

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

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



**Plan de Proyecto
“Facilitador
Logístico
Eléctrico
Transportador de
Equipo”**

Alumnos: Felipe Diaz Araos

José Yampara Yampara

Pablo Andia Lopez

Edynson Tola Fernandez

Martin Acevedo Tudela

Asignatura: Proyecto I

Profesor: Baris Nikolai Klobertanz

10 – 2025

Historial De Cambios

Fecha	Versión	Descripción	Autor(es)
11/09/2025	1.0	Recopilación de información	Todo el grupo
12/09/2025	1.1	Creación del nombre del proyecto	Martin Acevedo Pablo Andia
24/09/2025	1.2	Cambios en la portada	José Yampara Felipe Diaz
25/09/2025	1.3	Incorporación de la carta gantt	Todo el grupo
02/10/2025	1.4	Objetivos generales y específicos	Felipe Diaz
03/10/2025	1.5	Incorporación de Tablas de roles y actividades	Edynson Tola Felipe Diaz
09/10/2025	1.6	Finalización de Introducción	Edynson Tola
10/10/2025	1.7	Incorporación de tabla de riesgos	Pablo Andia
16/10/2025	1.8	Finalización de la conclusión	Felipe Diaz
17/10/2025	1.9	Revisión de ortografía y formato	Edynson Tola
11/09/2025	1.10	Actualización de la conclusión	Edynson Tola

Índice de contenidos

1. Panel General	6
1.1. Introducción	6
1.2. Objetivos	6
1.2.1. Objetivo General	6
1.2.2. Objetivos Específicos	6
1.3. Restricciones	7
1.4. Entregables	7
2. Organización del Personal	8
2.1. Descripción de los Roles	8
2.2. Personal que Cumplirá los Roles	8
2.3. Métodos de Comunicación	9
3. Planificación del Proyecto	10
3.1. Actividades	10
3.2. Carta Gantt	13
3.3. Gestión de Riesgos	13
4. Planificación de los Recursos	16
4.1. Hardware	16
4.2. Software	16
4.3. Estimación de Costos	17
5. Análisis y diseño	19
5.1. Especificación de requerimientos	19
5.1.1. Requerimientos funcionales	19
5.1.2. Requerimientos no funcionales	19
5.2. Arquitectura del software	20
5.3. Diseño inicial de la interfaz gráfica de usuario (GUI)	21
6. Implementación	22
6.1. Fundamentos de los movimientos	22
6.2. Descripción del sistema	22
6.2.1. Cliente	22
6.2.2. Servidor	22
6.2.3. Interfaz gráfica de usuario (GUI)	23
7. Resultados	24
7.1. Estado actual del proyecto	24
7.2. Problemas encontrados y solucionados	25
8. Prueba de funcionamiento del sistema	26
8.1. Descripción de la prueba de funcionamiento	26
8.2. Resultados observados para la prueba de funcionamiento	29
9. Conclusión	30
10. Referencias	31

Índice de tablas

Tabla 1: Roles asignados al personal	8
Tabla 2: Planificación, ejecución y resultados del proyecto	10
Tabla 3: Riesgos, gravedad y acciones remediales	15
Tabla 4: Costo de Hardware	17
Tabla 5: Costo de Software	17
Tabla 6: Costo de Trabajador	18
Tabla 7: Total de Costo	18
Tabla 8: Frecuencia de riesgos identificados y acciones remediales	25

Índice de figuras

Figura 1: Carta Gantt	13
Figura 2: Diagrama de la arquitectura cliente-servidor de F.L.E.T.E.	20
Figura 3: distribución de botones de la UI	21
Figura 4: Circuito de obstáculos planeado	27
Figura 5: Circuito de obstáculos realizado	28

1. Panel General

1.1. Introducción

La industria minera enfrenta el desafío de mantener altos niveles de productividad sin comprometer la seguridad de los trabajadores.

Como estudiantes de **Ingeniería Civil en computación e Informática**, el objetivo fue simular el proceso de **transporte de mineral en minería subterránea**, aplicando principios de automatización y control remoto mediante el uso del kit **LEGO Spike Prime**.

El proyecto busca poder crear un prototipo de transporte de minería, que logre optimizar y garantizar la seguridad de las personas.

Para esto, se seguirán las ideas empleadas en la Minería 4.0, en donde se implementaron mejoras tecnológicas en la automatización para aumentar la producción y la reducción de la exposición humana.

1.2. Objetivos

1.2.1. Objetivo General

Con el Set Lego Spike Prime se creará y programará un robot capaz de llevar a cabo la función de transportar carga simulado con bloques pequeños de lego, parecido a la labor de los vehículos de carga que se llevan a cabo en las obras, el robot dispondrá de una interfaz gráfica en donde se manejan las funciones hechas con Python. El robot estará hecho exclusivamente con las piezas del Spike Prime, con la finalidad de demostrar la automatización de transporte de carga en ambientes laborales.

1.2.2. Objetivos Específicos

- Estudiar los componentes del set Lego Spike Prime.
- Proponer y seleccionar el mejor prototipo para el proyecto.
- Construir el prototipo seleccionado.
- Investigar y analizar las librerías disponibles en Python, hasta encontrar uno que permita el control completo del robot.
- Programar las funciones del robot (avance, retroceso, giro y frenado), haciendo uso de buenas prácticas de programación como utilizar algoritmos que ayuden a procesar datos de manera eficiente y funciones que puedan ser útiles para los movimientos básicos del robot.
- Desarrollar una interfaz gráfica con Python y Tkinter que reciba las instrucciones del robot seleccionado.

1.3. Restricciones

- o Se debe programar solo en Python.
- o Solo se debe utilizar la plataforma Redmine para los documentos y avance del proyecto.
- o Se debe utilizar solo el Set de Lego Spike Prime.
- o El uso del set solo está habilitado en la universidad.
- o Cantidad de integrantes limitada a solo 5.
- o El control remoto está limitado por la conectividad Bluetooth y el alcance del hub del LEGO Spike Prime.
- o Tiempo límite para la finalización del proyecto.

1.4. Entregables

Bitácoras: Informes semanales que describen el avance general del equipo en el proyecto. Incluye actividades realizadas, dificultades encontradas, recomendaciones para mejorar y acciones tomadas.

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 nuestra organización y estrategia 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.

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 robot.

2. Organización del Personal

La organización del grupo se llevó a cabo mediante una reunión en donde se vieron las tareas para completar el proyecto, considerando las habilidades de cada integrante, se distribuyeron los roles, así se asignaron las funciones de jefe de proyecto, ensamblador, programador, documentador y diseñador, asegurando que cada miembro trabaje en la labor que se le asignó.

2.1. Descripción de los Roles

Jefe de proyecto: Representante del equipo, supervisa, organiza el progreso del proyecto y gestiona las reuniones grupales (dailys).

Ensamblador: Encargado del montaje, armado del robot y del conteo de piezas del set Lego Spike Prime.

