

Robot Balanceador de 2 Ruedas

Sistemas de Tiempo Real

Melina Caciani Toniolo

Bruno Zanetti

Franco Niderhaus

Mateo Larsen

Gabriel Ollier

Joaquin Chanquia





Objetivo

Desarrollar un sistema de balanceo de un vehículo de dos ruedas sobre una plataforma ESP32, logrando mantener el equilibrio de forma autónoma.

Implementar un sistema de control robusto que permita corregir las desviaciones del vehículo y mantenerlo en posición vertical, a partir de un algoritmo de PID.

Posibles funcionalidades del proyecto



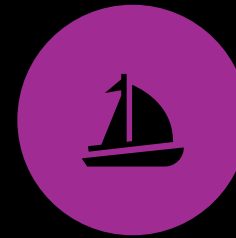
TRANSPORTAR PEQUEÑOS
OBJETOS COMO UN MATE,
LIBROS O DISPOSITIVOS
ELECTRÓNICOS ENTRE
HABITACIONES DE MANERA
AUTÓNOMA.



SERVICIO EN OFICINAS:
MOVER DOCUMENTOS,
PAPELES O MATERIAL DE
OFICINA ENTRE
ESCRITORIOS O ÁREAS DE
TRABAJO.



ASISTENTE EN TALLERES O
GARAJES: LLEVAR
HERRAMIENTAS
O MATERIALES



NAVEGACIÓN AUTÓNOMA:
EL ROBOT PODRÍA SEGUIR
UNA RUTA PREDEFINIDA



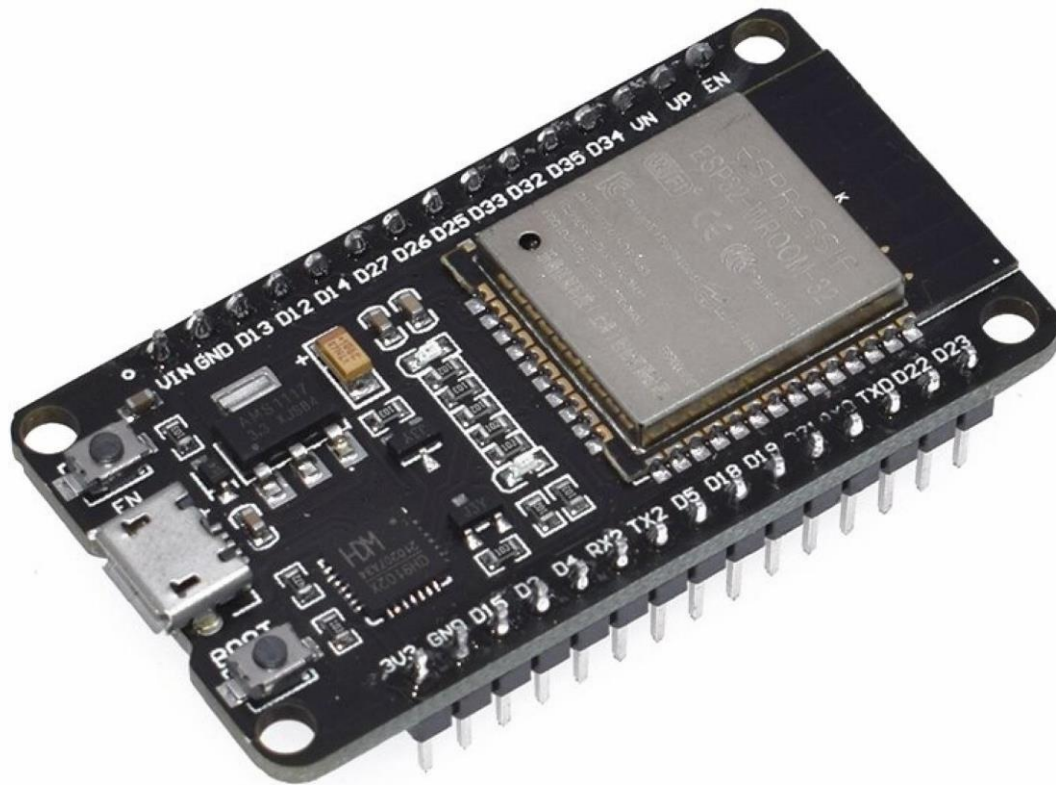
ROBOT GUÍA EN MUSEOS
O EXPOSICIONES.

