<!--Proyecto I-->

Clasificador de materiales

}

Integrantes : Marcos Caldas
Saoud Ahmed
Enzo Llancabure
Alex Campillay
Bastian Hernandez
Docente : Baris Klobertanz

Contenidos

Introducción

En esta presentación mostraremos los avances del proyecto, desde el analisis de diseño realizado por el grupo para planificar un esquema correcto tanto de la interfaz como de la estructura interna del codigo, además se describiran brevemente requerimientos funcionales y no funcionales, la implementación de dichos requerimientos en el robot y su correcta arquitectura interna.

ANALISIS DEL DISEÑO

Antes de presentar los requerimientos funcionales y no funcionales, se debe definir de manera correcta al cliente al que se espera entregar las resultados finales del proyecto y también al usuario que será el que este en contacto constante con el robot clasificador.

- Cliente : El cliente para el contexto del proyecto es una empresa minera, los cuales financian del sistema robótico encargado de la clasificación de material.
- Usuario : El usuario siendo un miembro de la empresa minera, se puede referenciar como un operador de maquinaria que será el encargado de supervisar y operar el clasificador de materiales.

Requerimientos Funcionales

Algunos de los requerimientos funcionales básicos que necesita cumplir el robot correctamente a la hora de su ejecución son:

- Identificación de los bloques de Lego.
- Correcto encolado de los bloques de Lego.
- Clasificación y almacenamiento de los bloques de Lego en compartimientos.
- Disponibilidad de Automatización del proceso.

REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales, se pueden definir como atributos de calidad del software, los cuales son esenciales para que los Stakeholders puedan cualificar el proyecto, algunos de los que elegimos son :

REQUERIMIENTO NO FUNCIONAL	DESCRIPCION	METRICA DE EVALUACION
DISPONIBILIDAD	El robot deberá mantener su operación durante un mínimo del 98% de la jornada operativa completa.	Porcentaje de tiempo operativo
ROBUSTEZ	El sistema debe ser capaz de manejar errores de sensores o interrupciones momentáneas de comunicación sin detenerse.	Numero de fallos no controlados
RENDIMIENTO	El robot automatizado debe ser capaz de responder ordenes del operador en menos de un segundo.	Tiempo promedio de respuesta a comandos
USABILIDAD	La interfaz debe ser intuitiva y auto explicativa de modo que un operador pueda usarlo de forma manual.	Tiempo de capacitación requerido y cantidad de errores de operación cometidos.

Arquitectura de software

Wireframe

Implementación

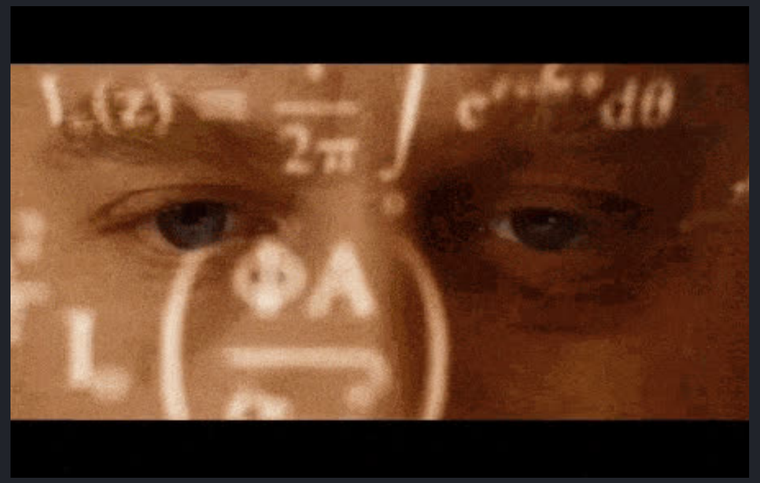
La implementación abarca temas complejos para la obtención de resultados respecto al desarrollo de la solución "Maquinaria de Clasificación de Materiales".
Esta sección se separa en dos temas fundamentales para la correcta obtención de resultados:

- Fundamentos de los movimientos
- Descripción del Sistema

FUNDAMENTOS DE LOS MOVIMIENTOS

Los fundamentos de los movimientos explican de manera teórica y experimental junto con la matemática-física aplicada el correcto funcionamiento del robot los cuales se basan en tres apartados para el correcto logro de los movimientos del robot. Los cuales son:

- **Caída del bloque lego**
- **Cálculo del Tiempo Real de posicionamiento**
- **Eficiencia Motor de empuje**



Caída del bloque Lego

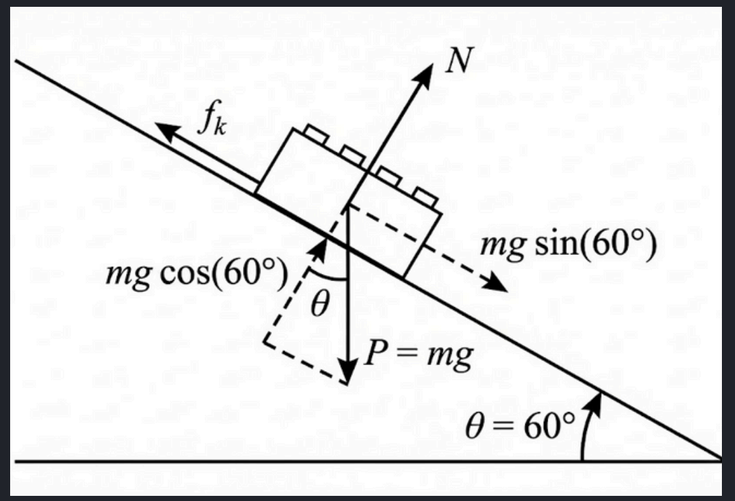
Se utiliza con el fin de poder determinar la aceleración ideal requerida para el correcto funcionamiento, tomando en cuenta los siguientes datos:

Masa del bloque (kg): 2.32 x 10-3 kg referenciada como P**

Coeficiente de roce cinético (uk): 0.654 (ABS)

Gravedad: 9.8 m/s2**

Ángulo de inclinación (theta): 60°



Caída de Bloque Lego

Los datos anteriores mencionados sirven para el calculo de cada fuerza identificada las cuales son Fuerza Normal, Fuerza de Roce, Fuerza generada por la gravedad (PX) estas fuerzas ayudan para la aplicación de la segunda ley de Newton para obtener la aceleración requerida.

	FORMULAS	RESULTADO (N)
FUERZA ROCE	**uk * N**	**7.44 × 10-3**
PX	P * sin(60)	1.969 × 10-2
FUERZA NORMAL	P * cos(60)	1.137 × 10-2

	FORMULA	RESULTADO
ACELERACIÓN	(PX - FUERZA ROCE) / P	5.28 m/s**2

Cálculo del Tiempo Real de posicionamiento

El robot opera bajo una dinámica de movimiento en un rango simétrico de -35° a +35°. Este ángulo específico se determinó tras validar distintos grados: un ángulo mayor resultaba ineficiente en tiempo de recorrido, mientras que un ángulo menor no otorgaba la distancia física suficiente para un correcto ordenamiento.
Cálculo Operativo: Se considera la distancia máxima posible (extremo a extremo = 70°) y la velocidad límite del Hub (300°/s).

	FORMULA	RESULTADO
TIEMPO	DISTANCIA/VELOCIDAD	0.23 s

Eficiencia Motor de empuje

El motor de empuje se encarga de posicionar los bloques en su sección de color. Para maximizar la eficiencia, se configuró un recorrido de 180° (ida y retorno) en lugar de un giro completo de 360°. Esta reducción de trayectoria optimiza el tiempo de ciclo, generando un movimiento rápido y preciso.

Cálculos de Validación (Tiempo y Fuerza): Se realizan dos cálculos fundamentales para validar que el sistema posee la velocidad necesaria para el impacto y la fuerza suficiente para mover la carga.
Datos: Velocidad (omega) = 300°/s | Radio de la palanca de empuje(r) = 4 cm | Torque = 18 N*cm

Eficiencia Motor de empuje

Los datos extraídos son necesarios para asegurar que el motor tiene la potencia suficiente para mover el bloque. Además, la velocidad añadida permite generar el impacto necesario, cumpliendo así con el objetivo de eficiencia respecto al tiempo.