Programador: Encargado del área de la codificación y funcionamiento del robot.

Documentador: Encargado de realizar las bitácoras semanales, llevar una retroalimentación sobre las dificultades y actividades a realizar cada semana. También es el encargado de realizar el informe.

Diseñador: Encargado de la creación de la interfaz (GUI), presentaciones y logos o imágenes necesarias para el correcto desarrollo del proyecto.

2.2. Personal que Cumplirá los Roles

Tabla 1: Roles asignados al personal

Rol	Responsable	Ayudantes
Jefe de proyecto	Martin Acevedo	Martin Acevedo
Ensamblador	José Yampara	José Yampara Felipe Diaz Martin Acevedo
Diseñador	Felipe Diaz	Edynson Tola José Yampara
Programador	Pablo Andia	Pablo Andia José Yampara Felipe Diaz
Documentador	Edynson Tola	Felipe Diaz Edynson Tola

2.3. Métodos de Comunicación

En cuanto a los medios de comunicación que utilizaremos, el medio principal será la aplicación de celular Whatsapp, se usará principalmente para acordar actividades, dividir trabajo y dar avisos importantes respecto al presente proyecto.

De manera conjunta, se hará uso de plataformas como Gmail para la transmisión de documentos y archivos relacionados a la confección del proyecto.

3. Planificación del Proyecto

Desglose detallado de las actividades realizadas, incluyendo la descripción de cada tarea, la identificación de los responsables correspondientes y el objetivo que lleva por cumplir.

3.1. Actividades

Tabla 2: Planificación, ejecución y resultados del proyecto

Nombre	Descripción	Responsables	Objetivo	Resultados
Planificación del proyecto	Se prevén las diferentes etapas a superar para lograr el objetivo general propuesto	Todo el grupo	Establecer un esquema de trabajo claro y asignar responsabilidades para optimizar el tiempo de ejecución	Definición del nombre del proyecto y división de roles
Asignación de roles	Se organizan los roles para el proyecto	Todo el grupo	Conocer las habilidades de cada integrante para asignarles un rol en específico de acuerdo con las habilidades del equipo	Se divide el trabajo en función del rol asignado a cada integrante
Análisis de tipos de robots	Se evaluaron los objetivos del proyecto, para la correcta elección del prototipo	Todo el grupo	Llegar a un acuerdo para poder realizar qué tipo de robot se usaría para el proyecto	Selección del robot “Carrito de carga” en Lego Education Spike”
Selección del robot “Carrito de reparto”	Se arma el robot según el tutorial de la app “Lego Education Spike - 3.5.1”	José Yampara Felipe Diaz Pablo Andia	Llevar a cabo el robot final que se realizará y empezar la producción y programación del robot	Separación de piezas para el armado del robot
Investigación sobre la mejor librería de python para el control del “Carrito de reparto”	Se seleccionó la librería Pybrick	Pablo Andia Edynson Tola	Encontrar una manera de brindar una conexión entre el robot y el entorno de desarrollo integrado	Ánálisis de las librerías disponibles
Conteo de piezas del Set Lego Spike Prime	Se hace un recuento y verificación de todas las piezas del set Lego Spike Prime	José Yampara	Organizar las piezas que se van a usar para la construcción del robot	Se hizo un inventario de las piezas

Realización de la bitácora número uno	Se detallaron las tareas que se hicieron en la primera semana	Edynson Tola	Registrar un avance semanal de lo que se ha hecho, mostrar los problemas encontrados y sus posibles soluciones, mostrar si el proyecto avanza según lo planeado y plantear las tareas que se harán en la siguiente semana	Se redacta la bitácora número uno de lo trabajado esta semana
Testeos de prueba con la codificación mediante bloques	Pruebas de movilidad	Pablo Andia Edynson Tola Martin Acevedo	Buscar algún tipo de algoritmo en el cual poder hacer funcionar el robot mediante el sistema de programación con bloques	Bosquejo de las funciones y algoritmos a desarrollar
Desarrollo en librería Pybricks	Se empieza de lleno con la programación e investigación de la librería en Python	Pablo Andia José Yampara	Programar los movimientos del robot para que cumpla con los requerimientos solicitados	Código más limpio y redactado con buenas técnicas de programación
Desarrollo de la estructura funcional y algorítmica	Se realizan las funciones esenciales para el robot seguido de un algoritmo rápido y eficiente	Pablo Andia Edynson Tola Martin Acevedo	Hacer las funciones principales de movimiento para el robot con un algoritmo que haga el proceso más rápido y disminuya el tiempo de respuesta del robot	Se le brindó movimiento al robot
Realización de la bitácora número dos	Se detallaron las tareas que se hicieron en la segunda semana	Edynson Tola	Registrar un avance semanal de lo que se ha hecho, mostrar los problemas encontrados y sus posibles soluciones, mostrar si el proyecto avanza según lo planeado y plantear las tareas que se harán en la siguiente semana	Se redacta la bitácora número dos de lo trabajado esta semana
Programación de interfaz de control en App Inventor	Confección de botones y menús mediante bloques	José Yampara Felipe Diaz	Brindar una manera más eficiente para poder movilizar al robot utilizando las funciones hechas en python en una interfaz agradable	Se logra el control manual mediante Bluetooth

Realización de la bitácora número tres	Se detallaron las tareas que se hicieron en la tercera semana	Edynson Tola	Registrar un avance semanal de lo que se ha hecho, mostrar los problemas encontrados y sus posibles soluciones, mostrar si el proyecto avanza según lo planeado y plantear las tareas que se harán en la siguiente semana	Se redacta la bitácora número tres de lo trabajado esta semana
Confección del primer informe de avance	Documentar lo que se va hacer en el proyecto	Felipe Diaz Edynson Tola	Dar e informar al usuario de que es lo que trata el proyecto y documentar a detalles el progreso que hubo de este	Primer informe de avance
Planificación del nombre del proyecto	Brindarle un nombre al proyecto para poder ser identificado	Todo el grupo	Hacer que el proyecto tenga una denominación para que se pueda distinguir sobre otros proyectos	Facilitador logístico eléctrico transportador de equipo (F.L.E.T.E)
Programación de movimientos	Codificación de las funciones del robot	Pablo Andia Martin Acevedo	Brindar movimiento al robot mediante código python usando Pybricks	Funciones funcionales para el movimiento del carrito de compras
Realización de la presentación	Confección de una presentación en Canva	Edynson Tola Felipe Diaz	A partir del informe crear un resumen mediante una presentación del proyecto y con este explicar lo que se realizó en el proyecto	Primera presentación
Realización de Bitácoras semanales	Recopilación de los avances realizados en la junta grupal de la semana	Edynson Tola	Registrar un avance semanal de lo que se ha hecho, mostrar los problemas encontrados y sus posibles soluciones, mostrar si el proyecto avanza según lo planeado y plantear las tareas que se harán en la siguiente semana	Bitácoras

