**Proyecto final**

**Mecatrónica**

**Desarrollo de un robot balancing**

**con navegación SLAM**

**Propuesta final**

* **Docentes:**

Ing. Cristian Leandro Lukaszewicz

Ing. Ezequiel Blanca

Ing. Juan Ignacio Szombach

* **Alumno:**

Garcés Patricio (DNI: 39.919.131)

Tabla de contenido

[Motivación 3](#_Toc207662406)

[Propuesta 3](#_Toc207662407)

[Alcance 4](#_Toc207662408)

[Docentes a consultar 4](#_Toc207662409)

[Especificación de requerimientos 5](#_Toc207662410)

[Diseño funcional 6](#_Toc207662411)

[Lista de materiales y costos 9](#_Toc207662412)

[Pinout ESP32 10](#_Toc207662413)

# Motivación

El desarrollo de un robot de balanceo con sistema de navegación basado en visión computacional surge en primer lugar de la búsqueda de conocimiento en áreas como sistemas de control, visión artificial y programación de embebidos a bajo nivel, en segundo lugar, de la necesidad de crear dispositivos robóticos capaces de interactuar de manera eficiente en entornos dinámicos y variables.

Este proyecto busca abordar este desafío mediante la integración de diferentes disciplinas de la ingeniería, promoviendo la innovación y el progreso en el campo de la mecatrónica.

# Propuesta

El proyecto consistirá en diseñar y construir un robot de balanceo equipado con un sistema de navegación basado en visión computacional. El sistema de visión computacional estará compuesto por una cámara, cuyo modelo y tipo específico será determinado durante el desarrollo del proyecto, que capturará imágenes del entorno del robot.

Estas imágenes serán procesadas para generar un plano 2D del área circundante, permitiendo al robot planificar rutas y evitar obstáculos durante su desplazamiento. El proyecto se llevará a cabo en tres etapas principales: diseño y construcción del robot de balanceo, implementación del sistema de navegación basado en visión computacional y pruebas experimentales para validar el funcionamiento del sistema en entornos simulados y reales

# Alcance

El presente proyecto tendrá como alcance el cumplimiento de los siguientes objetivos principales:

* Diseñar y construir un prototipo funcional de robot de balanceo capaz de mantener su estabilidad dinámica.
* Integrar un sistema de navegación basado en visión artificial, empleando una cámara para la captura de imágenes del entorno.
* Implementar un sistema de control manual del robot a través de una aplicación móvil.

Adicionalmente, como funcionalidades deseables se plantea:

* Desarrollar un algoritmo de procesamiento de imágenes que permita generar un plano 2D del área circundante.
* Realizar pruebas experimentales en entornos reales, con el fin de validar el desempeño del sistema, particularmente en la detección de obstáculos y en su capacidad de desplazamiento autónomo.

# Docentes a consultar

A fin de poder evacuar dudas y encontrar soluciones a los problemas que se puedan encontrar en el desarrollo del presente trabajo, se decide seleccionar a los siguientes docentes:

* Matías Joel Hirak (Fundamentos de Robótica)
* Alejando Simoncelli (Procesamiento de señales)

# Especificación de requerimientos

**Balance Dinámico:**

El robot debe ser capaz de mantener su equilibrio dinámico en tiempo real, utilizando un sistema de control PID basado en sensores de inclinación y giróscopo MPU6050 o MPU9250.

**Procesamiento de Imágenes:**

El robot utilizará una Raspberry Pi para procesar imágenes capturadas por una cámara de PS5.

**Control Remoto a través de una app:**

Se desarrollará una aplicación básica para Android que permitirá al usuario controlar de manera remota el robot.

La app deberá ser capaz de enviar comandos básicos al robot, como iniciar/detener el movimiento y ajustar la posición.