	FORMULAS	RESULTADO
TIEMPO DE CICLO	DESPLAZAMIENTO/VELOCIDAD	0.6 segundos
FUERZA DE EMPUJE	TORQUE/RADIO	4.5 Newtons

Implementación Cliente

Tenemos 2 tipos de Cliente: El cliente físico y el cliente lógico

Como cliente físico tenemos el celular y el computador

Como cliente lógico tenemos el código de la interfaz.

Cliente Lógico

```
155                    // -------- JOYSTICK --------
156                    SizedBox(
157                      width: 220,
158                      height: 220,
159                      child: Joystick(
160                        mode: JoystickMode.all,
161                        listener: (details) {
162                          // Usa WASD para movimientos
163                          if (details.y < -0.5) sendKey("w");
164                          else if (details.y > 0.5) sendKey("s");
165
166                          if (details.x < -0.5) sendKey("a");
167                          else if (details.x > 0.5) sendKey("d");
168                        },
169                      ),
170                    ),
171                  ],
172                ),
```

Por ejemplo acá tenemos una parte de nuestro código, el cual viene del código de Flutter.

Lo más importante en este código viene a ser la función sendKey, el cual se llama para que luego la función mande un mensaje.

Implementación Servidor

Tenemos 2 tipos de Servidor: El Servidor físico y el Servidor lógico

Como Servidor físico tenemos el LEGO Technic Large Hub.

Como Servidor lógico tenemos el código realizado en Python que usa métodos de pybricks.

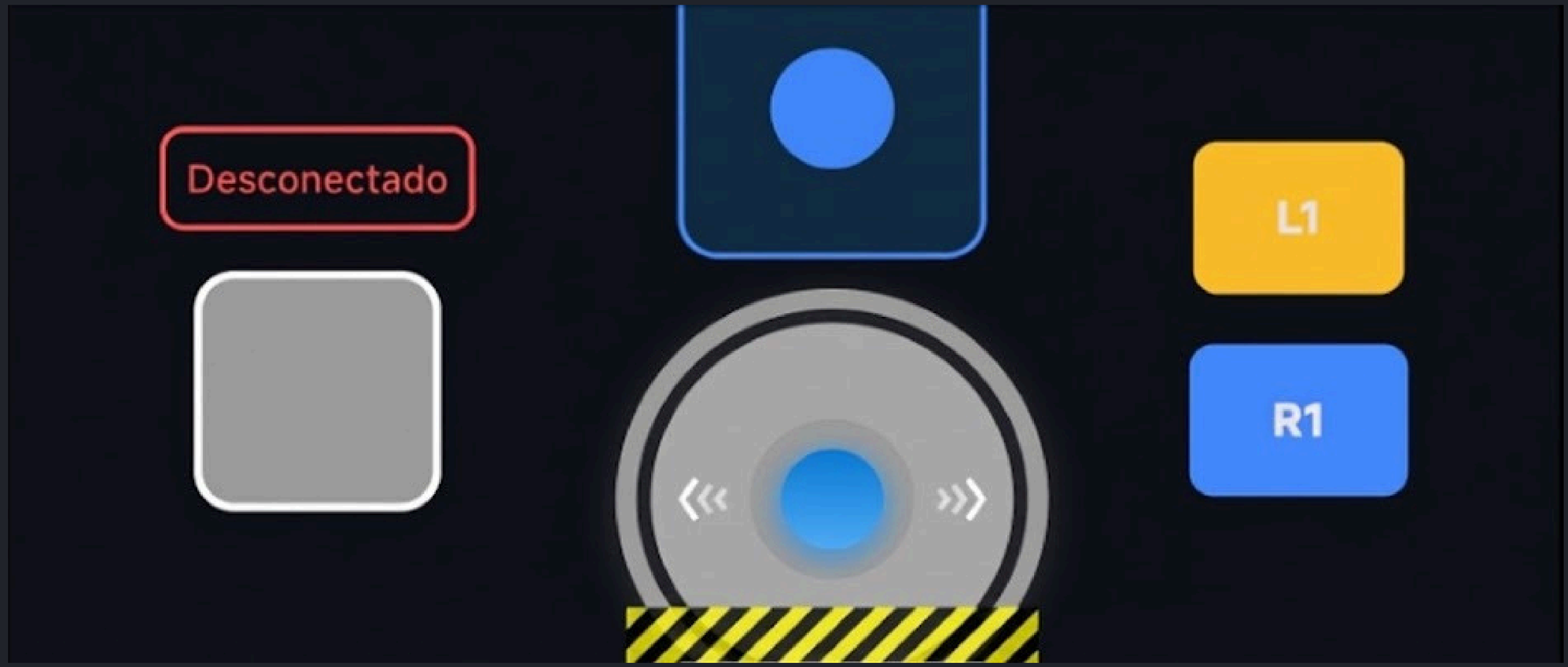
Servidor Lógico

```
15
16      if color_detectado == Color.RED or color_detectado == Color.YELLOW:
17
18
19          motor_posicion.run_target(1000, -35)
20          if color_detectado == Color.RED:
21
22              motor_empuje.run_angle(1000, -180)
23
24          elif color_detectado == Color.YELLOW:
25              motor_empuje.run_angle(1000, 180)
26
27
28      elif color_detectado == Color.GREEN or color_detectado == Color.BLUE:
29
30          motor_posicion.run_target(1000, 35)
31          if color_detectado == Color.GREEN:
32              motor_empuje.run_angle(1000, 180)
33
34          elif color_detectado == Color.BLUE:
35              motor_empuje.run_angle(1000, -180)
36
```

