que el HUB LEGO SPIKE Prime pueda entender. Finalmente el HUB ejecuta las acciones físicas por medio de los motores y el sensor completando el proceso de clasificación.

La comunicación entre el servidor y el prototipo se realiza a través de Bluetooth, lo que permite un control remoto seguro y flexible, manteniendo al operador fuera del área de riesgo, por si es que algún material sale de los lugares designados que tiene.

6.2.1. Cliente

El cliente como se ha mencionado con anterioridad corresponde a la aplicación de control ejecutada en el PC del operador minero, desarrollada en Python siendo esta una interfaz gráfica con la librería tkinter. Su principal función es permitir la interacción directa entre el operador y el sistema de clasificación.

A través de esta interfaz, el operador puede ejercer las siguientes capacidades vinculadas directamente a los requerimientos funcionales:

- **Gestión de Conexión:** Iniciar o finalizar el enlace inalámbrico con el servidor (HUB).
- **Control de Modo Automático (RF3):** Activar o desactivar la clasificación autónoma de piezas.
- **Control Manual de Clasificación (RF4):** Comandar los movimientos del motor principal y los contenedores de forma remota.
- **Monitoreo de Sensores (RF5):** Visualizar el color detectado por el sensor del prototipo.
- **Eliminación de material (RF6):** Eliminar el material no deseado.

Implementación del Cliente

```
class LegoGUI:
    def __init__(self, root: tk.Tk):
        self.root = root
        self.root.title("Control LEGO [ ] Motores A/B/C")
        self.root.geometry("700x600")

        self.log_queue = Queue()
        self._build_ui()
        self.worker = BLEWorker(self.log_queue, self.color_canvas, self.color_label)
        self.poll_logs()
```

Figura n°6: “Clase GUI”

La Figura N°6 muestra una de las partes más relevantes del código del cliente, específicamente la estructura de la clase principal encargada de la interfaz.

Clase LegoGUI: Esta clase centraliza la construcción de la interfaz de usuario (`_build_ui`) y la gestión de la cola de registros (`log_queue`) para mostrar eventos al operador.

```
Frame_btn = ttk.Frame(body, padding=20)
Frame_btn.pack(fill='both', expand=True)
btn_coman=[
    ("ROJO", [img_rojo, "#d9534f", "white", [img_rojo, "#b94a42"], lambda: self.worker.send_command("rojo")),
    ("AZUL", [img_azul, "#0275d8", "white", [img_azul, "#025aa5"], lambda: self.worker.send_command("azul")),
    ("AMARILLO", [img_amarillo, "#f0ad4e", "black", [img_amarillo, "#d08b36"], lambda: self.worker.send_command("amarillo")),
    ("VERDE", [img_verde, "#5cb85c", "white", [img_verde, "#4cae4c"], lambda: self.worker.send_command("verde")),
    ("LEER COLOR", [img_leer_color, "#00e1ff", "black", [img_leer_color, "#0093a7"], lambda: self.worker.send_command("Leer")),
    ("AUTOMÁTICO", [img_automatico, "#00e1ff", "black", [img_automatico, "#0093a7"], lambda: self.worker.send_command("auto")),
    ("ELIMNIAR PIEZA", [img_eliminar_pieza, "#00e1ff", "black", [img_eliminar_pieza, "#0093a7"], lambda: self.worker.send_command("Eliminar")),
    ("SALIR", [img_salir, "#00e1ff", "black", [img_salir, "#0093a7"], self.root.quit)
]
```

Figura n°7: “Implementación de los botones principales”

La Figura n°7 presenta la implementación de los botones de control, donde cada acción está vinculada a un comando enviado a través del worker hacia el servidor.

Implementación de los botones principales: En este segmento se definen los botones de colores (Rojo, Azul, Amarillo, Verde) que ejecutan la función `send_command`, cumpliendo con el requerimiento de control manual (RF4)

```

def on_connect(self):
    self.status.configure(text="Estado: conectando")
    self.worker.start()

    def wait_ready():
        if self.worker.running.is_set():
            self.status.configure(text="Estado: conectado")
        else:
            self.root.after(200, wait_ready)

    wait_ready()

def on_disconnect(self):
    self.worker.stop()
    self.status.configure(text="Estado: sin conexión")
    self._log("Desconectado.")