**Generación de Plano 2D(Deseable):**

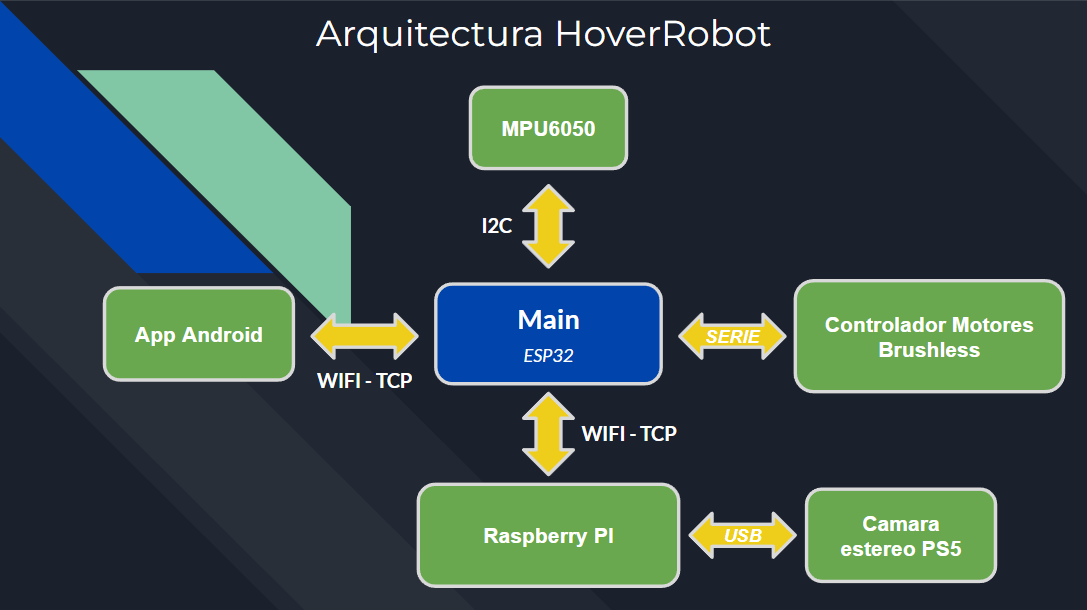
Las imágenes capturadas por la cámara serán procesadas en la Raspberry Pi para generar un mapa 2D del entorno circundante en tiempo real, similar a un sistema LiDAR.

**Prevención de Colisiones(Deseable):**

El mapa 2D generado será utilizado para identificar objetos en el entorno del robot y evitar colisiones durante su movimiento.

# Diseño funcional

**Arquitectura general**



**Unidad de Control Principal (Main – esp32):**

Es el núcleo del sistema que gestiona la estabilidad del robot. Procesa las señales del acelerómetro y el giróscopo (como los integrados en los módulos MPU6050 o MPU9250) para realizar servir de entrada a los distintos PID obteniendo los ajustes necesarios en los motores y mantener el equilibrio.

El ESP32 también se comunica vía UART con una placa de control de hoverboard reprogramada, que maneja directamente los motores del robot.

**Sistema de Visión Computacional (Raspberry Pi + Cámara de PS5):**

Captura y procesa imágenes del entorno en tiempo real. La Raspberry Pi se encarga de convertir estas imágenes en un mapa de profundidad permitiendo al robot identificar cuando hay un objeto o superficie que no le permita avanzar.

**Sistema de Actuación (Controlador y motores de hoverboard):**

Los motores son controlados por una placa de hoverboard reprogramada para recibir instrucciones desde el ESP32.

Esta placa será grabada con un firmware custom para manejar los motores de manera eficiente (Control FOC), adaptando la velocidad y dirección de cada motor según los datos recibidos.

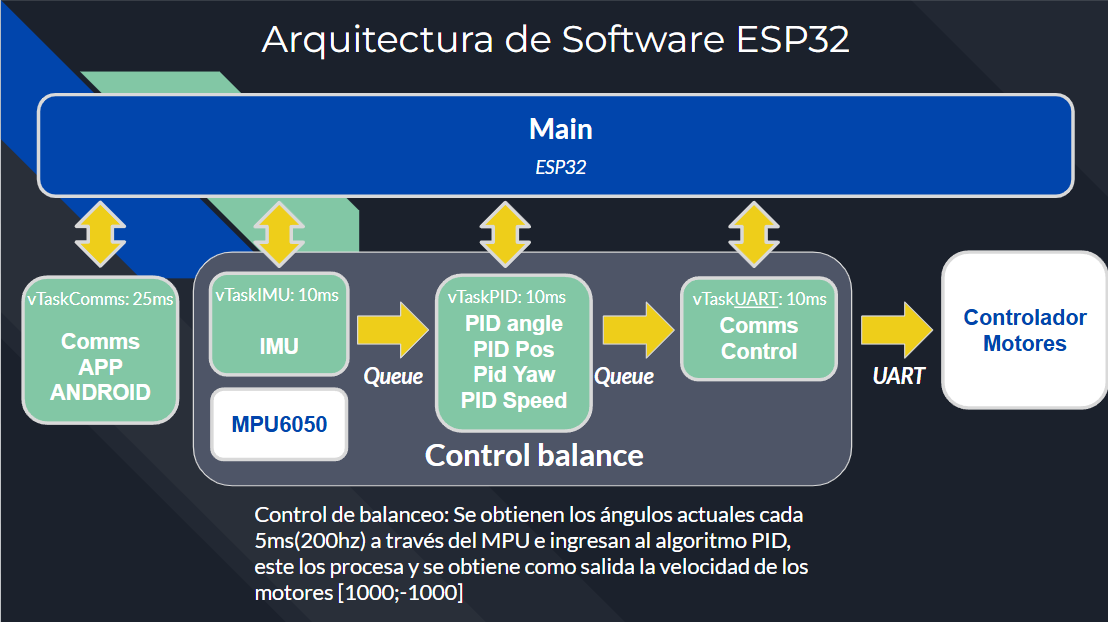
**Interfaz de Comunicación (WIFI):**

Facilita la comunicación entre el robot, la aplicación Android y la raspberry pi, permitiendo el control remoto y la visualización de datos en tiempo real.

**Aplicación Android:**

Permite al usuario controlar el robot de manera remota, enviar comandos básicos y recibir en tiempo real la representación del entorno captada por el robot.

**Arquitectura de software ESP32**



El firmware del ESP32, basado en FreeRTOS, organiza el funcionamiento del robot a través de varias tareas concurrentes que se ejecutan en tiempo real. El sistema se encarga de leer constantemente los datos del acelerómetro y giroscopio para calcular la estabilidad del robot, ajustando los motores mediante la comunicación UART con la placa de control de hoverboard reprogramada. Además, el firmware gestiona la recepción de datos del sistema de visión computacional, procesando la información recibida de la raspberry pi para integrarla en la lógica del robot. Simultáneamente, el ESP32 maneja la comunicación con la aplicación Android a través de Wi-Fi, permitiendo el control remoto y la visualización en tiempo real.

# Lista de materiales y costos

|  |  |  |  |
| --- | --- | --- | --- |
| Materiales | Cantidad | Costo Unitario (Usd) | Costo (Usd) |
| Esp32 | 1 | Usd 10 | Usd 10 |
| Step Down HV | 1 | Usd 2 | Usd 2 |
| Hoverboard en desuso | 1 | Usd 30 | Usd 30 |
| Placa perforada | 1 | Usd 3.5 | Usd 3.5 |
| Raspberry pi | 1 | Usd 100 | Usd 100 |
| Cámara PS5 | 1 | Usd 30 | Usd 30 |
| Base fibrofacil | 2 | Usd 5 | Usd 10 |
| Perfiles aluminio 40x40x200 | 2 | Usd 3 | Usd 6 |
| Rollo filamento 3d 1kg PLA | 1 | Usd 12 | Usd 12 |
| Tornillería | - | Usd 10 | Usd 10 |
| Módulos CAN | 2 | Usd 4.25 | Usd 8.5 |
| Total |  |  | Usd 222 |

# Pinout ESP32