Materiales a utilizar

Placa ESP32

Características principales del microcontrolador:

- Conectividad Wi-Fi y Bluetooth
- Posee un procesador dual-core
 - 3,3V – 5V
 - 180 mA – 260 mA



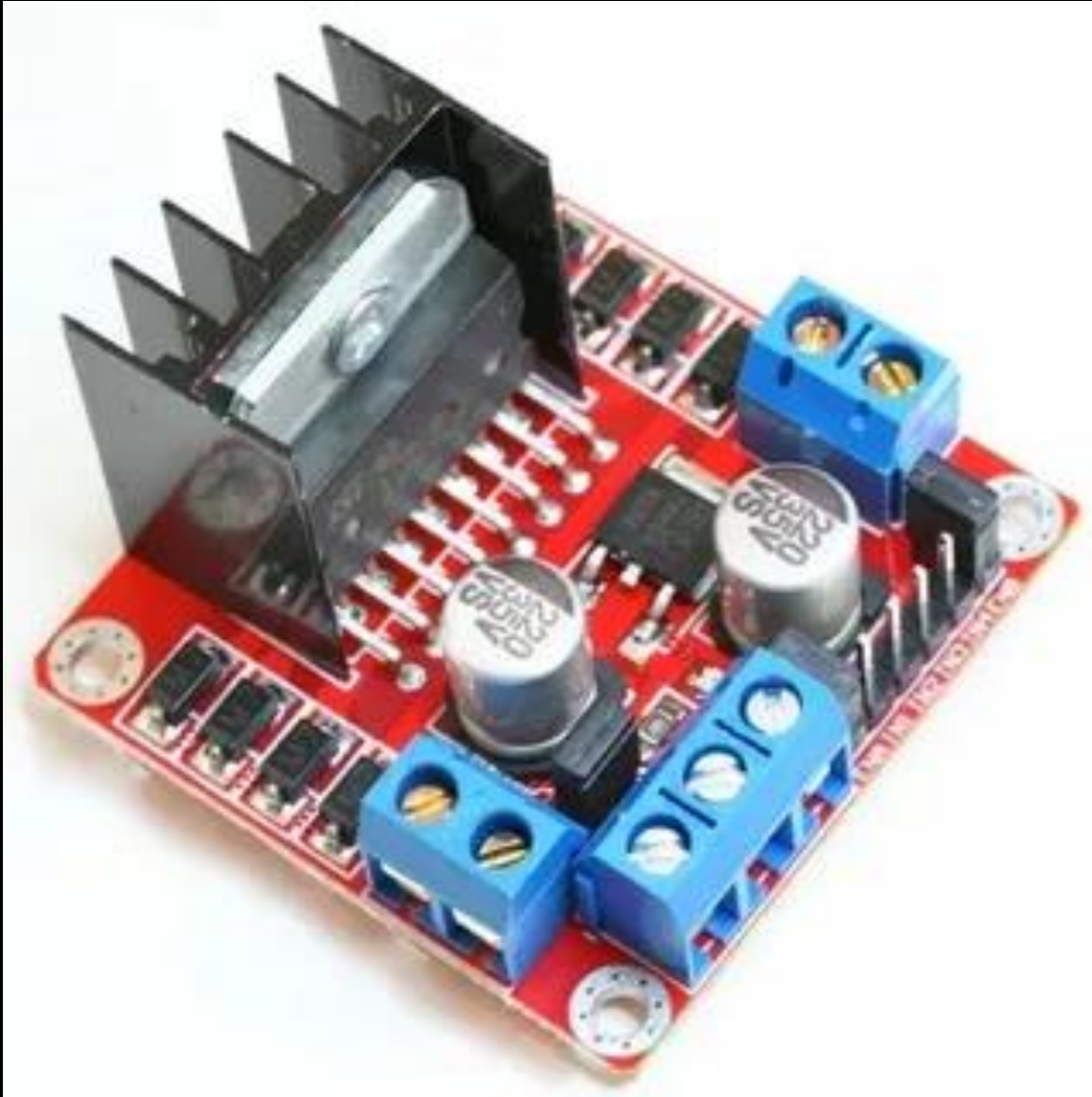
Motores

- Se utilizarán 2 motores de corriente continua conectados a un encoder cada uno
- Este motor se utilizará para mover una rueda de plástico estándar
- 12V / 2A