3.2. Carta Gantt

Una carta gantt es un tipo de diagrama que representa el progreso que ha tenido el proyecto según los objetivos específicos, trazando así una línea de tiempo por cada objetivo planteado, mostrando así el avance de cada uno de estos. A continuación se mostrará la carta gantt de el proyecto:

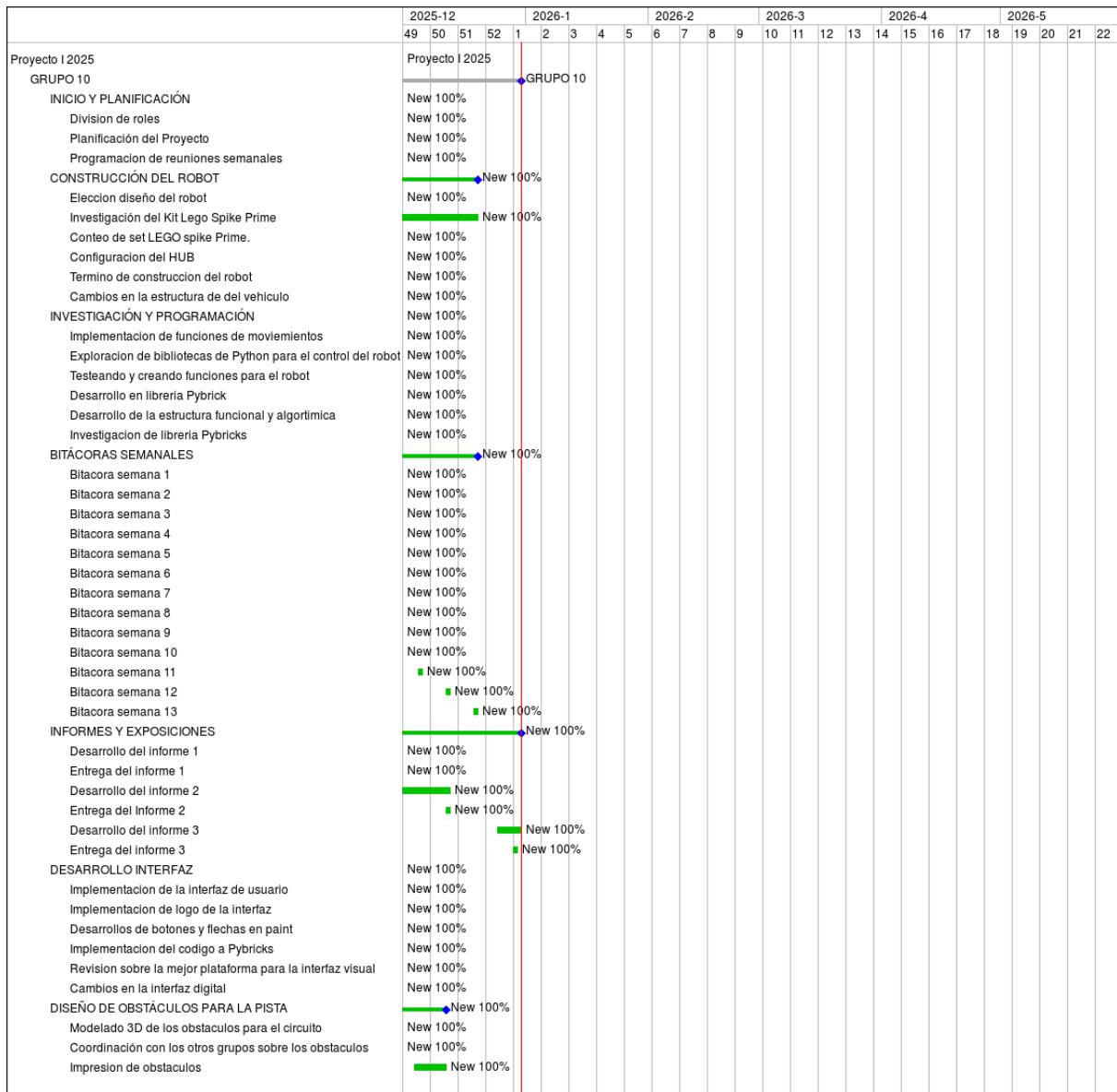


Figura 1: Carta Gantt

3.3. Gestión de Riesgos

Durante el transcurso del proyecto pueden presentarse diferentes tipos de riesgos que necesitan una respuesta inmediata del equipo para ser erradicados. Los posibles riesgos se han clasificado en cuatro niveles según el impacto que puede llegar a tener en el proyecto:

1. **Daño momentáneo** : Riesgo menor que no para el desarrollo del proyecto. Generalmente tiene solución inmediata.
2. **Daño menor** : Riesgo de poca importancia pero que es reiterativo. Puede solucionarse en cualquier momento.
3. **Daño relevante** : Riesgo que retrasa el correcto desarrollo del proyecto, se debe resolver a la mayor brevedad posible.
4. **Daño Crítico** : Riesgos que deben solucionarse de forma inmediata, de lo contrario puede provocar la detención del proyecto.

Tabla 3: Riesgos, gravedad y acciones remediales

Riesgo	Nivel de Gravedad	Acción Remedial
Personal faltando al horario asignado de trabajo	4	Preguntar la causa de su ausencia, para poder gestionar otra reunión en horarios disponibles
Falla de registro en la plataforma redmine	4	Comunicar al administrador de la página para encontrar una solución.
Recibir equipo defectuoso	4	Conseguir un reemplazo del equipo con el encargado de las piezas
Horario insuficiente para el cumplimiento de tareas en conjunto	3	Coordinación de una reunión fuera de clases, en el mejor horario posible para cada integrante
Desempeño del robot no es eficiente	2	Utilizar conocimientos propios para reconocer las fallas o de última instancia buscar información en línea
Incomprensión de fallo con bibliotecas	2	Buscar ejemplos en medios oficiales para solucionar el problema
Error en la codificación	2	Investigar el origen del error e intentar resolver el problema buscando información en internet
Atraso en el cumplimiento de tareas	2	Priorizar tareas más importantes para agilizar la productividad del proyecto. Agendar reuniones en horarios libres si es necesario
Ausencia de piezas	1	Verificar si fue ausencia de fábrica o error de algún integrante, y buscar la pieza perdida
Dificultades con la conexión wifi	1	Usar cable ethernet o compartir red por datos móviles

4. Planificación de los Recursos

En esta sección se detallan los recursos utilizados en el proyecto resaltando los materiales tecnológicos necesarios para la ejecución exitosa del proyecto. Se ha realizado un conteo de los materiales que dispondremos en uso para el proyecto.

4.1. Hardware

- Set Lego Spike Prime, necesario para la correcta construcción del robot.
- Computador con el sistema operativo necesario para poder codificar las funciones de movimiento del robot.
- Celulares.
- Cable Ethernet.

4.2. Software

- Sistema operativo Windows.
- Redmine.
- Visual Studio Code.
- Paint.
- Py Bricks.
- Whatsapp.
- Gmail.

4.3. Estimación de Costos

En cuanto a la estimación de costos, estos se presentan en las siguientes tablas a fin de exponer esta información de manera más dinámica y clara:

Tabla 4: Costo de Hardware

Producto	Cantidad	Precio unitario
Set Lego Spike Prime	1	\$ 370.000 CLP
Notebook Acer Aspire Lite	1	\$ 499.990 CLP
Notebook Lenovo v14 g2 ALC AMD Ryzen 5	4	\$ 699.990 CLP
Mouse Logitech MX Master 3S Grafito	1	\$ 89.990 CLP
Celular Xiaomi POCO X4 PRO 5G	1	\$ 399.990 CLP
Total:	8	\$ 2.059.960 CLP

Tabla 5: Costo de Software

Producto	Cantidad	Precio unitario
Windows 11 home - Licencia	5	\$ 9.990 CLP
Plataforma Redmine	5	\$ 0 CLP
Visual Studio Code	2	\$ 0 CLP
PyBricks	1	\$ 0 CLP
Whatsapp	5	\$ 0 CLP
Gmail	5	\$ 0 CLP
Total :	24	\$ 159.990 CLP

Tabla 6: Costo de Trabajador*Fecha de inicio: 11-09-2025*

Rol	Horas	Precio / Hora
Jefe de proyecto	42 horas	\$ 27.342 CLP
Programador	33 horas	\$ 17.000 CLP
Ensamblador	9 horas	\$ 4.460 CLP
Diseñador	15 horas	\$ 3.385 CLP
Documentador	24 horas	\$ 4.500 CLP
Total :	123 horas	\$ 1.908.279 CLP ((Horas * Precio))

Notas:

- *Las horas de trabajo, son las horas de trabajo en el horario de clases*
- *Las horas extras son horas trabajadas fuera del horario de clases, pero dentro del departamento Ingeniería Civil en Computación e Informática.*

Tabla 7: Total de Costo

Costos	Monto
Hardware	\$ 2.059.960 CLP
Software	\$ 9.990 CLP
Empleados	\$ 1.908.279 CLP
Total :	\$ 4.128.229 CLP

5. Análisis y diseño

5.1. Especificación de requerimientos

En cuanto a los requerimientos del proyecto, para lograr una mejor comprensión de estos, es necesario reiterar el objetivo general del proyecto: crear un robot capaz de transportar carga de manera remota y segura, simulando de tal manera un entorno real. Esta iniciativa surge para una empresa minera interesada en automatizar procesos de transporte de carga con el fin de reducir la exposición al riesgo de sus trabajadores, específicamente de los operadores de carga.

Bajo este contexto, se identifica al cliente como la empresa minera y a los usuarios como los operadores de carga que manejan el sistema a distancia.

Teniendo en consideración lo anteriormente señalado, se da introducción a los aspectos técnicos que permitirán alcanzar dicho objetivo.

5.1.1. Requerimientos funcionales

RF1 Movilidad: El robot debe ser capaz de poseer la movilidad de un vehículo normal, es decir, debe ser capaz de: virar a la izquierda, virar a la derecha, avanzar, retroceder y detenerse.

RF2 Transporte: El robot debe ser capaz de transportar objetos en su contenedor especializado.

RF3 Ejecución de órdenes: El robot debe ser capaz de responder a las órdenes ingresadas por el usuario.

RF4 Indicador de estado operativo: El robot debe ser capaz de indicar a través de su interfaz gráfica su estado de conexión.

RF5 Velocidad: Se debe poder modificar la marcha o potencia con que avanza el robot e indicar esta.

5.1.2. Requerimientos no funcionales

RNF1 Disponibilidad: El robot debe ser capaz de poseer plena facultad de sus funciones por al menos 1 hora de trabajo continuo.

RNF2 Robustez: El sistema programado para ejecutar el movimiento del robot debe ser capaz de gestionar de manera efectiva el 90% de los fallos y evitar que a causa de estos, se detengan las funciones del vehículo en su totalidad.

RNF3 Rendimiento: El sistema debe ser capaz de cumplir las órdenes ingresadas por el usuario con una latencia de no más de 1 segundo, esto con el fin de no entorpecer la experiencia del usuario con tiempos de espera muy largos.

RNF4 Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de utilizar, para que el 100% de los usuarios pueda realizar una tarea designada tras una breve inducción.

RNF5 Componentes: Los componentes de movimiento del robot, tales como ruedas, motores, estructura en general, deben estar en buena condición y funcionar de manera correcta y coordinada para que este pueda efectuar su normal funcionamiento el 99.9% del tiempo.

5.2. Arquitectura del software

La arquitectura de software utilizada en el presente proyecto sigue los lineamientos de una arquitectura cliente-servidor. La comunicación entre los distintos sistemas de software se realiza de manera directa, sin la implementación de mecanismos de seguridad adicionales. Como consecuencia, el intercambio de información entre los sistemas se lleva a cabo de forma fluida, secuencial y sin intermediarios.

A continuación se muestra un diagrama de cómo se comunican estos sistemas y una breve explicación de estos mismos:

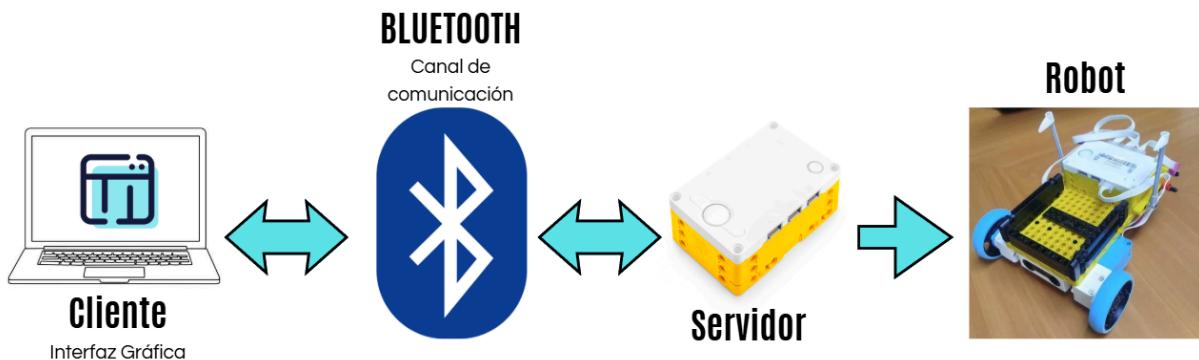


Figura 2: Diagrama de la arquitectura cliente-servidor de F.L.E.T.E.

En la figura 2 se muestra la arquitectura cliente-servidor que sigue el robot, donde se distribuyen las tareas para que el servidor ejecute las tareas de control y movimiento y el cliente interactúe con el usuario mediante la interfaz, llevando a cabo un mayor rendimiento, y adaptabilidad asegurando que se pueda cambiar la lógica de control, cambiar la interfaz gráfica o componentes del hardware sin comprometer al sistema completo.