```

Figura n°8: “Implementación de solicitud de conexión cliente-servidor”

Con respecto a la comunicación, la Figura n°8 ilustra cómo el cliente solicita la conexión al servidor.

Gestión de estados de conexión en la interfaz: Estas funciones controlan el flujo de estado del cliente: al presionar "Conectar", se activa el hilo del worker y la interfaz cambia su estado visual a "conectando" o "conectado" una vez establecida la comunicación Bluetooth con el HUB.

Es importante destacar que la comunicación no es un flujo de datos binarios simples, sino un intercambio de scripts. La librería pybricksdev en el computador del cliente se comunica con el firmware del HUB. Cuando el usuario presiona un botón en la GUI para clasificar un mineral verde, el sistema realiza el siguiente flujo:

Para ver en más detalle el código de cliente entrar al repositorio de GitHub [\[13\]](#).

6.2.2. Servidor

En esta arquitectura, el servidor se analiza desde dos perspectivas complementarias que permiten la ejecución de las tareas de clasificación:

1. **Servidor Físico:** Corresponde al **LEGO Spike Prime HUB**. Es el hardware que ejecuta el firmware de Pybricks para ofrecer los servicios de control de motores y lectura de sensores.
2. **Servidor Lógico:** Es el componente de software encargado de representar al hardware y gestionar la comunicación técnica con él. Este reside parcialmente en el computador del cliente y utiliza la librería `pybricksdev` para enviar instrucciones al HUB mediante la conexión Bluetooth establecida.

Cabe destacar que este “Servidor lógico”, se ejecutará en el servidor Físico, por lo que sigue viviendo dentro del cliente.

BLEWorker es el código del cliente que establece esta conexión entre el cliente y el servidor.

Implementación del Servidor

La lógica del servidor se basa en la generación y envío de scripts temporales que el HUB puede interpretar. A continuación, se describen las funciones y componentes principales vinculados a los requerimientos funcionales:

```
class BLEWorker:
    def __init__(self, log_queue: Queue, color_canvas=None, color_label=None):
        self.loop = asyncio.new_event_loop()
        self.thread = threading.Thread(target=self._thread_main, daemon=True)
        self.queue = None
        self.hub = None
        self.running = threading.Event()
        self.log_queue = log_queue
        self.color_canvas = color_canvas
        self.color_label = color_label
```

Figura nº9: “Clase BLEWorker”

BLEWorker (Servidor Lógico): Es el componente encargado de mantener el hilo de comunicación activo. Gestiona la conexión Bluetooth y actúa como el puente que transporta los comandos desde la interfaz hacia el servidor físico, garantizando la disponibilidad (RNF2) y baja latencia (RNF3).

```
async def _runner(self):
    try:
        self.log("Buscando hub Bluetooth")
        device = await find_device("sp2")
        self.hub = PybricksHubBLE(device)
        await self.hub.connect()
        self.log("Conectado al Hub.")
        self.running.set()

        while True:
            comando = await self.queue.get()
            await execute_command(self, comando, self.log)
```

Figura nº10: “Implementación de la conexión a Bluetooth”

La función de la figura nº10 es la que mantiene la conexión Bluetooth.

```

def create_program(comando: str) -> str:
    actions = {
        "verde": """
target_B = 0
current_B = motorB.angle()
diff_B = (target_B - current_B) % 360
if diff_B > 180:
    diff_B = diff_B - 360
if abs(diff_B) > TOLERANCE:
    motorB.run_target(600, target_B)
motorA.run_angle(600, -180)
""",
        "amarillo": """
target_B = 180
current_B = motorB.angle()
diff_B = (target_B - current_B) % 360
if diff_B > 180:
    diff_B = diff_B - 360
if abs(diff_B) > TOLERANCE:
    motorB.run_target(600, target_B)
motorA.run_angle(600, -180)
""",
        "azul": """
target_C = 180
current_C = motorC.angle()
diff_C = (target_C - current_C) % 360
if diff_C > 180:
    diff_C = diff_C - 360
if abs(diff_C) > TOLERANCE:
    motorC.run_target(600, target_C)
motorA.run_angle(600, 180)
""",
        "rojo": """
target_C = 0
current_C = motorC.angle()
diff_C = (target_C - current_C) % 360
if diff_C > 180:
    diff_C = diff_C - 360
if abs(diff_C) > TOLERANCE:
    motorC.run_target(600, target_C)
motorA.run_angle(600, 180)
""",
        "Leer": """
sensor = ColorSensor(Port.B)
color = sensor.color()
print(color)
""",
        "Eliminar": """
motorA.run_angle(100, 180)
#motorA.track_target(motorA.angle()-180)
"""
    }

```

Figura n°11: "Implementación de función para crear código del HUB"
create_program (Lógica de Control): Esta función es fundamental para la operación. No envía un

programa estático al HUB, sino que genera dinámicamente archivos temporales que contienen las instrucciones específicas en lenguaje Python (compatibles con Pybricks). Estos "programas" temporales son los que definen cómo se ejecutará la **identificación de materiales (RF1)** y la **clasificación física (RF2)**.

```
async def execute_command(worker, comando: str, log_cb=None):
    program = create_program(comando)

    with tempfile.NamedTemporaryFile(mode='w', suffix='.py', delete=False, encoding='utf-8') as tf:
        tf.write(program)
        temp_path = tf.name

    try:
        if log_cb:
            log_cb(f"Ejecutando: {comando}")
        # Para recibir datos, print_output debe ser True
        if "Leer" in comando:
            try:
                color=await worker.hub.run(temp_path, wait=True)
                color_str = color.strip().lower()
                worker.log(f"Color detectado: {color_str}")
                if worker.color_canvas:
                    worker.color_canvas.configure(bg=color_str)
                if worker.color_label:
                    worker.color_label.configure(text=f"Color detectado: {color_str}")
            except Exception as e:
                worker.log(f"Error leyendo color: {e}")
        else:
            await worker.hub.run(temp_path, wait=True, print_output=True)

        #await hub.run(temp_path, wait=True, print_output=True)
        if log_cb:
            log_cb(f"Ejecutado: {comando}")

    except Exception as e:
        if log_cb:
            log_cb(f"Error ejecutando comando: {e}")

    finally:
        try:
            os.unlink(temp_path)
        except:
            pass
```

Figura nº12: "Implementación de función para ejecutar comandos"

execute_command (Ejecución): Es la función encargada de tomar el script generado por create_program y enviarlo al servidor físico utilizando pybricksdev. Esta herramienta es la que permite que el HUB reciba y ejecute las instrucciones en tiempo real, ya sea para el **modo automático (RF3)** o el **control manual (RF4)**.

Para ver en más detalle el código de servidor entrar al repositorio de GitHub [\[13\]](#).

6.2.3. Interfaz gráfica de usuario (GUI)

La interfaz gráfica de usuario tiene como principal objetivo ser el medio de comunicación entre el operador y el sistema. Fue diseñada con el objetivo de ser intuitiva, clara y fácil de usar, incluso para usuarios sin experiencia previa.

A continuación se mostrará una figura con la interfaz gráfica de usuario actual del sistema y se especificará una por una todas sus funciones.



Figura n°13: “Interfaz gráfica del prototipo robótico”

- **Botón conectar:** Inicia la conexión vía bluetooth entre el sistema en el computador y el hub Spike Prime.
- **Botón desconectar:** Finaliza la conexión entre el sistema en el computador y el hub Spike Prime.
- **Display de color:** Visualizador que muestra el color detectado por el sensor de color.
- **Display de texto:** Visualizador que muestra el nombre del color detectado; en caso de no haber piezas muestra “No detectado”
- **Botón Rojo:** Clasifica la pieza a lado izquierdo (vista delantera del prototipo) en el contenedor de posición 0°.
- **Botón Azul:** Clasifica la pieza a lado izquierdo (vista delantera del prototipo) en el contenedor de posición 180°.
- **Botón Amarillo:** Clasifica la pieza a lado derecho (vista delantera del prototipo) en el

contenedor de posición 0°.

- **Botón Verde:** Clasifica la pieza a lado derecho (vista delantera del prototipo) en el contenedor de posición 180°.
- **Botón Leer Color:** Inicia el sensor de colores para luego mostrar el color detectado en el display de colores.
- **Botón Automático:** Inicia el prototipo robótico en modo automático, clasificando los bloques de forma automática.
- **Botón Eliminar Pieza:** Elimina la pieza deslizando lentamente con el fin de que no se ingrese junto a las demás piezas.
- **Botón Salir:** Finaliza el código, terminando todas las acciones y cerrando la interfaz gráfica.
- **Display de Registro:** Visualizador que muestra los comandos ejecutados por el sistema y los errores que pueden producirse.

