PROYECTO HALLO CAM EPS32







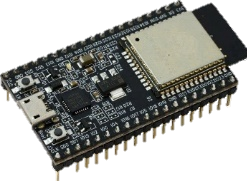
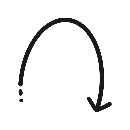


**Elaborado por:**

Eliasib Cadena Méndez

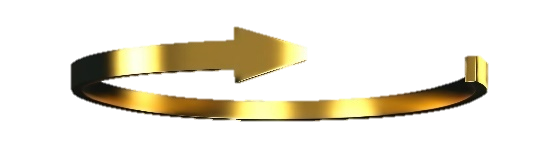
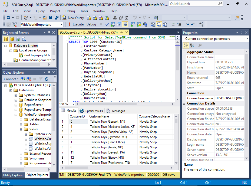
Ingeniero De Sistemas

Bucaramanga/Santander

Logotipo

Descripción generada automáticamente

Los servos motores realizan los cambios en la sección que les corresponde moviendo cada componente



Esp32 lee los datos en las tablas verifica el estado de cada servomotor y realiza los cambios en los servomotores

Acceder desde el dispositivo a la web o APP con los controles de la cámara como mover y zoom

Diseñado por : Ing. Eliasib Cadena M.

Web – Servidor Sqlserver o MySQL

Acceso al servidor realizando cambios en la tabla con los datos de cada servomotor

Giro horizontal de la cámara

Diagrama

Descripción generada automáticamente

El diagrama de flujo describe el proceso que inicia en un dispositivo ESP32, el cual está programado para conectarse a un servidor. Este dispositivo gestiona la posición en grados de un servomotor y controla los LEDs de estado. Al conectarse a la red Wi-Fi, el ESP32 accede a un archivo llamado esp32.php ubicado en el localhost.

Este archivo se encarga de consultar la base de datos MySQL para determinar el estado actual de cada servomotor. A través de esta consulta, el archivo devuelve información sobre el estado en el que se encuentran o deberían estar los componentes. Además, el archivo puede actualizar la base de datos MySQL para cambiar el estado de los servomotores según sea necesario.

Una vez que MySQL proporciona esta información al archivo PHP, este se comunica de nuevo con el ESP32 a través de Wi-Fi. El ESP32, al recibir la información sobre el estado de los servomotores, verifica y ejecuta las acciones necesarias para asegurar que los componentes funcionen según lo esperado.

Diagrama

Descripción generada automáticamente

# **Software Necesario y Configuración**

Para este proyecto, he optado por utilizar **XAMPP** en su versión 8.2.12 de **Apache Friends**. Este software es una herramienta sencilla y eficaz para gestionar servidores web y bases de datos, lo que lo convierte en una excelente opción para el desarrollo de aplicaciones web. Aunque existen otras alternativas como **WAMP**, ambas soluciones cumplen funciones similares; sin embargo, **XAMPP** destaca por su simplicidad de configuración, especialmente cuando se trabaja con dispositivos móviles o dispositivos IoT.

Una de las principales razones por las que he elegido **XAMPP** es su facilidad al momento de configurar entornos de desarrollo en los que se necesite que dispositivos externos, como teléfonos móviles o sistemas IoT, puedan interactuar con el servidor web sin complicaciones. Al probar ambos entornos, **WAMP** requiere una configuración más compleja y prolongada para lograr lo mismo, lo que hace que **XAMPP** sea una opción más eficiente para este tipo de proyectos.

La idea a largo plazo es que este proyecto no se limite a un entorno local. En el futuro, se planea que el **ESP32** pueda conectarse a un servidor en la nube con **MySQL** e interactuar con los documentos alojados en dicho servidor. De esta manera, la página web reflejará los cambios realizados en la base de datos, y una aplicación móvil también podrá visualizar y controlar los cambios, permitiendo una interacción total con el **ESP32** desde cualquier lugar. Inicialmente, el proyecto comenzará con el encendido y apagado de un **LED**, pero más adelante se cambiará este por un **servomotor**, y el **LED** solo servirá como un indicador de estados.

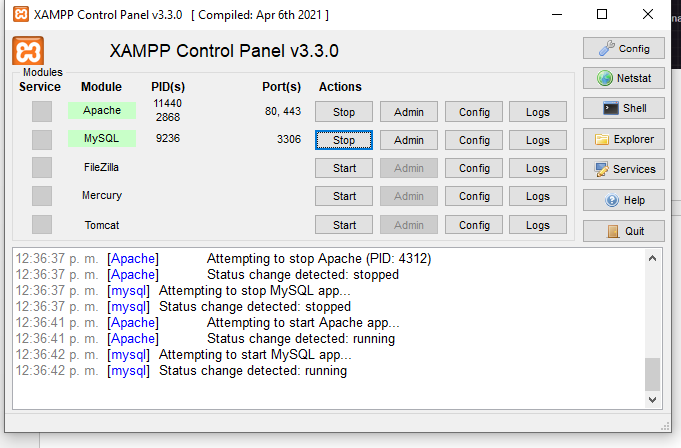


Ilustración 1 Pantallazo del Panel de Control de XAMPP v3.3.0. En esta imagen se muestra el estado de los servicios Apache y MySQL en ejecución. Esta configuración es utilizada en el proyecto para manejar el servidor web local y la base de datos MySQL, facilitando el desarrollo y pruebas de las funcionalidades del proyecto de cámaras.

## Configuración de Xamppp

Se realizaron modificaciones en el archivo config.inc.php con el propósito de realizar pruebas en el usuario de inicio de phpMyAdmin. Específicamente, se cambió la línea de código:

$cfg['Servers'][$i]['auth\_type'] = 'config';

por la siguiente línea:

$cfg['Servers'][$i]['auth\_type'] = 'cookie';

Este cambio tiene como objetivo modificar el tipo de autenticación de config a cookie, lo que permite que el usuario ingrese sus credenciales en cada inicio de sesión, en lugar de tenerlas pre configuradas.

El resultado final de la sección modificada del archivo config.inc.php es el siguiente:

/\* Authentication type and info \*/

// Comentar la línea original de autenticación

// $cfg['Servers'][$i]['auth\_type'] = 'config';

// $cfg['Servers'][$i]['user'] = 'root';

// $cfg['Servers'][$i]['password'] = '';

// $cfg['Servers'][$i]['AllowNoPassword'] = true;

// Añadir la nueva configuración de autenticación

$cfg['Servers'][$i]['auth\_type'] = 'cookie'; // Nueva autenticación por cookie

// Las demás configuraciones permanecen igual

$cfg['Servers'][$i]['extension'] = 'mysqli';

$cfg['Lang'] = '';

Este cambio asegura que phpMyAdmin solicite un usuario y contraseña en cada sesión, mejorando la seguridad del acceso al gestor de bases de datos.

En la siguiente ilustración podemos ver como se puede apreciar el control desde la web alojada en el móvil:

