UNIVERSIDAD DE TARAPACÁ
Universidad del Estado

Ingeniería
Computación e Informática

CLASIFICADOR DE COLORES LEGO

Alumnos: Ariel Ortega
Constanza Serrano
Carla Flores
Jerry Quiroga
Martín Chávez
Asignatura: Proyecto I
Profesor: Baris Klobertanz

Índice de
CONTENIDOS

Índice de
CONTENIDOS

INTRODUCCIÓN

Nuestro proyecto simula un selector de minerales en una mina. Diseñamos un robot con LEGO SPIKE Prime que identifica y separa materiales por color, igual que se haría en un proceso minero real. Lo controlamos mediante una aplicación creada por nosotros en Python, que permite que el robot tome decisiones inteligentes y sea muy preciso al mover los "minerales".

OBJETIVOS DEL PROYECTO

1. Objetivo General:

- Construir y programar un robot capaz de reconocer y distribuir colores mediante una interfaz en Python y los componentes del robot (sensores y motores).

2. Objetivos Específicos:

O.E 1: Experimentar con el hardware del set LEGO SPIKE Prime.

O.E 2: Ensamblar un robot estable con sensor de color.

O.E 3: Diseñar una interfaz gráfica de usuario usando Tkinter.

O.E 4: Documentar avances y soluciones en bitácoras semanales.

O.E 5: Coordinar roles y cronogramas del equipo.

O.E 6: Implementar Python y Pybricks para la conexión con Visual Studio.

RESTRICCIONES Y ENTREGABLES

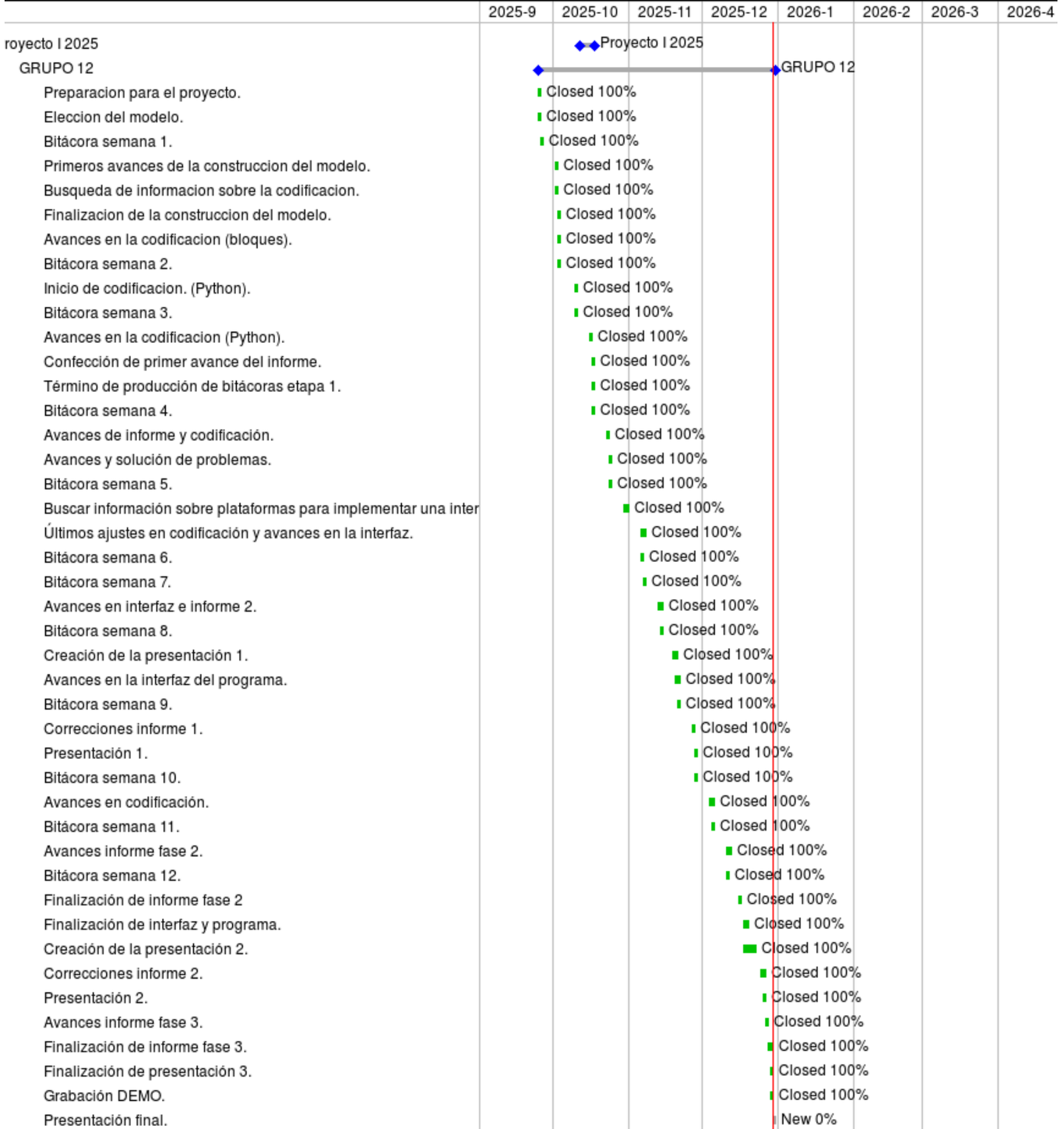
RESTRICCIONES CLAVE:

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

ENTREGABLES PRINCIPALES:

- Bitácoras: Reportes semanales de progreso y obstáculos.
- Carta Gantt: Visualización de tiempos y recursos.
- Informe de Formulación: Estrategia, roles y metodología

CARTA GANTT


2025-9
2025-10
2025-11
2025-12
2026-1
2026-2
2026-3
2026-4
royecto I 2025
Proyecto I 2025
GRUPO 12
GRUPO 12
Preparacion para el proyecto. Closed 100%
Eleccion del modelo. Closed 100%
Bitácora semana 1. Closed 100%
Primeros avances de la construccion del modelo. Closed 100%
Busqueda de informacion sobre la codificacion. Closed 100%
Finalizacion de la construccion del modelo. Closed 100%
Avances en la codificacion (bloques). Closed 100%
Bitácora semana 2. Closed 100%
Inicio de codificacion. (Python). Closed 100%
Bitácora semana 3. Closed 100%
Avances en la codificacion (Python). Closed 100%
Confección de primer avance del informe. Closed 100%
Término de producción de bitácoras etapa 1. Closed 100%
Bitácora semana 4. Closed 100%
Avances de informe y codificación. Closed 100%
Avances y solución de problemas. Closed 100%
Bitácora semana 5. Closed 100%
Buscar información sobre plataformas para implementar una inter Closed 100%
Últimos ajustes en codificación y avances en la interfaz. Closed 100%
Bitácora semana 6. Closed 100%
Bitácora semana 7. Closed 100%
Avances en interfaz e informe 2. Closed 100%
Bitácora semana 8. Closed 100%
Creación de la presentación 1. Closed 100%
Avances en la interfaz del programa. Closed 100%
Bitácora semana 9. Closed 100%
Correcciones informe 1. Closed 100%
Presentación 1. Closed 100%
Bitácora semana 10. Closed 100%
Avances en codificación. Closed 100%
Bitácora semana 11. Closed 100%
Avances informe fase 2. Closed 100%
Bitácora semana 12. Closed 100%
Finalización de informe fase 2 Closed 100%
Finalización de interfaz y programa. Closed 100%
Creación de la presentación 2. Closed 100%
Correcciones informe 2. Closed 100%
Presentación 2. Closed 100%
Avances informe fase 3. Closed 100%
Finalización de informe fase 3. Closed 100%
Finalización de presentación 3. Closed 100%
Grabación DEMO. Closed 100%
Presentación final. New 0%

ORGANIZACIÓN DEL EQUIPO

Distribución de Roles

- Jefe de Proyecto: Representante del equipo, supervisa y organiza la planificación para el progreso del proyecto. (Carla Flores)
- Programador: Encargado de la codificación y el funcionamiento del robot. (Ariel Ortega)
- Ensambladores: Encargado de ensamblaje y armado de piezas.