7. Resultados

En la siguiente sección se detallan los resultados finales alcanzados tras el término de todas las fases de desarrollo, implementación y pruebas del proyecto "Sorting Mining". Se presenta una evaluación que contrasta los logros obtenidos con las expectativas planteadas en la etapa de diseño.

A continuación, se expone el estado definitivo del sistema, analizando el cumplimiento de los requerimientos funcionales y no funcionales, así como la resolución de los desafíos técnicos que se presentaron a lo largo del desarrollo del proyecto.

7.1. Estado actual del proyecto

Tras concluir todas las etapas de desarrollo, el proyecto "Sorting Mining" ha finalizado satisfactoriamente, cumpliendo con la gran mayoría de las metas propuestas al inicio del semestre. A continuación, se detalla el estado final de los requerimientos establecidos en el diseño:

Cumplimiento de Requerimientos Funcionales:

- Identificación y Clasificación (RF1, RF2, RF3): Se logró la detección fiable de los cuatro colores (rojo, azul, amarillo y verde). El sistema opera de forma autónoma y cíclica, moviendo cada pieza a su contenedor correspondiente mediante un flujo continuo de "detectar-decidir-mover".
- Control Manual y GUI (RF4, RF5): La interfaz gráfica desarrollada en Tkinter permite el envío de comandos inalámbricos en tiempo real, cumpliendo con la teleoperación manual del brazo clasificador y los contenedores.
- Gestión de Materiales (RF6, RF7): Se implementó con éxito la función de eliminación de materiales no deseados y el control de rotación de los contenedores de depósito para la recepción de piezas en ángulos de 0° y 180°

Cumplimiento de Requerimientos No Funcionales:

- Disponibilidad y Rendimiento: El prototipo garantizó una disponibilidad operativa del 100% durante las pruebas finales. La latencia de comunicación vía Bluetooth se mantuvo bajo los 500 milisegundos, asegurando una respuesta inmediata a los comandos del operador.
- Robustez y Seguridad: El sistema maneja de forma elegante las piezas no clasificables a través del botón de descarte, evitando colapsos en la línea de producción. Asimismo, la GUI integra la lógica de parada y salida segura para proteger los actuadores del kit LEGO Spike Prime.
- Usabilidad: La interfaz final resultó intuitiva y clara, permitiendo que el operador controle el sistema sin asistencia externa, validando los principios de diseño universal propuestos.
- Seguridad: El sistema implementa un protocolo de detención segura a través del botón 'SALIR' en la interfaz gráfica. Al activarse, el software garantiza la finalización inmediata de los procesos lógicos y el corte de comunicación con los actuadores, asegurando que el prototipo quede en un estado de reposo seguro

7.2. Problemas encontrados y solucionados

Problema 1: Posición incorrecta de los bloques al ingresar al sistema

Uno de los primeros problemas detectados durante la etapa de ensamblaje fue la forma en que los bloques ingresaban al sistema. Al comienzo, los bloques no siempre ingresaban de la misma forma al prototipo: a veces caían de lado, otras inclinados o en una posición que dificultaba que la garra los tomara bien. Esto provocaba fallos en la clasificación y hacía que el sistema fuera menos confiable de lo esperado.

Problema de categoría 2

Solución: Para resolver este inconveniente, se rediseñó la entrada de los bloques, guiándolos de tal manera que solo pudieran ingresar en posición vertical, estandarizando su orientación antes de ser clasificados. Este problema está directamente relacionado con el riesgo de desempeño deficiente del prototipo, ya que afectaba la calidad del funcionamiento general y obligó a modificar la estructura para cumplir con lo planificado.

Problema 2: Desincronización en la secuencia operativa del sistema de clasificación

Durante la implementación del sistema de sorting de colores, se identificó otro inconveniente importante asociado en la lógica de control: en algunas pruebas, el sistema ejecutaba movimientos antes de terminar de procesar correctamente la lectura del sensor de color. Esto provocaba que ciertos bloques no se enviaran al contenedor correcto, reduciendo la precisión de la clasificación.