En cuanto a lo anteriormente mencionado y mostrado en la figura 2 se explican a continuación:

Cliente: Aquí se aloja la interfaz gráfica, permite la interacción entre el usuario y el sistema mediante elementos visuales como ventanas, botones, menús, iconos y cuadros de diálogo. Su función principal es facilitar el uso del sistema, permitiendo al usuario ejecutar acciones, ingresar datos y visualizar información de manera clara e intuitiva, sin necesidad de interactuar directamente con la línea de comandos.

Bluetooth: Su rol es ser el canal de comunicación inalámbrica entre el cliente y el servidor mediante un protocolo de comunicación establecido, además comunica el estado de conexión del sistema. Se envían datos en formato byte de parte del cliente hacia el servidor.

Servidor: Es el encargado de recibir los comandos tipo byte del cliente, los interpreta y traduce en acciones físicas sobre el robot.

Robot: Corresponde al componente de hardware que materializa las órdenes del cliente, realizando así movimientos físicos en el entorno real.

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

Para el diseño inicial se consideró una distribución de botones lo más intuitiva posible, buscando la semejanza con distribuciones óptimas como en los videojuegos (en especial para el pad direccional), a continuación se presenta la distribución de botones seleccionada para la interfaz gráfica:



Figura 3: distribución de botones de la UI

Como se observa en la figura 3, se optó por una distribución fácil de entender e intuitiva, donde los botones en el pad direccional obedecen a la regla de congruencia, es decir, el botón izquierdo del pad direccional da la orden al vehículo de doblar hacia la izquierda, lo mismo con el botón de la derecha del pad direccional, mientras que el botón superior es para dar la orden de avanzar al vehículo, el botón inferior da la orden de retroceder, mientras que el botón central funciona como freno, es decir, detiene el vehículo en su posición actual.

En cuanto al segundo set de botones “Maniobras Especiales” en la actualidad aún se está evaluando activamente qué funciones agregar a la interfaz, por lo que de manera preliminar se cuenta con esa sección para poder agregar funcionalidades extra, como lo pueden ser, modos de manejo variados (velocidad lenta, velocidad media o turbo), una bocina, activación de la cámara frontal, activar un modo estacionamiento, etcétera.

6. Implementación

En este capítulo se detalla la construcción lógica y la arquitectura de software del proyecto "Facilitador Logístico Eléctrico Transportador de Equipo". Se describe cómo se traducen las órdenes del usuario en movimientos físicos del robot a través de la comunicación entre el ordenador y el Hub LEGO Spike Prime.

6.1. Fundamentos de los movimientos

Para llevar a cabo una operación óptima del robot, se espera que este sea capaz de recorrer un metro en un tiempo aproximado de dos segundos, por ende, la velocidad esperada mínima que el robot debe ser capaz de generar es de 0,5 mt/seg, siendo esta la velocidad mínima admisible.

Tomado en consideración la velocidad mínima admisible, un tramo de aproximadamente 10 metros debería ser recorrido en un tiempo mínimo de 20 segundos (Cálculos obtenidos dividiendo la distancia a recorrer; 10 metros por la velocidad mínima admisible).

Teniendo esto en consideración, se seguirán efectuando testeos para evaluar un aumento o disminución de la velocidad mínima admisible.

6.2. Descripción del sistema

La arquitectura del sistema sigue un modelo de ejecución remota mediante una interfaz, donde el procesamiento de movimientos y de conexión son mediante un código base implementado en Python, dando como resultado que el robot reciba las instrucciones y pueda realizar movimientos.

6.2.1. Cliente

El "Cliente" en esta arquitectura corresponde al ordenador (Notebook) que ejecuta la aplicación principal desarrollada en Python. Su función es actuar como el punto de origen de las instrucciones.

6.2.2. Servidor

En esta arquitectura, el servidor corresponde al Hub inteligente LEGO Spike Prime ejecutando el firmware de Pybricks.

Funcionamiento:

1. Estado de Espera (Idle): El Hub se mantiene en modo de escucha Bluetooth "advertising".
2. Recepción: Al recibir una solicitud de conexión del Cliente (pybricksdev), acepta el stream de datos que contiene el código Micro Python compilado.
3. Ejecución: El intérprete interno del Hub ejecuta el script recibido (por ejemplo el adelante.py). Durante este tiempo, el Hub toma el control exclusivo de los puertos A, B, C y E.
4. Finalización: Al terminar el script, el Hub libera los recursos y retorna automáticamente al estado de espera, listo para la siguiente instrucción.

6.2.3. Interfaz gráfica de usuario (GUI)

La interfaz gráfica (GUI) ha sido desarrollada utilizando la librería estándar **Tkinter** de Python, implementando un diseño moderno tipo "*Flat*" para maximizar la inmersión y la usabilidad. La GUI actúa como el panel de control principal del sistema F.L.E.T.E, permitiendo el acceso remoto al robot en tiempo real.

La interfaz se estructura en cuatro secciones funcionales principales:

1. Encabezado de Estado y Conexión:

- Situado en la parte superior (barra azul cian), contiene el nombre del hub "GP-10".
- Incluye un indicador de estado visual que cambia de color (Gris: *Offline*, Naranja: *Buscando*, Amarillo: *Cargando*, Verde: *Online*) según la retroalimentación del hilo secundario de conexión.
- Botón "CONECTAR" que inicia el hilo de búsqueda Bluetooth y botón "X" para el cierre seguro de la aplicación y destrucción de procesos.

2. Caja de Cambios Virtual:

- Implementa tres modos de velocidad seleccionables mediante botones o atajos de teclado (*Shift* para subir, *Ctrl* para bajar):
 - 1^a Lenta (400°/s): Para maniobras de precisión y estacionamiento.
 - 2^a Normal (800°/s): Velocidad crucero por defecto.
 - 3^a Sport (1500°/s): Para desplazamientos rápidos en tramos rectos.

3. Panel de Control de Tracción (Joystick Virtual):

- Disposición de botones intuitivos basada en el estándar direccional WASD.
- Los botones poseen estados visuales reactivos que se iluminan tanto al hacer clic con el mouse como al presionar la tecla física correspondiente.
- Incluye control de bocina ("Espacio") y frenado de emergencia.

4. Accesos Rápidos (Maniobras Complejas):

- Botones programados para enviar macro-comandos que combinan tracción y dirección simultáneamente.
- Permite realizar curvas cerradas hacia adelante y hacia atrás sin necesidad de coordinar dos teclas manualmente.