(Jerry Q. - Martin C.)

- Documentador: Encargado de registrar el avance del proyecto, junto con la redacción de los informes. (Constanza Serrano)

COMUNICACIÓN Y COORDINACIÓN

Mecanismos de Comunicación:

- Comunicación presencial: Resolución inmediata de problemas durante clases mediante reuniones matutinas.
- Discord: Será usado para las reuniones, mediante sus canales de texto y voz.
- Instagram: Mensajería rápida y avisos grupales.
- Redmine: Gestión oficial de documentos y tareas.

PLANIFICACIÓN (ACTIVIDADES)

01 Preparación: Elección del modelo "Separador de Colores" y construcción de la base.

02 Construcción: Finalización del modelo físico al 100%.

03 Codificación Inicial: Pruebas funcionales con bloques para familiarizarse con su lógica.

04 Transición a Python: Traducción de bloques a código e investigación de librerías externas.

GESTIÓN DE RIESGOS

RIESGOS CRÍTICOS Y SOLUCIONES

- Errores en Codificación (catastrófico): Solucionado mediante programación en pares y pruebas unitarias.
- Atrasos en Cronograma (catastrófico): Priorización de tareas esenciales sobre estéticas.
- Fallas de Conexión Bluetooth(circunstancial): Implementación de API Pybricks para optimizar enlace.
- Interferencia de Luz (menor): Calibración al inicio y uso de barreras físicas.
- Falta de documentación tecnica(critico):Investigación en foros externos y migración a librerías con mejor soporte

PLANIFICACIÓN DE RECURSOS

Hardware y Software

1. Hardware:

- Set LEGO® Education SPIKE™ Prime + Expansión.
- Computadores Lenovo (Ryzen 7) para programación e investigacion .

2. Software:

- Python & VS Code: Desarrollo del código fuente.
- Pybricks: Gestión de conexión y firmware.
- Redmine & Office: Gestión y documentación.

PRESUPUESTO ESTIMADO

Estimación de Costos del Proyecto

Producto	Precio (CLP)	Cantidad	Subtotal
Set SPIKE™ Prime de LEGO® Education	$383.196	1	$383.196
Set de Expansión SPIKE™ Prime (LEGO® Education)	$150.375	1	$150.375
Lenovo V14 G2 Ryzen 7	$729.990 c/u	3	$2.189.970
Total Hardware		**4 productos**	**$2.723.541**

Producto	Precio (CLP)
Licencia Microsoft Office	$14.000
Total Software	**$14.000**

COSTOS DE TRABAJADOR

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

COSTOS DE TRABAJADOR

- Periodo de Gestión: 15 semanas (12/09 al 29/12).
- Dedicación Base: 20 hrs semanales por rol (según horario Intranet).
- Horas Extra: Trabajo autónomo fuera de aula para cumplimiento de hitos.
- Metodología de Cálculo:

  Suma de horas (base + extras) multiplicada por la tarifa horaria.

- Tarifas: Basadas en rangos reales del mercado TI en Chile (2025).
- Costo Semanal Operativo: $2.882.000 CLP (incluyendo horas extra de cierre).

Requerimientos funcionales

Automatización y Percepción: El robot utiliza su sensor de color integrado para reconocer cada bloque. Esta información es procesada para que el sistema clasifique y deposite el objeto en el compartimento correspondiente de forma autónoma.

Control y Conectividad: Todo el flujo de trabajo está vinculado a una interfaz gráfica programada en Python. Desde esta interfaz, el robot recibe sus instrucciones y, a la vez, permite un control manual completo. Contamos con botones específicos para controlar la posición, la apertura o cierre del brazo y un botón de parada de emergencia.

Precisión Motriz: Para asegurar que el bloque llegue a su destino, implementamos un movimiento sincronizado de los motores. Esto garantiza que el desplazamiento sea exacto desde el punto de lectura inicial hasta la zona de descarga.

Requerimientos no funcionales

Atributo	Objetivo de Desempeño (Métrica)	Estrategia de Implementación
Disponibilidad	Tiempo de actividad del 95% durante la sesión.	Gestión de energía eficiente del Hub y reconexión automática vía Bluetooth.
Rendimiento	Respuesta a comandos en < 1 segundo.	Uso de protocolos ligeros en Pybricks y procesamiento en hilos (threading) en Python.
Usabilidad	Interfaz con curva de aprendizaje mínima.	Diseño de GUI con botones de acción directa y retroalimentación visual de estado.
Robustez	Tolerancia a fallos del 100% en lecturas críticas.	Mecanismos de reintento, detección de colores fuera de rango y parada de emergencia.

Arquitectura de Software

Modelo Cliente-Servidor

Componentes Principales:

- Cliente: Interfaz gráfica desarrollada en Python para el envío de instrucciones.

- Servidor:El servidor, alojado en el Hub SPIKE Prime, utiliza Pybricks para gestionar el hardware en tiempo real. Funciona como un intérprete asíncrono que traduce comandos Bluetooth en movimientos precisos y lecturas de sensores. Esta arquitectura permite una clasificación autónoma con retroalimentación constante hacia la interfaz.

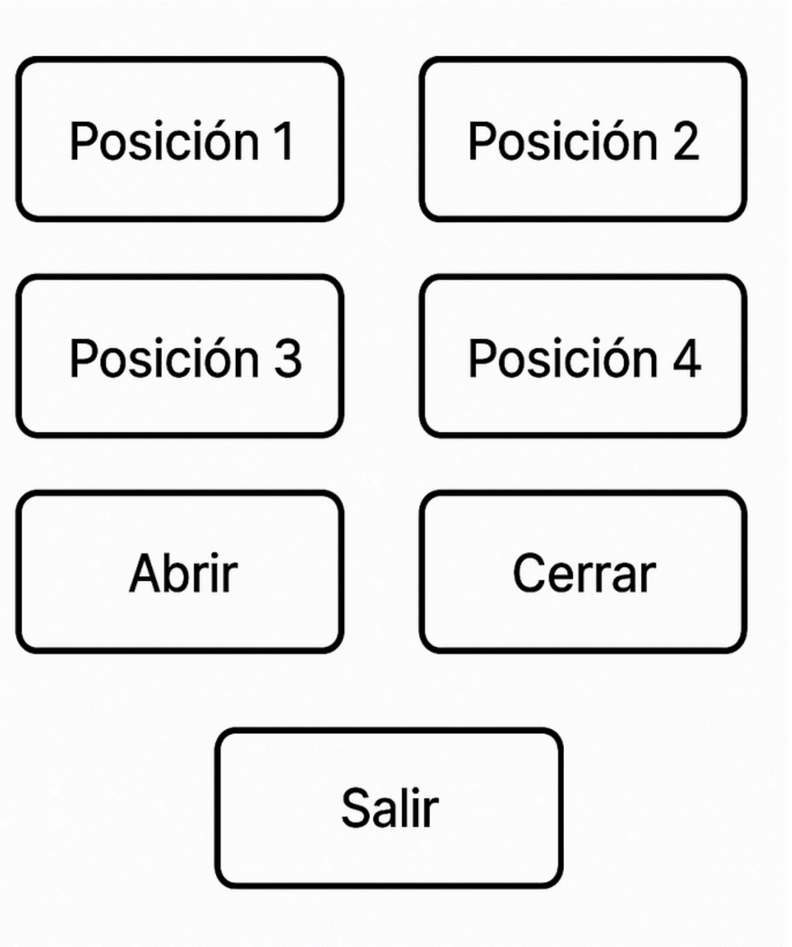
Flujo de Comunicación:

- Interacción del usuario con la interfaz.
- Envío de comandos vía Bluetooth al controlador.
- Ejecución de movimientos por parte del robot.
- Identificación de color y retroalimentación al cliente.