Problema de categoría 5

Solución: La solución consiste en reorganizar la ejecución programática. Se reestructuró la lógica de control, incorporando puntos de verificación intermedios y reorganizando el orden de los comandos, con el fin de asegurar que cada etapa del proceso de clasificación, detección evaluación, decisión y movimiento, se complete de manera satisfactoria antes de avanzar a la siguiente acción. La reestructuración de la lógica de control permitió eliminar definitivamente la desincronización entre sensores y actuadores. Este problema se vincula con el riesgo de error en la codificación descrito en la matriz de riesgos, ya que se originó en una implementación inicial poco ordenada de la lógica del programa.

Problema 3: Durante el desarrollo del proyecto se presentó una dificultad técnica muy relevante relacionada con la integración del HUB con la interfaz gráfica desarrollada en Tkinter. El equipo no contaba inicialmente con una solución funcional que permitiera establecer esta comunicación que debía existir entre el HUB y la interfaz mediante Bluetooth, lo que imposibilitó la continuación del proyecto durante un periodo de dos semanas. Esto debido a que sin una conexión estable no era posible continuar con el desarrollo ni comprobar el funcionamiento del prototipo. El problema específico era la necesidad de integrar la librería de Tkinter con el HUB.

Problema de categoría 5

Solución: La solución se logró mediante el apoyo puntual de un compañero externo al equipo de trabajo quien nos colaboró en la implementación inicial de la conexión entre el HUB y la librería Tkinter. Esta colaboración fue exclusivamente aislada a este problema específico sin participación continua de este compañero en el desarrollo, diseño ni toma de decisiones del proyecto. La integración técnica entre el HUB y la interfaz Tkinter quedó plenamente operativa, permitiendo una comunicación inalámbrica robusta y de baja latencia. Esta situación se relaciona con el riesgo de dificultades con la conexión que se había contemplado previamente, ya que afectó directamente la comunicación entre el cliente y el prototipo y generó retrasos en las pruebas planificadas.

Problema 4: Error de lectura por proximidad excesiva en el sensor de color

Durante las pruebas del modo manual, se detectó que el sensor de color entregaba lecturas erróneas o nulas ("color negro") cuando las piezas se posicionaban demasiado cerca de la lente. Esto impedía que el operador remoto conociera el color del bloque en la interfaz gráfica, dificultando la toma de decisiones manual.

Problema de categoría 2

Solución: Se determinó que el sensor requiere una distancia mínima de aproximadamente 3 mm para que la luz proyectada se refleje correctamente en la superficie de la pieza. Para solucionar esto de forma permanente, se instaló una capa transparente sobre el sensor. Esta solución mediante un espaciador físico transparente no sólo corrigió el error de lectura, sino que estandarizó la distancia del sensor, eliminando la variabilidad en los datos recibidos por la GUI y permitiendo una operación remota 100% fiable. Cabe destacar que este ajuste no afectó el modo automático, ya que en dicha función el sensor realiza la detección de los bloques de forma dinámica mientras estos se desplazan por la rampa, antes de alcanzar el punto de detención.

8. Prueba de funcionamiento

Las pruebas de funcionamiento realizadas con el prototipo tienen el objetivo de comprobar que los componentes que lo conforman cumplan con los requisitos establecidos al inicio del proyecto. En particular, se evaluó el desempeño del sistema clasificador en la detección y clasificación de minerales, representados por piezas de cuatro colores (rojo, amarillo, verde y azul), además del correcto funcionamiento de la interfaz de usuario para el control remoto del prototipo.

8.1 Descripción de la prueba de funcionamiento

Las pruebas se desarrollarán con el propósito de comprobar el desempeño del sistema bajo condiciones normales y condiciones irregulares de operación, simulando escenarios similares a los que podrían presentarse en la práctica.

Prueba de condiciones normales de operación

Las piezas deberán ser depositadas tanto manualmente, por los integrantes del equipo, como mediante el brazo robótico, en el canal de recepción. Desde allí, deberán descender sin inconvenientes hasta ser detectadas por el sensor de color.

A través de la interfaz gráfica de usuario, se deberá visualizar y confirmar el color de cada pieza, verificando así el funcionamiento del sistema de detección basado en el sensor de color.