5. Consola de Registro (Log):

- Panel inferior con fondo oscuro que muestra en tiempo real los eventos del sistema, como cambios de marcha, intentos de conexión y

comandos serializados enviados

7. Resultados

7.1. Estado actual del proyecto

En cuanto al estado actual del proyecto, se puede decir con total seguridad que se ha alcanzado hasta el momento la funcionalidad deseada, puesto que ya se cuenta con una interfaz funcional y una operación eficiente del robot en distintos escenarios, lo cual desde la perspectiva planteada en un inicio, la cual exhibe (de manera resumida):

“La creación de un robot con el set de piezas LEGO Spike Prime, el cual será controlado mediante una interfaz gráfica, mediante el uso de Python, con la finalidad de demostrar la automatización de transporte de carga en ambientes laborales.”

Se puede afirmar que el desarrollo del proyecto se encuentra muy bien encaminado. Esto se refleja en la cantidad de objetivos específicos que se han cumplido hasta el día en el cual se está redactando este reporte, a continuación se presenta una lista de los objetivos específicos planteados en la primera etapa y el estado actual en el que se encuentran dichos objetivos:

- Estudiar los componentes del set Lego Spike Prime

Estado actual: cumplido.

En una fase temprana, se tuvo que evaluar el alcance y las distintas posibilidades del set de LEGO Spike Prime.

- Proponer y seleccionar el mejor prototipo para el proyecto

Estado actual: cumplido.

Se seleccionó de manera unánime entre los miembros del grupo, el diseño de robot que se deseó llevar a cabo.

-Construir el prototipo seleccionado

Estado actual: cumplido.

El diseño seleccionado logró ser ensamblado sin mayores percances.

-Investigar y analizar las librerías disponibles en Python

Estado actual: cumplido.

Si bien ya se encuentra terminada la investigación respecto a las diferentes tecnologías que se consideraron para programar el robot, a día de hoy el equipo sigue en busca de soluciones más eficientes e intenta mantenerse al día con las últimas tendencias, por lo que de ser necesario, si surgen actualizaciones que permitan un mejor rendimiento del robot, se optará por aprender e implementar dicha tecnología.

-Programar las funciones del robot (avance, retroceso, giro y frenado), haciendo uso de buenas prácticas de programación como utilizar algoritmos que ayuden a procesar datos de manera eficiente y funciones que puedan ser útiles para los movimientos básicos del robot

Estado Actual: cumplido.

Las funciones de movimiento del robot se encuentran listas en su totalidad, se logró implementar con éxito lo planeado en el informe anterior, características las cuales

eran implementar cambios de velocidades en la interfaz (es decir, controlar la velocidad de las ruedas entre: lento, normal y rápido) y la bocina del móvil.

-Desarrollar una interfaz gráfica con Python y Tkinter que reciba las instrucciones del robot seleccionado.

Estado Actual: cumplido

Actualmente la funcionalidad de la interfaz gráfica y su interacción con el código (back-end) ya se encuentra lista y operativa, se han llevado a cabo pruebas de movimiento y han resultado exitosas, el hecho de que el objetivo no esté cumplido en su totalidad, es debido a que, en este momento, el equipo está en la búsqueda activa de una solución más eficiente para acortar el tiempo de delay entre la acción seleccionada por el usuario en la interfaz gráfica, hasta que el robot efectivamente la ejecuta. Por lo que no se dará por terminado este objetivo específico hasta lograr una baja en el delay (delay actual: 1.5 segundos, delay deseado: <1 segundo).

Tomando en consideración los avances llevados a cabo hasta el momento, se afirma con total certeza que el proyecto lleva un 100% de avance.

7.2. Problemas encontrados y solucionados

En cuanto a este apartado, se mostrarán los problemas enfrentados hasta esta etapa de desarrollo del proyecto, y las medidas reparatorias implementadas para proseguir con la correcta ejecución de las actividades.

Para lograr una mejor visualización de estos riesgos, se presenta una tabla explicativa que muestra la frecuencia que tuvo el problema en cuestión, el tipo de problema, y la medida reparatoria correspondiente:

Tabla 8: Frecuencia de riesgos identificados y acciones remediales

Frecuencia	Riesgos	Acción remedial
+20	Error en la codificación	Investigar el origen del error e intentar resolver el problema buscando información en internet
15	Dificultades con la conexión wifi	Usar cable ethernet o compartir red por datos móviles
8	Personal faltando al horario asignado de trabajo	Preguntar la causa de su ausencia, para poder gestionar otra reunión en horarios disponibles
7	Incomprensión de fallo con bibliotecas	Buscar ejemplos en medios oficiales para solucionar el problema

3	Atraso en el cumplimiento de tareas	Priorizar tareas más importantes para agilizar la productividad del proyecto. Agendar reuniones en horarios libres si es necesario
1	Vehículo se mueve con dificultad o es incapaz de moverse libremente	Probar a intercambiar dentro de lo posible piezas que generan el problema

Se puede deducir de manera fehaciente, que hasta el momento, no han ocurrido problemas que comprometan el avance del proyecto de manera significativa, por lo que sin ninguna duda, se está yendo por un buen camino. De manera secundaria, cabe agregar que, las medidas reparatorias han funcionado de manera efectiva, puesto que nos permiten resolver de manera rápida los problemas a medida que aparecen, por lo que se puede concluir que se ha logrado un correcto avance del proyecto.

Se añade también, la aparición de un problema que no se tuvo en consideración al momento de planificar la gestión de riesgos, el problema que se presentó, ocurrió en la fase de pruebas de funcionamiento el robot, donde se observa una carencia de agarre por parte de las ruedas del móvil, lo que terminó ocasionando un movimiento errático y poco controlado del movimiento del vehículo, además que de manera recurrente, este se atoraba en determinados puntos de la pista, como lo pueden ser desniveles de obstáculos o desniveles propios de la pista.

Tomando en cuenta lo siguiente, se debió optar por una medida remedial para solventar el problema, para lo cual se optó por cambiar las ruedas del vehículo, donde se instalaron ruedas de mayor radio y el problema se solucionó a la brevedad.

8. Prueba de funcionamiento del sistema

A continuación se mostrarán los resultados de las pruebas a las que se sometió el robot, en conjunto con la descripción detallada de estas, se mencionan los diversos problemas que se presentaron en esta fase y cómo se solucionaron, además de exponerse los resultados obtenidos.

8.1. Descripción de la prueba de funcionamiento

En cuanto al instrumento seleccionado para poder medir las capacidades del robot, se optó, en conjunto con colegas de la misma clase, crear un circuito con obstáculos, circuito el cual se planificó y creó en conjunto. Cabe recalcar que todos los involucrados en la planificación de este circuito cuentan con un proyecto similar al propio (vehículos terrestres de cuatro ruedas), por lo que, como bien se mencionó antes, se trabajó en conjunto, donde se tuvo que acordar; el tipo de obstáculos que se usarían, el trayecto a recorrer, y la manera en que se dispondrán los obstáculos en el circuito.

