

## Documentación código ESP32.

El código presentado en la carpeta **ProgramaciónESP32** consiste en el centro del sistema de tracción del proyecto **LimpiaVidrios**. Este sistema permite controlar los motores que mueven el robot a través de un controlador de videojuegos, integrando tanto la lógica de control como la interfaz de usuario.

### Estructura del Proyecto

El proyecto se compone de los siguientes archivos principales:

- **LimpiaVidrios.ino**: Archivo principal que contiene la lógica de control y la integración del joystick.
- **motores.h**: Archivo de cabecera que declara la clase Motores.
- **motores.cpp**: Archivo de implementación que define el funcionamiento de las funciones de la clase Motores.

---

## Archivo LimpiaVidrios.ino

El archivo **LimpiaVidrios.ino** es el núcleo del proyecto. Se encarga de manejar la interacción con el controlador de videojuegos y controlar el movimiento del robot, además de la interacción por comandos del teclado.

### Función dumpGamepad

Esta función imprime en el monitor serial el estado del controlador, incluyendo el índice, el estado del D-pad, los botones presionados y la posición de los ejes del joystick. Proporciona una visión clara de la entrada del usuario y facilita el diagnóstico y la depuración.

### Función processGamepad

Esta función maneja la entrada del controlador y ejecuta el movimiento correspondiente de los motores en función del estado del D-pad y los ejes del joystick. Si se detecta que no hay movimiento en ninguna de las entradas, los motores se bloquean para evitar que el robot se desplace involuntariamente. Esto asegura un control seguro y preciso del robot.

### Función loop

La función loop es la parte del código que se ejecuta de manera continua en el microcontrolador. Se encarga de la lectura serial de la interacción del usuario a través del teclado del computador, y permite que se haga la lectura de los datos proporcionados por el controlador de videojuegos. Esto permite una comunicación constante entre el usuario y el robot, asegurando que las instrucciones se interpreten y ejecuten en tiempo real.

---

## Archivo motores.h

El archivo **motores.h** declara la clase **Motores**, que encapsula las variables y funciones necesarias para controlar los motores del robot. La clase contiene métodos para configurar los pines de control de los motores, bloquear los motores y mover el robot en diferentes direcciones (adelante, atrás, izquierda y derecha). Esta estructura modular facilita la gestión y reutilización del código.

---

## Archivo motores.cpp

El archivo **motores.cpp** contiene la implementación de las funciones de la clase **Motores**.

### Constructor

El constructor de la clase inicializa los pines de control de los motores y un pin adicional utilizado para el bloqueo de motores. Esto asegura que todos los pines necesarios estén configurados correctamente antes de su uso.

### Método setup()

Este método configura los pines de los motores como salidas y llama al método de bloqueo de motores al inicio del programa. Esto garantiza que el robot no se mueva accidentalmente al encenderse.

### Métodos de Movimiento

La clase Motores incluye métodos para mover el robot en diferentes direcciones. Cada uno de estos métodos activa o desactiva los pines

de control correspondientes para lograr el movimiento deseado y también envía un mensaje al monitor serial para informar el estado del movimiento. Esto permite un seguimiento en tiempo real de las acciones del robot y facilita el control durante la operación.