Posteriormente, las piezas deberán clasificarse en sus contenedores correspondientes utilizando los botones presentes en la interfaz gráfica, con el fin de validar la correcta separación según el tipo de material.

Además, se deberá verificar el modo de ejecución automático mediante el botón **AUTOMÁTICO** de la interfaz, comprobando que el sistema detecte y clasifique las piezas sin intervención del usuario.

Prueba de condiciones irregulares de operación

Aunque inicialmente no se contemplaban casos particulares, se considerará su incorporación debido a que representan situaciones más cercanas a la realidad.

La prueba consistirá en introducir una pieza de un color diferente de los cuatro preestablecidos. En caso de no ser reconocida por el sistema, la interfaz gráfica deberá permitir su descarte mediante el botón **ELIMINAR PIEZA**, desplazándola hacia un costado y evitando su ingreso a los contenedores de clasificación. Este mismo procedimiento deberá verificarse también en el modo automático del prototipo.

8.2 Resultados observados para la prueba de funcionamiento

Resultados de prueba de condiciones normales de operación:

Se depositaron las piezas en el canal receptor del prototipo, tanto de forma manual como con la ayuda del brazo robótico. Una vez llegaron a la parte inferior donde se halla el sensor de color y utilizando el botón LEER COLOR presente en la interfaz gráfica, se logró visualizar el color de la pieza. Posteriormente se utilizaron los botones de colores para clasificar la pieza y depositarla a su contenedor correspondiente.

Resultados de prueba de condiciones irregulares de operación:

Similar al caso anterior, se depositó la pieza en el clasificador y activando el modo automático desde la interfaz gráfica las piezas fueron detectadas y clasificadas sin intervención.

Finalizadas la pruebas se detectaron dos fallos principales:

- **Fallo en el diseño de la estructura del clasificador:** cuando las piezas son depositadas y llegan a la parte inferior, si presentan un tono oscuro, el sensor no las detecta adecuadamente y las interpreta como color negro, el cual no está considerado entre los colores definidos (rojo, amarillo, azul y verde). Esto ocurre porque el sensor requiere una distancia mínima de 3 mm para que la luz proyectada se refleje y permita detectar el color correcto.
- **Problema al desconectar el dispositivo:** al finalizar la operación o si se deja de usar por un prolongado periodo de tiempo, solo se desconecta el computador, mientras que el hub permanece en un estado que requiere ser reiniciado para poder reconectarse.

Exceptuando estos errores, los resultados obtenidos fueron satisfactorios. Se observa un funcionamiento óptimo del prototipo tanto en la clasificación manual como automática, aunque esta última está limitada a cinco iteraciones. Se cumplió con los requisitos definidos, logrando detectar las piezas mediante el sensor de color y depositarlas en sus contenedores correspondientes, todo ello a través de una interfaz de usuario que permite el control remoto del prototipo.

9. Conclusión

En conclusión el proyecto desarrollado denominado Sorting Mining nos permitió aplicar de forma práctica los conocimientos que hemos ido adquiriendo en lo que va del transcurso de la carrera. Además de que nos permitió desarrollar y fortalecer competencias claves en nuestra formación como Ingenieros Civiles en Computación e Informática, tales como el trabajo en equipo, el proceso de gestación y desarrollo de proyectos más formales, además de emplear fundamentos y la arquitectura cliente-servidor.

El proyecto se desarrolló utilizando el set Lego Spike Prime para la implementación de un prototipo robótico funcional con la capacidad de identificar y clasificar cuatro tipos de materiales representados por piezas de colores (rojo, azul, amarillo, verde). Cumpliendo satisfactoriamente los objetivos estipulados al comienzo del proyecto. Implementado exitosamente una interfaz gráfica de usuario con tkinter para el control remoto del prototipo logrando así el propósito de realización de este proyecto, preservar la seguridad de los trabajadores mineros y digitalizar el proceso minero de clasificación.

Aunque no se estipulaba al comienzo del proyecto, el prototipo cuenta con funciones que lo vuelven más robusto, permitiendo desplazar piezas que no pertenecen a los cuatro designados previamente.

Aunque el estado actual del prototipo cumple con los requerimientos básicos de funcionalidad y ejecución de tareas, se tiene ciertas proyecciones a futuro que mejoran y solventa ciertas funciones del prototipo, como por ejemplo:

Implementar más botones a la interfaz gráfica para un modo más manual de control, donde se pueda mover individualmente cada contenedor y la barra clasificadora.