A continuación, se presenta el diseño de circuito que se utilizó para poder llevar a cabo las pruebas de funcionamiento:

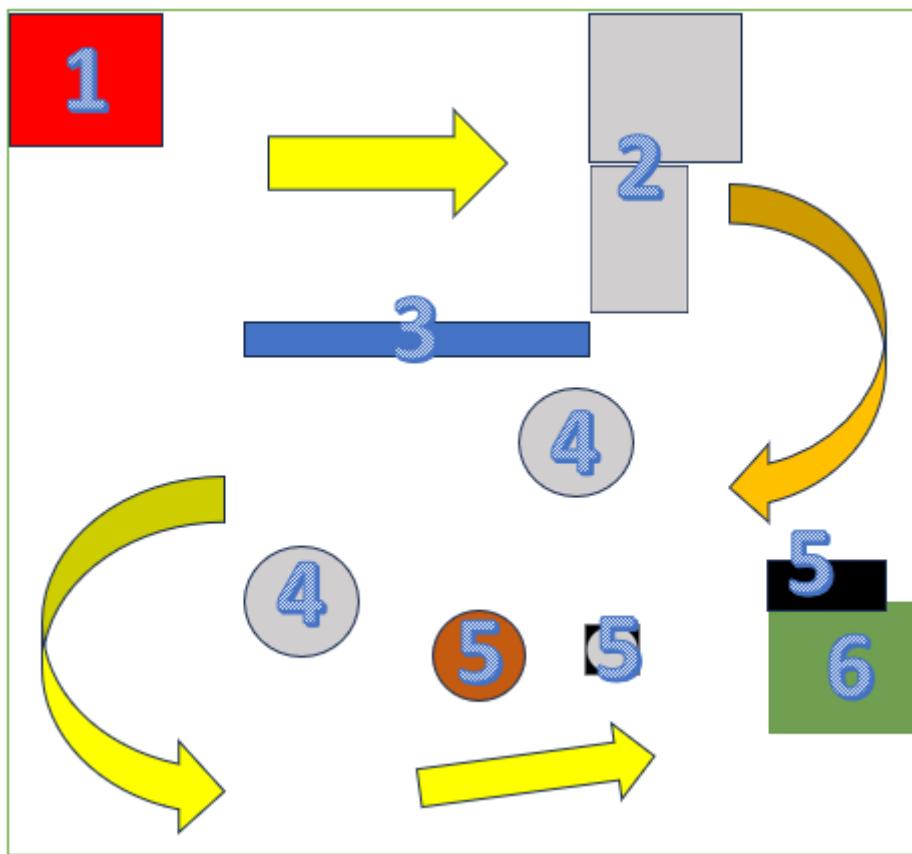


Figura 4: Circuito de obstáculos planeado

Se procederá a explicar cada uno de los elementos enumerados en la figura, esto con el fin de entender de mejor manera los propósitos de estos obstáculos, y aclarar el rol que cumplen en el circuito:

Flechas amarillas: Representan el sentido el cual se debe seguir para recorrer completamente el circuito.

Objeto 1: Es un cuadro rojo pintado en la pista, que representa el punto inicial desde donde partirá el vehículo.

Objeto 2: Objetos 3D que representan desniveles o “lomos de toro” dispuestos en el circuito, uno grande y otro pequeño, donde cada uno presenta una altura de dos centímetros y un centímetro respectivamente, se mandaron a hacer con diferentes alturas para poder así favorecer a los vehículos que no sean capaces de sortear el desnivel más grande. El desnivel grande es el cuadro de mayor tamaño ubicado más al norte, mientras que el desnivel pequeño se encuentra debajo de este, además su superficie es menor respecto al grande.

Objeto 3: Es un objeto 3D puesto con el propósito de delimitar zonas. Mide aproximadamente veinte centímetros de largo, cinco centímetros de ancho y diez de alto.

Objeto 4: Son objetos 3D cilíndricos, los cuales pretenden obstaculizar el camino, ambos obstáculos son cilindros de igual medida, de radio aproximado de tres centímetros y cinco centímetros de altura.

Objeto 5: Son objetos 3D que se improvisaron al momento de armar el circuito, estos objetos pretenden obstaculizar el circuito y presentan formas y medidas variables, donde aproximadamente el volumen promedio entre los tres objetos es de nueve centímetros cúbicos.

Objeto 6: Es un cuadro verde pintado en la pista, que representa el punto de final del circuito. El vehículo al detenerse en esta zona, se da por terminada la prueba de funcionamiento.

A continuación se expone una imagen que muestra el resultado obtenido al intentar recrear el circuito planeado en la vida real.