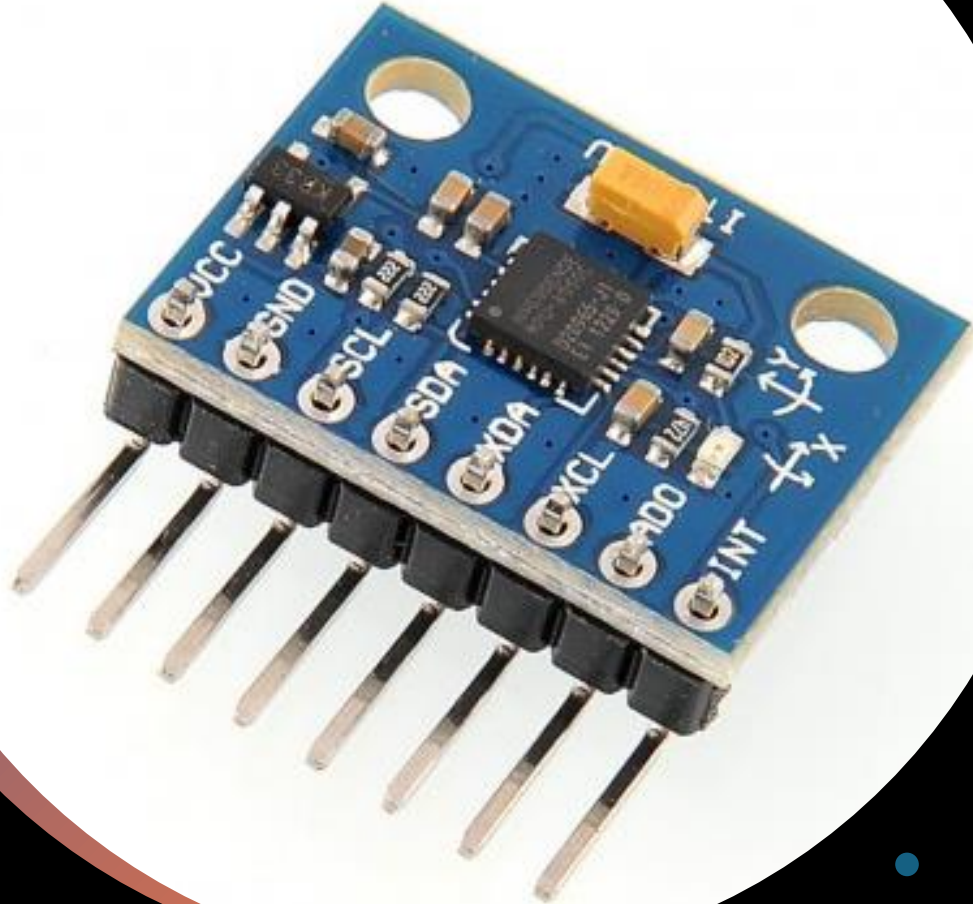


Controlador Punte H L298N

- Se utilizará un Punte H para controlar los motores DC.
- 4,5V – 7V
- 13mA – 70mA