Optimizar la función del botón automático, pues actualmente cuenta con un limitante de solo cinco piezas, se busca que este opere de forma indefinida hasta que el usuario desee detener el proceso. Cabe destacar que esta restricción operativa de 5 ciclos en el modo automático, utilizada durante las pruebas, fue implementada deliberadamente como un protocolo de seguridad para supervisar la integridad del hardware LEGO. Para una escala industrial, esta limitación es fácilmente parametrizable, ya que la arquitectura del software soporta flujos de trabajo masivos sin pérdida de estabilidad.

Implementar audios de los colores al momento de detectar las piezas y en el mejor de los casos en cada botón, indicando la acción que se está realizando.

Finalmente, la realización de este proyecto ha sido una experiencia bastante desafiante, tenido momentos que si no hubiera sido por ayuda de algunos compañeros hubiera resultado en un trabajo menos funcional. Actualmente el prototipo cumple con los requisitos estipulado inicialmente por lo consideramos que se ha dado una adecuada solución al problema planteado, no así siendo la definitiva y aún está abierto el camino para un futuro trabajo.

10. Referencias

- [1] “set lego education spike prime”. lego.com. Disponible: <https://www.lego.com/es-us/product/lego-education-spike-prime-set-45678>
- [2] “set lego education spike prime expansion”. lego.com. Disponible: <https://www.lego.com/es-us/product/lego-education-spike-prime-expansion-set-45681>
- [3] “Lenovo LOQ 15IAX9”. riplely.cl. Disponible: https://simple.ripley.cl/notebook-gamer-lenovo-loq-intel-core-i5-16gb-ram-512gb-ssd-nvidia-rtx-3050-156-2000401757082p?color_80=gris&s=mdco
- [4] “Lenovo V14 G2 ALC [82KC00DECL]” opcstore.cl Disponible: <https://opcstore.cl/products/notebook-lenovo-v14-alc-8gb-ssd-512gb-14-hd-w10-pro>
- [5] “Dell AMD Ryzen 5 3450U” riplely.com Disponible: <https://simple.ripley.cl/notebook-dell-inspiron-3505-amd-ryzen-5-8-gb-ram-256gb-ssd-156-2000381708951p?s=mdco>
- [6] “salario medio de Ingeniero informático en Chile”: <https://cl.computrabajo.com/salarios/ingeniero-informatico>
- [7] Aminerals, “Innovación”, Aminerals, [En línea]. Disponible en: <https://www.aminerals.cl/innovacion>
- [8] “Automatización en la industria minera ¿Cuáles son los desafíos y beneficios?”, YouTube, vídeo en línea, 2025. Disponible en: <https://youtu.be/XRxxaLYVreU> .
- [9] Consejo Minero, “Cifras actualizadas de la minería”, Consejo Minero, [En línea]. Disponible en: <https://consejominero.cl/mineria-en-chile/cifras-actualizadas-de-la-mineria/>.
- [10] SONAMI, “SONAMI proyecta que en 2025 la minería continuará aportando más del 10% del PIB”, SONAMI, 19 nov. 2024. [En línea]. Disponible en: <https://www.sonami.cl/v2/noticias/sonami-proyecta-que-en-2025-la-mineria-continuara-aportando-mas-del-10-del-pib/>).
- [11] LEGO Education, “Technic Large Angular Motor – Technical Specifications”, LEGO Education, s. f. [En línea]. Disponible en: https://assets.education.lego.com/v3/assets/blt293eea581807678a/bltb9abb42596a7f1b3/5f8801b5f4c5ce0e93db1587/le_spike-prime_tech-fact-sheet_45602_1hy19.pdf?locale=en-us
- [12] LEGO Education, “Technic Medium Angular Motor – Technical Specifications”, LEGO Education, s. f. [En línea]. Disponible en: https://le-www-live-s.legocdn.com/sc/media/files/support/spike-prime/techspecs_technicmediumangularmotor-19684ffc443792280359ef217512a1d1.pdf
- [13] Camilo2074, “Proyecto1 - Sistema de Clasificación de Bloques con LEGO SPIKE Prime”, *GitHub Repository*, 2025. [En línea]. Disponible en: <https://github.com/camilo2074/proyecto1>