Diseño inicial de la interfaz gráfica de usuario (GUI)

Objetivo: Permitir el control del robot y la visualización de su estado sin necesidad de conocimientos técnicos.

- Componentes del Wireframe:
- Control de Movimiento: 4 botones para desplazarse a posiciones específicas.
- Acción de Carga: Botón para iniciar según el color asignado.
- Manipulación: Botones dedicados para Abrir y Cerrar la compuerta.
- Gestión de Sesión: Botón de salida para finalizar la operación.



Fundamentos de los movimientos

El sistema utiliza un control por tiempo en lugar de sensores de contacto. Al fijar una velocidad angular constante, transformamos la posición de cada color en un intervalo de tiempo preciso.

$$t = \frac{\theta}{\omega}$$

¿Cómo la aplicamos? (El Proceso)

PASO 1: El sensor detecta el bloque. Ejemplo: **AZUL**.

PASO 2: El código busca el ángulo asignado. Ejemplo: $\theta = 180°$.

PASO 3: Con una velocidad constante de $\omega = 90°/s$, el robot calcula:

$$t = \frac{180°}{90°/s} = 2.0 \text{ segundos}$$

PASO 4: El motor se activa exactamente **2 segundos** y se detiene en el objetivo.

Implementación cliente

La interfaz actúa como el centro de control del sistema, enviando instrucciones específicas al robot según la interacción del usuario:

- Controles de Navegación: Incluye 4 botones de posición para el desplazamiento del brazo.
- Gestión de Tareas: Dispone de un botón para iniciar la operación basada en el color detectado.
- Control de Actuadores: Posee botones individuales para Abrir y Cerrar la compuerta.
- Finalización: Cuenta con un botón de Salida para terminar la sesión de forma segura.

```
self.thread = threading.Thread(target=self._thread_main, daemon=True)
self.loop = asyncio.new_event_loop()
```

Se crea un hilo de ejecución independiente del programa principal.
La laptop debe gestionar la interfaz gráfica (GUI) y la conexión Bluetooth al mismo tiempo. Sin este hilo, la ventana se congelaría mientras espera que el robot termine de moverse.

```
program = create_program(cmd)
await hub.run(path)
```

El cliente genera un archivo temporal de MicroPython y lo "inyecta" en la memoria del Hub. El cliente actúa como el cerebro estratégico. En lugar de que el robot tenga todo el código guardado, la laptop decide qué instrucciones enviar según lo que el usuario presione en pantalla.

```
command=lambda: worker.send_command("clasificar")
```

La interfaz captura el clic del usuario y pone una tarea en la "cola" del trabajador Bluetooth.
Es la capa de interacción humana. Traduce un evento físico (clic) en un mensaje digital que el hilo de comunicación enviará al Hub de forma no bloqueante.

Implementación servidor

El servidor actúa como el ejecutor físico del sistema, procesando las órdenes en tiempo real:

- Entorno de Control: Programado íntegramente en Pybricks para una gestión eficiente del hardware.
- Funciones Principales:
  -Ejecución Motriz: Traduce los comandos del cliente en movimientos precisos del motor.
  -Percepción: Encargado de la lectura activa del sensor de color.
- Procesamiento de Instrucciones:
  -Recibe comandos vía Bluetooth desde la interfaz Python.
  -Convierte los datos en acciones: posicionamiento, apertura o cierre de la compuerta.