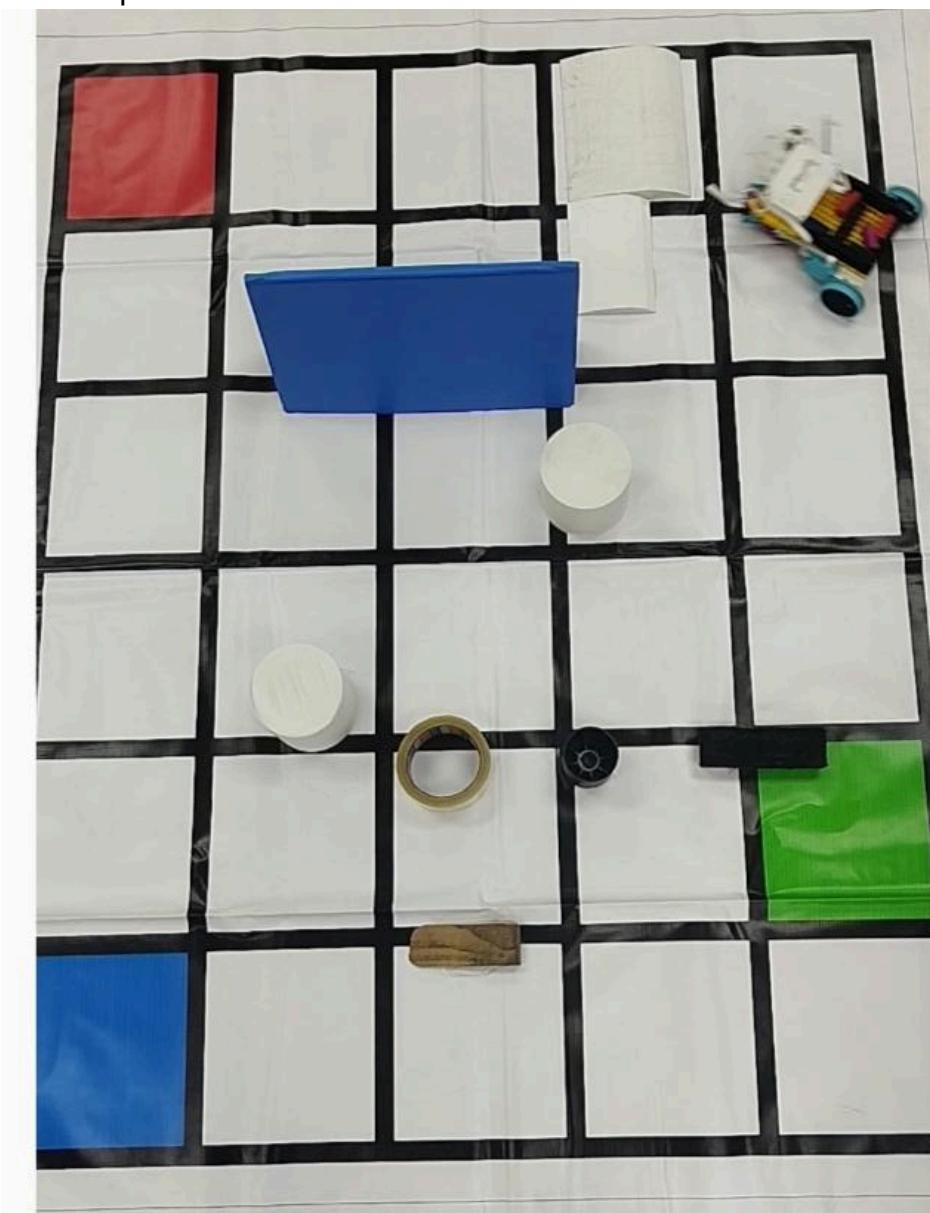


Figura 5: Circuito de obstáculos realizado

8.2. Resultados observados para la prueba de funcionamiento

Durante la ejecución de las pruebas funcionales mínimas, se observaron los siguientes resultados basados en la última versión del software "GP-10":

- **Estabilidad de la Conexión:** La implementación de la arquitectura Cliente-Servidor asíncrona resultó exitosa. La interfaz gráfica se mantuvo funcionando perfectamente durante el proceso de escaneo y conexión Bluetooth, validando la eficacia del hilo secundario (*Worker Thread*).
- **Control de Latencia:** Se observó una respuesta casi inmediata entre la pulsación de la tecla (PC) y la acción del robot. El "Evaluador de Estado" implementado en el código eliminó el conflicto de comandos al presionar múltiples teclas (ej: W+D), permitiendo transiciones suaves entre avanzar recto y girar, sin que el robot se detuviera o realizara movimientos erráticos.
- **Gestión de Velocidad:** La caja de cambios virtual funcionó correctamente. Al cambiar a "1^a Lenta", el robot redujo su velocidad, permitiendo sortear los obstáculos cilíndricos del circuito con alta precisión. En "3^a Sport", el robot alcanzó su velocidad máxima en las rectas del circuito.
- **Sistema de Seguridad:** La lógica de eventos *Press/Release* funcionó según lo esperado. Al soltar cualquier tecla o botón de la interfaz, el robot detuvo sus motores y centró la dirección automáticamente, cumpliendo con los requisitos de seguridad operativa para detener el vehículo ante la inacción del operador.
- **Feedback al Usuario:** La consola de registro y los indicadores visuales de los botones permiten al operador saber en todo momento qué comando estaba recibiendo el robot, facilitando la depuración durante la prueba.

9. Conclusión

El proyecto permitió comprender cómo la **automatización y el control remoto** pueden mejorar significativamente la seguridad de los trabajadores o empleadores. Logrando mejorar la productividad y eficiencia del transporte de cargas.

Mediante el uso del **LEGO Spike Prime** y una interfaz desarrollada con **Python y Tkinter**, logramos simular un sistema de transporte que minimice la intervención humana en zonas peligrosas, y así prevenir accidentes en situaciones laborales, ya sea en obras u otro tipo de trabajo relacionado con la construcción.

Se menciona también que, hasta el momento, no se han presentado problemas de gravedad considerable que impidan el correcto término del proyecto, por lo que, se puede deducir que se está llevando el proyecto por el camino correcto, se prevé que el proyecto puede estar en su totalidad listo mucho antes de la entrega final. Teniendo esto en consideración, el equipo se enfocará en pulir cada detalle para lograr un mejor desempeño en las fases finales del proyecto.

10. Referencias

Acer. (s.f.). *Notebook Acer Aspire Lite AI15-41p-r0zy*. MercadoLibre. Recuperado el 25 de diciembre de 2025, de <https://www.mercadolibre.cl/notebook-acer-ai15-41p-r0zy-1-32gb-512gb-ssd-156-pulgadas-color-gris/p/MLC44976382>

GetOnBrd. (s.f.). *Insights: Success rates & salaries*. Recuperado el 25 de diciembre de 2025, de <https://insights.getonbrd.com/#/reports/success-rates-salaries>

Indeed. (s.f.). *Sueldos de Jefe de proyectos TI en Chile*. Recuperado el 25 de diciembre de 2025, de <https://cl.indeed.com/career/jefe-de-proyectos-ti/salaries>

Kitrum. (s.f.). Client-Server Architecture: Advantages and Disadvantages. Recuperado el 25 de diciembre de 2025, de <https://kitrum.com/blog/client-server-architecture-advantages-and-disadvantages/>

LEGO Education. (s.f.). *LEGO® Education SPIKE™ Prime Set*. Recuperado el 25 de diciembre de 2025, de <https://www.lego.com/es-us/product/lego-education-spike-prime-set-45678>

Logitech. (s.f.). *Mouse wireless Logitech MX Master 3S grafito*. MercadoLibre. Recuperado el 25 de diciembre de 2025, de <https://www.mercadolibre.cl/mouse-wireless-logitech-mx-master-3s-grafito/p/MLC19473530>

OPC Store. (s.f.). *Notebook Lenovo V14 G2 ALC AMD Ryzen 5*. Recuperado el 25 de diciembre de 2025, de <https://opcstore.cl/products/notebook-lenovo-v14-alc-8gb-ssd-512gb-14-hd-w10-pro>

Robert Half. (s.f.). *Guía Salarial Tecnología*. Recuperado el 25 de diciembre de 2025, de <https://www.roberthalf.com/cl/es/insights/guia-salarial/tecnologia>

ROBOTIX Hands-on Learning. (2014). *Manual de usuario: Spike Prime [PDF]*. <https://www.robotix.es/es/recursos-gratuitos>

Softpro. (s.f.). *Windows 11 Home - Licencia*. Recuperado el 25 de diciembre de 2025, de <https://softpro.cl/producto/windows-11-pro/>

TuSalario.org. (s.f.). *Salario de Ensambladores y Documentadores*. Recuperado el 25 de diciembre de 2025, de <https://tusalario.org/Chile>

Universidad de Tarapacá. (2025). *Guía para la creación del proyecto*. Intranet UTA. <https://portal.uta.cl/sign-in?redirectURL=%2Fhome>

Xiaomi. (s.f.). *Celular Xiaomi POCO X4 PRO 5G*. MercadoLibre. Recuperado el 25 de diciembre de 2025, de <https://listado.mercadolibre.cl/xiaomi-pocophone-poco-x4-pro-5g>