Acerca de nuestro servidor lógico, tenemos de ejemplo una parte de nuestro código Control_Automatico_Archivo_Base.py

Este código usa 2 métodos los cuales se encargan de mover los motores de nuestro hardware

Interfaz grafica de usuario(GUI)



Resultados

Los requerimientos funcionales y no funcionales logrados con su descripción contextual son los siguientes

REQUERIMIENTOS FUNCIONAL	DEFINICION	DESCRIPCIÓN
RF1	Identificación de Materiales	LOGRADO: El sensor reconoce el color del bloque y decide la separación basándose en ello.
RF2	Flujo Continuo (FIFO)	LOGRADO: El sistema reconoce y ordena bloques de manera sucesiva, manteniendo un control constante.
RF3	Clasificación en Celdas	LOGRADO: Se sincroniza el posicionamiento (giro) con el empuje para colocar el bloque en su destino exacto.
RF4	Disponibilidad de Automatización del proceso.	EN REVISIÓN: El bucle automático opera sin fallos. El modo manual es funcional a nivel lógico, pero presenta fallos de conexión con la interfaz externa actualmente.

Resultados

REQUERIMIENTOS NO FUNCIONALES	**EVALUACIÓN**	CUMPLIMIENTO
DISPONIBILIDAD	Tiempo de recuperación ante fallas (< 10 min).	La recuperación es casi inmediata mediante reinicio del Hub/Pybricks tras pérdida de conexión.
ROBUSTEZ	Número de fallos no controlados (Manejo de errores).	Baja tolerancia a obstrucciones físicas (requiere ayuda manual).
RENDIMIENTO	Tiempo de respuesta a comandos (< 1 seg).	el sistema ejecuta las órdenes de clasificación en tiempos inferiores a 1 segundo.
USABILIDAD	Tiempo de capacitación y cantidad de errores.	5-10 minutos (Gracias a interfaz tipo PS4).

Problemas Encontrados

- Dificultades en la implementación de la interfaz gráfica y la comunicación entre la aplicación y el hub.
- La aplicación no detectaba ni buscaba el hub del robot mediante Bluetooth debido a permisos del sistema operativo Android.
- Se solucionó solicitando permisos de ubicación y Bluetooth al ejecutar la aplicación.
- Problemas con el framework Flutter, ya que solo estaba instalado en un computador del grupo.
- Esto retrasaba las pruebas y correcciones del proyecto.
- Se solucionó instalando Flutter en todos los equipos del grupo.

Conclusión