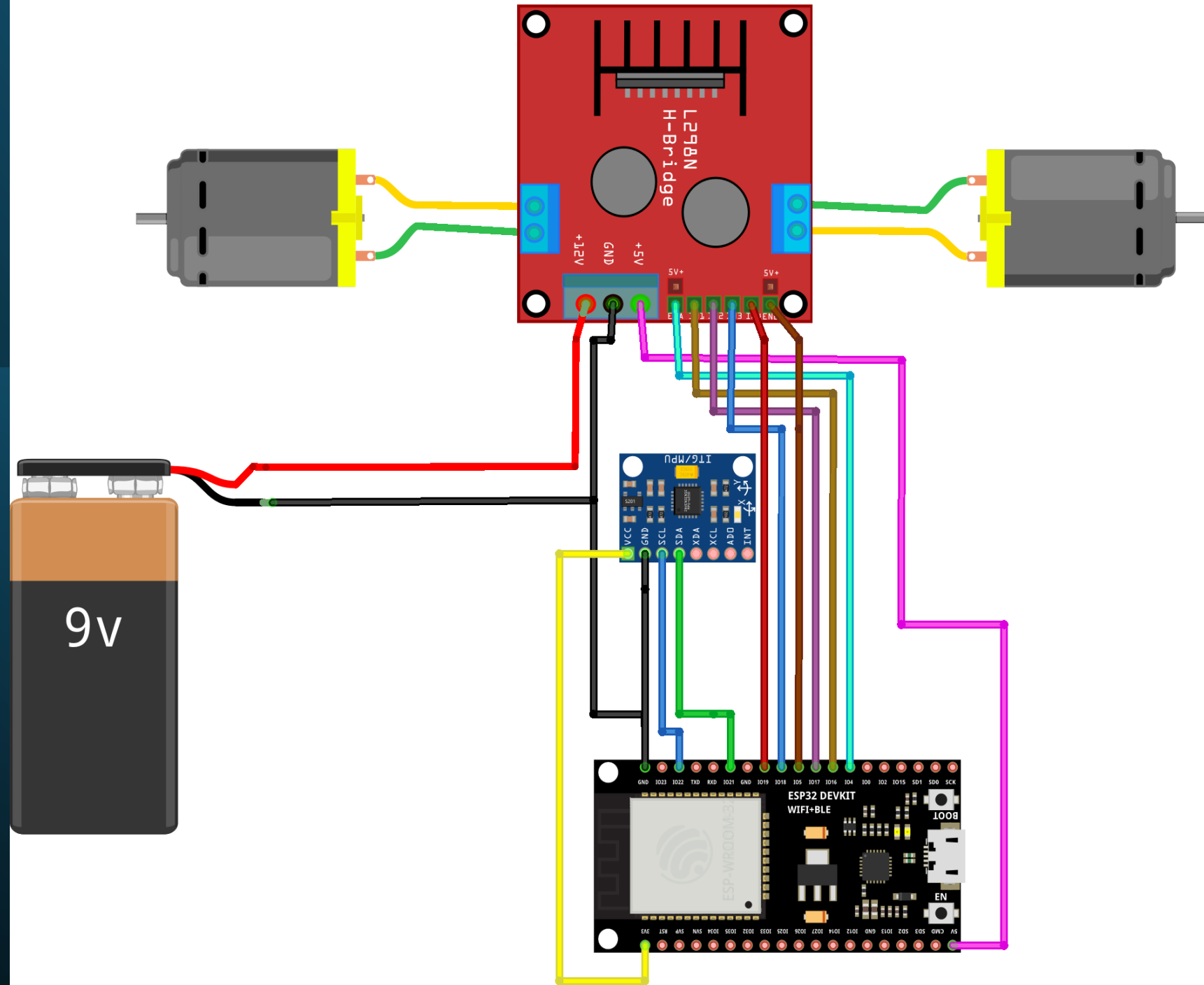


Acelerómetro y Giroscopio MPU6050

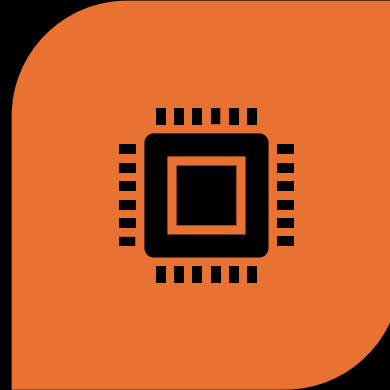


- Se utilizará este sensor para determinar la inclinación del robot y enviar la información de la misma al ESP32
- 3,3 V
- 3,9 mA

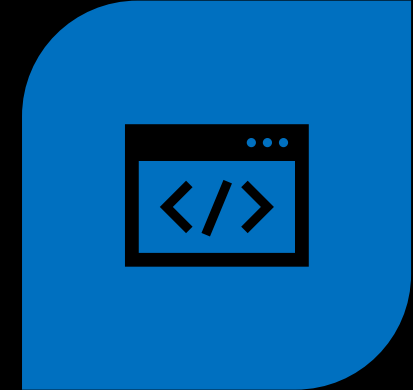
Esquema gráfico del proyecto



Protocolos de comunicación



- I2C: SE UTILIZARÁ PARA ENVIAR DATOS AL MICROCONTROLADOR DESDE EL SENSOR MPU ACCELERÓMETRO Y GIROSCOPIO.



- HTTP: SERÁ UTILIZADO PARA LA COMUNICACIÓN ENTRE LA APLICACIÓN WEB Y EL ESP32, YA QUE DESDE LA APP SE ENVIARÁN COMANDOS AL ROBOT.

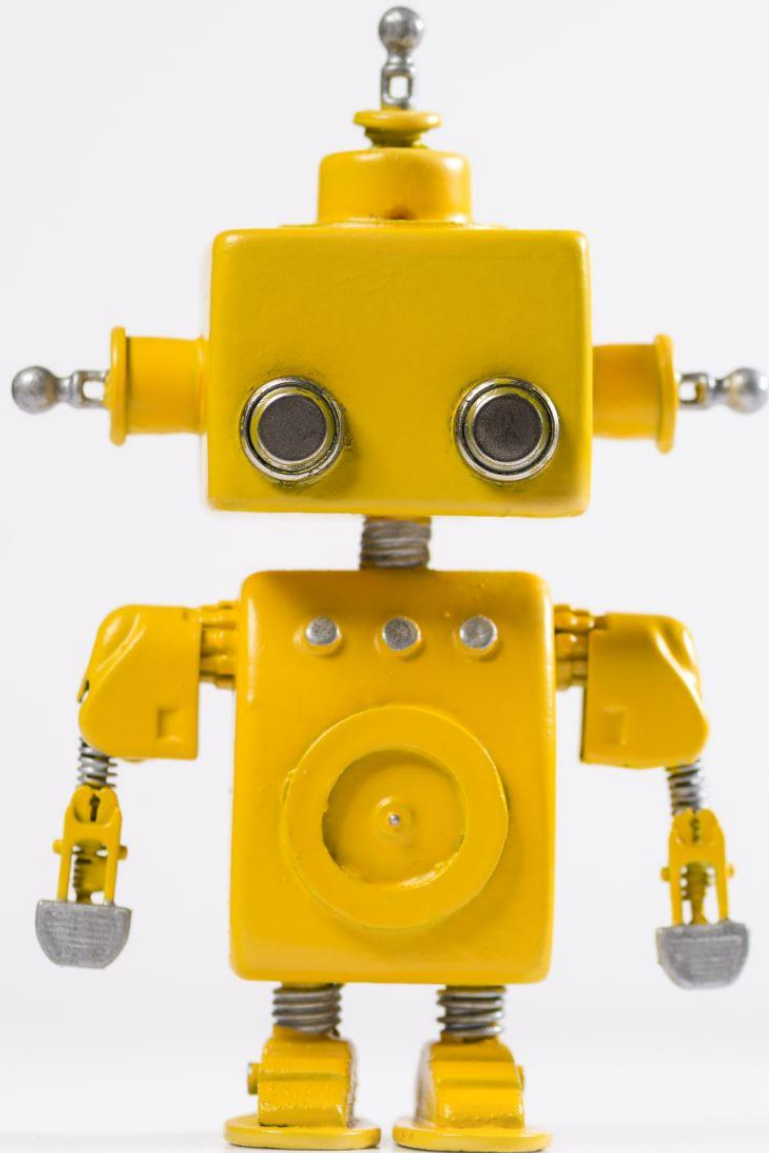
Entorno de desarrollo

- Utilizaremos el lenguaje de programación C como lenguaje de programación principal en el proyecto. Pero también nos apoyaremos con el lenguaje de Arduino para poder utilizar librerías públicas.
- Como plataformas utilizaremos los entornos Visual Studio Code y el IDE de Arduino.

Desarrollo de firmware

- Plataforma: [platformio](#)
- El firmware permite realizar las funciones del microcontrolador como controlar los motores, procesar las lecturas del sensor y ejecutar el algoritmo PID en tiempo real para mantener el equilibrio del robot, garantizando que las instrucciones se ejecuten con precisión y en el tiempo adecuado.





Simulación

- Plataforma: [Fritzing](#)
- La simulación replicará el comportamiento del robot en tiempo real, demostrando cómo responde a diferentes situaciones. Todo lo que el robot real ejecuta, desde el control de equilibrio hasta los ajustes de los motores mediante el algoritmo PID, se reflejará de manera idéntica en la simulación. Esto garantiza que los requerimientos y la lógica de control se validen en un entorno virtual antes de su implementación física.

Desarrollo de Software

- Plataforma: Visual Studio
- El desarrollo de software en este proyecto será el de la aplicación capaz de manejar al robot, con la cual podremos mover al mismo en la dirección que se deseé.



Tareas Iniciales



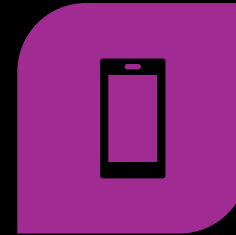
**ELABORACIÓN DE
INFORMES INICIALES**



**ANÁLISIS ELÉCTRICO
(ESPECIFICACIONES DE LOS
COMPONENTES.)**



**COMENZAR A EL
DESARROLLO DEL
FIRMWARE.**



**DESARROLLAR LA
APLICACIÓN DE CONTROL
DEL ROBOT**



**CONEXIONADO DE
COMPONENTES
HW. (REQUIERE HARDWARE)**