

Clasificador de colores LEGO

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

Asignatura: Proyecto I
Profesor: Baris Klobertanz

Tabla de Contenido

1

Introducción

2

Requerimientos
funcionales

3

Requerimientos no
funcionales

4

Arquitectura de
Software

5

Diseño inicial de la
interfaz gráfica de
usuario

6

Fundamentos de los
movimientos

7

Implementación
cliente

8

Implementación
servidor

9

Implementación GUI

10

Estado actual del
proyecto

11

Problemas
encontrados y
solucionados

12

Prueba de
Funcionamiento

13

Resultados y
Demostración

14

Conclusión

Introducción

Este proyecto integra el uso del kit LEGO Education SPIKE Prime con una interfaz personalizada en Python para desarrollar un robot capaz de identificar y posicionar bloques de colores de manera autónoma. El desarrollo se fundamenta en un enfoque colaborativo y técnico, donde la resolución de problemas y la investigación teórica permiten cumplir con los objetivos de la asignatura a través de un prototipo funcional y escalable.

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:
Disponibilidad	Operación continua sin fallos de energía o conexión.
Rendimiento	Tiempo de respuesta a comandos inferior a 1 segundo.
Usabilidad	Interfaz intuitiva con controles claros para el usuario.
Robustez	Gestión de errores de sensores mediante reintentos y alertas.

Arquitectura de Software

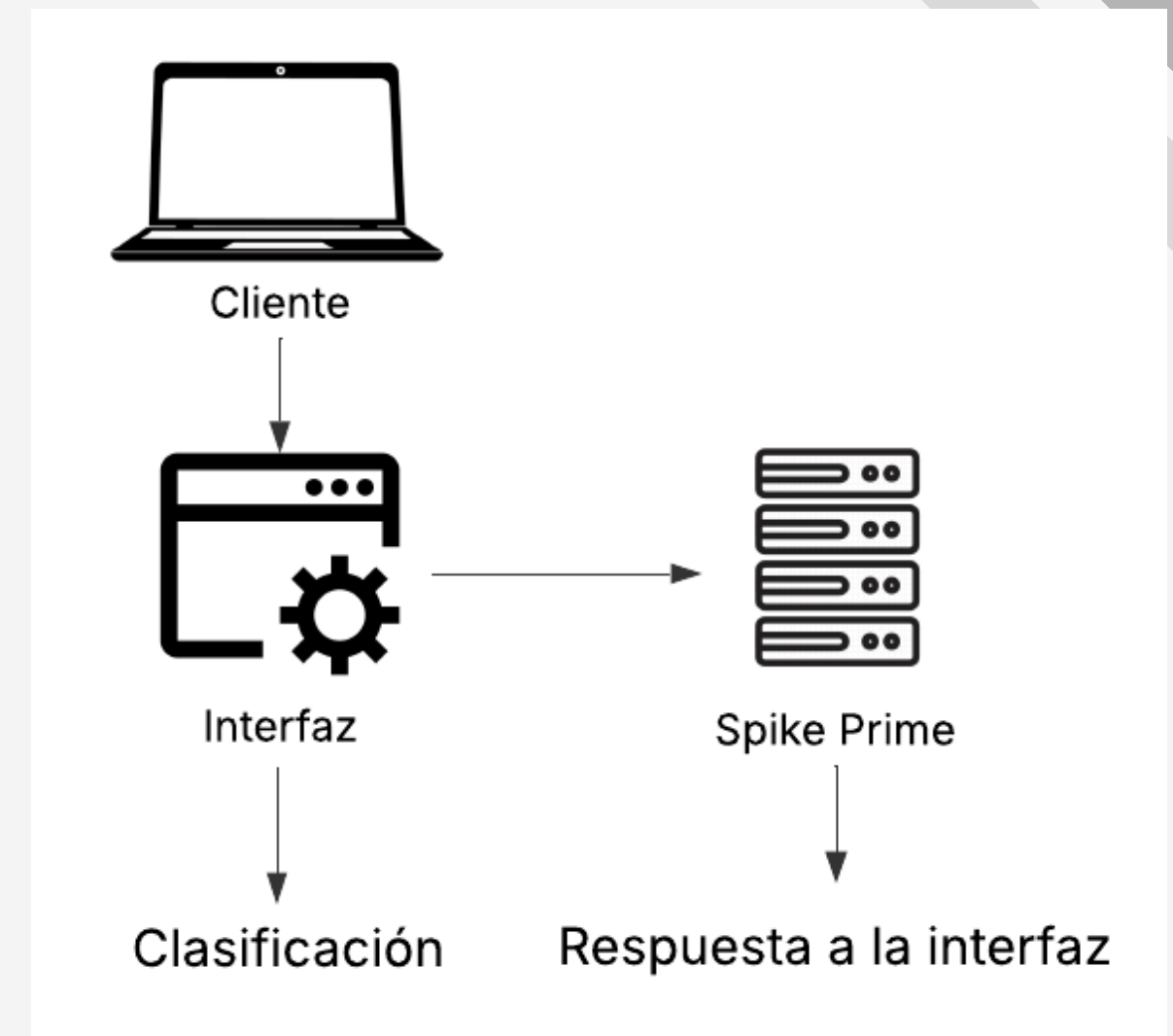
Modelo Cliente-Servidor

Componentes Principales:

- Cliente: Interfaz gráfica desarrollada en Python para el envío de instrucciones.
- Servidor: Hub SPIKE Prime programado con Pybricks, encargado de ejecutar movimientos y lectura de sensores.

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

- Principio Base: Uso de principios de movimiento rotacional para el desplazamiento preciso del brazo hacia las estaciones de color.

Cálculo de Trayectoria:

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

- θ (Theta): Ángulo de giro deseado.
- ω (Omega): Velocidad angular del motor.
- t : Tiempo de ejecución.

- Ejemplo de Aplicación: Para un giro de 90° a una velocidad de $360^\circ/\text{s}$, se estableció un tiempo de ejecución de 0.25 segundos.
- Resultado: Definición de parámetros constantes que garantizan un movimiento estable, repetible y preciso.

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.

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 objetivo.
- 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

Objetivo de la Prueba: Validar la capacidad del robot para ejecutar el ciclo completo de operación según la configuración seleccionada.

Acciones Evaluadas:

- Sincronización inicial mediante la interfaz Python.
- Ejecución de movimientos hacia las posiciones predefinidas.
- Respuesta de los actuadores (abrir/cerrar) y sensores.

Criterio de Aceptación: Cumplimiento total de las tareas obligatorias definidas en la guía del proyecto sin intervención manual externa.

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.

Conclusión

El proyecto validó el uso de Python y SPIKE Prime para crear un prototipo funcional de clasificación de colores. Superamos los desafíos técnicos de la transición de bloques a código mediante la depuración y una gestión organizada en Redmine. El éxito se basó en el trabajo colaborativo y pruebas incrementales, dejando como trabajo futuro la optimización de la precisión y velocidad del sistema. Se cumplieron así todos los objetivos de integración de hardware y software planteados.

Muchas Gracias

Clasificador de colores LEGO

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

Asignatura: Proyecto I
Profesor: Baris Klobertanz