Implementación GUI

- Diseño Funcional: Estructura simplificada que contiene exclusivamente los botones necesarios para el movimiento y control del mecanismo.
- Abstracción del Código: Diseñada para que cualquier usuario opere el sistema sin necesidad de conocimientos técnicos o comprensión del código subyacente.
- Vinculación Lógica: Cada elemento visual está asociado a una función específica que envía instrucciones inmediatas al robot.

Estado actual del proyecto

Logros Alcanzados (Checked):

- Conectividad: Comunicación bidireccional establecida entre Python y SPIKE Prime.
- Control Remoto: Operación manual funcional desde la interfaz personalizada.
- Precisión de Movimiento: Ejecución exitosa de desplazamientos a las 4 posiciones.
- Mecanismos: Control total de los actuadores para apertura y cierre.
- Diseño: Implementación del prototipo inicial (Layout) de la GUI.

Próximos Pasos (To-do):

- Optimización de Latencia: Refinar el flujo de datos para una respuesta más ágil del robot.
- Pulido de UX/UI: Mejora estética y funcional de la interfaz gráfica para el usuario final.

Problemas encontrados y solucionados

Problema Detectado	Acción Correctiva / Solución
Inestabilidad en la conexión	Migración a Pybricks y actualización a una versión de firmware más estable.
Imprecisión en el posicionamiento	Recalibración de ángulos de giro y pruebas de repetibilidad de los motores.
Obstrucción por cableado	Instalación de sujetadores mecánicos en puntos estratégicos para liberar el rango de movimiento.

PRUEBA DE FUNCIONAMIENTO

DESCRIPCIÓN DE LA PRUEBA DE FUNCIONAMIENTO

- Validación funcional mínima del Robot Clasificador.
- Integración entre hardware (LEGO SPIKE Prime), software de control e interfaz gráfica.
- Simulación de un proceso de automatización minera supervisada por un operador humano.

REQUISITOS PREVIOS

- Estructura de LEGO compuesta por al menos 4 colores diferentes.
- Repetición de al menos un color dentro de la secuencia.
- Uso de un mínimo de 10 bloques para validar continuidad del proceso.

PRUEBA DE FUNCIONAMIENTO

SECUENCIA DE FUNCIONAMIENTO

1. Recepción de carga mediante brazo robótico externo.
2. Detección individual del bloque y reconocimiento de color mediante sensor.
3. Visualización del color en la interfaz gráfica y confirmación del operador.
4. Clasificación automática del bloque en su cuadrante correspondiente.

RESULTADOS OBSERVADOS

- Identificación correcta del 100% de los bloques procesados.
- Comunicación estable entre la GUI y el Hub SPIKE Prime durante toda la prueba.
- Clasificación mecánica precisa, sin errores de posicionamiento.
- Control total del proceso por parte del operador sin modificar el código.

Resultados y Demostración

Resultados Observados:

- Conectividad: Enlace Bluetooth estable durante toda la sesión.
- Precisión: El robot alcanzó las coordenadas de clasificación con éxito.
- Interacción: La GUI procesó y envió comandos en tiempo real, recibiendo retroalimentación visual.

Video Demo (Enlace o Archivo Incrustado):

- Contenido de la demo: Operación desde la GUI, envío de acciones, movimiento del robot y finalización exitosa de la tarea.

Video Demo

Enlace:

https://drive.google.com/file/d/1Cr_3lKPgEbS8j9hBv68YdBY9rOKNvi8V/view?usp=sharing

Manual de Usuario

Conclusión

¡GRACIAS!

UNIVERSIDAD DE TARAPACÁ
Universidad del Estado

Ingeniería
Computación e Informática

CLASIFICADOR DE COLORES LEGO

Alumnos: Ariel Ortega
Constanza Serrano
Carla Flores
Jerry Quiroga
Martín Chávez
Asignatura: Proyecto I
Profesor: Baris Klobertanz