

SMARTIFY

Domotización mediante Arduino

Alejandro Fernández Rico

Desarrollo de Aplicaciones Multiplataforma

Víctor Colomo Gómez

Indice Paginado

1.RESUMEN.....	4
2.JUSTIFICACIÓN DEL PROYECTO.....	5
3.OBJETIVOS.....	5
4.DESARROLLO	6
<i>Tecnologías Utilizadas</i>	7
METODOLOGÍA UTILIZADA	16
PLANIFICACIÓN DEL PROYECTO.....	17
DIAGRAMA BBDD.....	18
DIAGRAMA DE CASOS DE USO	18
<i>Formato 2 Columnas</i>	19
Iniciar Sesión.....	19
Registrarse	20
Encender Luz	21
Apagar Luz	22
Leer Sensor LDR.....	23
Gestión Usuarios	24
Ver Logs.....	25
DIAGRAMAS DE FLUJO	26
<i>Iniciar sesión</i>	26
<i>Registrarse</i>	27
<i>Control Luces</i>	28
<i>Luces Automáticas</i>	29
EXPLICACIÓN CÓDIGO.....	30
<i>Verificación Administrador</i>	30
<i>Conexión BBDD</i>	30
<i>Leer Tarjeta</i>	31
<i>Control Luces desde Arduino</i>	31
<i>Control Luces JavaScript</i>	32
RESULTADOS Y ANÁLISIS.....	32
CAPTURAS DEL PROYECTO	33
<i>Pantalla de Login</i>	33
<i>Pantalla Principal</i>	33
<i>Control de Luces</i>	34
<i>Perfil</i>	34

<i>Modificar Perfil.....</i>	35
<i>Administrar Usuarios</i>	35
<i>Ver Logs BBDD.....</i>	36
5.CONCLUSION	36
6.LÍNEAS FUTURAS DE INVESTIGACIÓN	37
7.BIBLIOGRAFÍA	38

1.RESUMEN

El presente Trabajo de Fin de Grado (TFG) se centra en el desarrollo de un sistema de domótica para una maqueta de casa utilizando la placa de desarrollo ESP32. La domótica, un campo en constante evolución, ofrece soluciones innovadoras para mejorar la calidad de vida en los hogares, optimizar el consumo energético y garantizar la seguridad. En este contexto, la placa ESP32 se presenta como una herramienta idónea gracias a su conectividad Wi-Fi y Bluetooth, así como su potente capacidad de procesamiento, lo que la convierte en la base perfecta para la implementación de un sistema de control inteligente.

El proyecto aborda no solo la automatización de los dispositivos y sistemas en la maqueta de casa, sino que también tiene como objetivo desarrollar una interfaz de usuario responsiva que permita controlar y monitorear la maqueta desde cualquier dispositivo con acceso a Internet, ya sea un ordenador de escritorio, una tableta o un teléfono inteligente. Esta interfaz web responsiva proporcionará una experiencia de usuario intuitiva y accesible, adaptándose de manera fluida a diferentes tamaños de pantalla y dispositivos, lo que permitirá a los usuarios interactuar con la maqueta de casa de manera eficiente y conveniente desde cualquier lugar y en cualquier momento.

2.JUSTIFICACIÓN DEL PROYECTO

Este proyecto combina investigación tecnológica, educación, eficiencia energética y seguridad del hogar mediante la integración de la domótica y la tecnología IoT. Al centrarse en la placa de desarrollo ESP32, se facilita una plataforma accesible y versátil para la creación de soluciones de automatización doméstica. Este sistema permite el control remoto y la automatización de múltiples dispositivos en una maqueta de casa. La implementación de sensores y actuadores diversos no solo mejora la interacción con el entorno doméstico, sino que también fomenta la innovación en la gestión de recursos energéticos. La investigación tecnológica en protocolos de comunicación y la optimización de la interfaz de usuario garantiza una experiencia fluida y segura para los usuarios.

Además, se pone un énfasis significativo en la eficiencia energética, desarrollando algoritmos que optimizan el consumo de energía y promueven prácticas sostenibles. La seguridad del hogar es otro pilar fundamental del proyecto, con el desarrollo de medidas avanzadas para proteger el sistema contra ciberataques y garantizar la privacidad del usuario.

3.OBJETIVOS

A. OBJETIVO GENERAL

Desarrollar un sistema de domótica basado en la placa de desarrollo ESP32 que permita el control remoto de una maqueta de una casa mediante una página web responsiva, garantizando la integración efectiva entre hardware y software, así como la seguridad y usabilidad del sistema.

B. OBJETIVOS ESPECÍFICOS

- Integrar una variedad de sensores y actuadores para permitir la interacción y automatización de diferentes aspectos del entorno doméstico en la maqueta.
- Gestionar un servidor web utilizando la placa ESP32 para alojar una interfaz de usuario que permita controlar y monitorear el sistema de domótica desde cualquier dispositivo con conexión a Internet.

- Desarrollar una interfaz de usuario web responsiva que garantice una experiencia de usuario fluida y accesible en diversos dispositivos, como ordenadores de escritorio, tabletas y teléfonos inteligentes.
- Implementar mecanismos de comunicación entre el servidor web y los dispositivos físicos en la maqueta de casa, permitiendo la interacción en tiempo real desde la interfaz web.
- Investigar y aplicar protocolos de comunicación eficientes para facilitar la interacción entre el servidor web y los dispositivos conectados, asegurando una comunicación confiable y segura.
- Realizar pruebas exhaustivas del sistema en diferentes condiciones y escenarios para garantizar su funcionalidad, confiabilidad y seguridad en entornos domésticos simulados.
- Recopilar y analizar datos sobre el rendimiento del sistema y la experiencia del usuario, identificando áreas de mejora y optimización.
- Documentar detalladamente el proceso de diseño, implementación y pruebas del sistema, proporcionando instrucciones claras para su configuración, operación y mantenimiento.
- Preparar y realizar una presentación efectiva del producto ante el público objetivo, destacando sus características, funcionalidades y beneficios, con el objetivo de generar interés, obtener retroalimentación valiosa y promover su adopción en el ámbito de la domótica y la tecnología del hogar inteligente.

4.DESARROLLO

- En primer lugar, se aborda la domotización de una maqueta de casa, lo cual implica la integración de dispositivos electrónicos y sistemas de control para automatizar funciones del hogar. Para ello, se utilizará la placa de desarrollo ESP32, reconocida por su versatilidad y capacidades inalámbricas, lo que la hace idónea para este propósito.
- El proceso comienza con la configuración y programación de la placa ESP32, que incluye la instalación del entorno de desarrollo integrado (IDE) de Arduino y la escritura de código en lenguaje C/C++ para controlar los diferentes componentes de

la maqueta, como luces, sensores y actuadores. Este aspecto técnico se centra en la comprensión de los principios de la programación y el funcionamiento de los dispositivos electrónicos.

- Por otro lado, se desarrollará una interfaz de control web responsiva para permitir el monitoreo y la gestión remota de la maqueta de la casa. Esta parte del proyecto se basa en tecnologías web como HTML, CSS y JavaScript, que se utilizan para diseñar una interfaz intuitiva y fácil de usar. Además, se empleará un servidor web local, como XAMPP, para facilitar la comunicación entre la placa ESP32 y la página web.
- La integración de estos componentes implica la sincronización entre el hardware y el software, asegurando que la maqueta de la casa pueda ser controlada de manera eficiente y segura desde cualquier dispositivo con acceso a internet. Para lograrlo, se seguirán procedimientos específicos que involucran la configuración de la red inalámbrica, la asignación de direcciones IP y la implementación de protocolos de comunicación adecuados

Tecnologías Utilizadas

XAMPP.

XAMPP es un paquete de software gratuito y de código abierto que facilita la creación de un entorno de desarrollo local para sitios web dinámicos. El nombre "XAMPP" es un acrónimo que proviene de las primeras letras de los componentes del paquete:

- **X:** Sistema operativo (Cross-platform): XAMPP.
- **A:** Servidor web Apache: Apache es uno de los servidores web más populares del mundo y forma la base del entorno de desarrollo de XAMPP.
- **M:** Base de datos MySQL: XAMPP incluye MySQL como el sistema de gestión de bases de datos predeterminado, lo que permite a los desarrolladores almacenar y recuperar datos para sus aplicaciones web.
- **P:** PHP: PHP es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo web del lado del servidor.

- **P: phpMyAdmin:** phpMyAdmin es una herramienta de administración de bases de datos basada en web que facilita la administración de bases de datos MySQL a través de una interfaz gráfica de usuario.

Arduino

Arduino es una plataforma de hardware de código abierto y fácil de usar que se utiliza para crear proyectos electrónicos interactivos. Consiste en una placa de circuito impreso con un microcontrolador y un entorno de desarrollo integrado (IDE) que facilita la programación y la interacción con los componentes electrónicos.

- **Entorno de Desarrollo Integrado (IDE):** El entorno de desarrollo integrado de Arduino es un software que permite escribir, compilar y cargar programas en las placas Arduino. El IDE proporciona un editor de código, herramientas de compilación y carga, y una interfaz de usuario para facilitar el desarrollo de proyectos.
- **Lenguaje de Programación:** Arduino utiliza un lenguaje de programación basado en Wiring, que es similar a C/C++. El lenguaje de programación de Arduino incluye una serie de funciones y librerías predefinidas que facilitan la interacción con los componentes electrónicos y la creación de proyectos complejos.
- **Librerías y Shields:** Arduino cuenta con una amplia comunidad de desarrolladores que han creado librerías y shields (placas de expansión) que amplían las capacidades de las placas Arduino. Estas librerías proporcionan funciones adicionales para controlar diferentes tipos de dispositivos y sensores, mientras que los shields ofrecen funcionalidades específicas, como conectividad Wi-Fi, GPS, control de motores, etc.

HTML

HTML es el lenguaje estándar utilizado para crear y diseñar páginas web. Se trata de un lenguaje de marcado que define la estructura y el contenido de una página web utilizando una serie de elementos y etiquetas predefinidos.

- **Estructura:** HTML define la estructura básica de una página web, incluyendo elementos como encabezados, párrafos, listas, enlaces, imágenes, formularios,

tablas, entre otros. Estos elementos permiten organizar y presentar el contenido de una página web de manera semántica y estructurada.

- **Etiquetas:** HTML utiliza etiquetas para marcar y delimitar diferentes partes del contenido de una página web. Cada etiqueta comienza con un símbolo de menor que (<) seguido del nombre de la etiqueta, y termina con un símbolo de mayor que (>). Por ejemplo, <p> se utiliza para marcar un párrafo, <h1> para un encabezado de nivel 1, <a> para un enlace, etc.
- **Atributos:** Las etiquetas HTML pueden contener atributos que proporcionan información adicional sobre el elemento.
- **Elementos Anidados:** Los elementos HTML pueden anidarse dentro de otros elementos para crear una estructura jerárquica. Por ejemplo, un párrafo (<p>) puede contener texto y otros elementos como enlaces (<a>), imágenes (), listas (,) y más.
- **Compatibilidad con Navegadores:** HTML es un estándar web ampliamente adoptado y es compatible con la mayoría de los navegadores web modernos.

CSS

CSS es un lenguaje de estilo utilizado para controlar el aspecto y el diseño de las páginas web. Permite a los desarrolladores web definir estilos y reglas que se aplican a los elementos HTML, lo que les permite modificar aspectos como el color, la tipografía, el espaciado, el diseño y otros aspectos visuales de una página web.

- **Selector y Propiedades:** CSS utiliza selectores para apuntar a elementos específicos en una página web y aplicar estilos a esos elementos. Cada regla CSS consiste en un selector y un conjunto de propiedades y valores.
- **Cascada y Herencia:** CSS sigue el principio de cascada, lo que significa que las reglas CSS se aplican en un orden específico y las reglas más específicas tienen prioridad sobre las reglas más generales. Además, los estilos se heredan de los elementos padres a los elementos hijos, lo que permite establecer estilos comunes para grupos de elementos.
- **Unidades de Medida y Valores:** CSS admite una variedad de unidades de medida y valores para especificar dimensiones y propiedades de estilo.

- **Estilos Externos e Internos:** CSS se puede aplicar a una página web de varias maneras. Los estilos CSS pueden estar incrustados directamente en el documento HTML utilizando etiquetas `<style>` o pueden estar en un archivo CSS externo vinculado al documento HTML mediante la etiqueta `<link>`. También es posible aplicar estilos CSS directamente en línea utilizando el atributo `style` en los elementos HTML.
- **Compatibilidad con Navegadores:** CSS es un estándar web ampliamente adoptado y es compatible con la mayoría de los navegadores web modernos. Sin embargo, es importante tener en cuenta que algunas características de CSS pueden comportarse de manera diferente en diferentes navegadores, por lo que es importante realizar pruebas de compatibilidad cruzada.

JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos que se utiliza principalmente en el desarrollo web para agregar interactividad y funcionalidad a las páginas web. Junto con HTML y CSS, JavaScript forma la base de la programación en el lado del cliente en el entorno web.

- **Interactividad:** Una de las principales características de JavaScript es su capacidad para crear páginas web interactivas y dinámicas. Con JavaScript, los desarrolladores pueden responder a eventos del usuario, como clics de ratón, pulsaciones de teclas, desplazamientos, etc., y realizar acciones en consecuencia, como mostrar u ocultar elementos, cambiar el contenido de la página, validar formularios, etc.
- **Manipulación del DOM:** JavaScript permite a los desarrolladores acceder y manipular el Document Object Model (DOM), que es una representación en memoria de la estructura de una página web.
- **Programación Event-Driven:** JavaScript es un lenguaje de programación orientado a eventos, lo que significa que gran parte de su funcionalidad se basa en la respuesta a eventos del usuario y del sistema. Los eventos son desencadenados por acciones como clics de ratón, pulsaciones de teclas, cambios en el estado del documento, etc., y JavaScript puede ejecutar funciones específicas en respuesta a estos eventos.

- **Librerías y Frameworks:** JavaScript cuenta con una gran cantidad de librerías y frameworks que simplifican el desarrollo web y proporcionan funcionalidades adicionales. Algunas de las librerías y frameworks más populares incluyen jQuery, React.js, AngularJS, Vue.js, entre otros, que ofrecen soluciones para el desarrollo de aplicaciones web, interfaces de usuario interactivas y aplicaciones de una sola página (SPA), entre otros.
- **Compatibilidad con Navegadores:** JavaScript es un estándar web ampliamente adoptado y es compatible con la mayoría de los navegadores web modernos. Los navegadores ejecutan el código JavaScript directamente en el cliente, lo que permite una experiencia de usuario interactiva y receptiva sin necesidad de comunicación con el servidor.

PHP

PHP es un lenguaje de programación del lado del servidor ampliamente utilizado para el desarrollo web. Se utiliza principalmente para generar contenido dinámico e interactuar con bases de datos, pero también se puede usar para realizar una variedad de tareas del lado del servidor, como manipulación de archivos, envío de correos electrónicos, gestión de sesiones, entre otros.

- **Interpretado y Del Lado del Servidor:** PHP es un lenguaje de script del lado del servidor, lo que significa que el código PHP se ejecuta en el servidor web antes de que se envíe el contenido al navegador del usuario. El servidor interpreta el código PHP y genera HTML, que luego se envía al navegador para su visualización.
- **Integración con HTML:** Una de las características clave de PHP es su capacidad para integrarse fácilmente con HTML. El código PHP puede incluirse directamente en archivos HTML utilizando etiquetas especiales `<?php ... ?>`, lo que permite a los desarrolladores mezclar código PHP y HTML para crear páginas web dinámicas y personalizadas.
- **Acceso a Bases de Datos:** PHP ofrece una amplia gama de funciones y extensiones para interactuar con bases de datos relacionales, como MySQL, PostgreSQL y SQLite. Esto permite a los desarrolladores crear aplicaciones web

dinámicas que almacenan y recuperan datos de una base de datos de manera eficiente.

- **Gestión de Formularios:** PHP es especialmente útil para el procesamiento de formularios web. Puede recibir datos de formularios HTML enviados por los usuarios, validar y filtrar la entrada del usuario para evitar ataques de seguridad, y procesar los datos ingresados en el formulario antes de almacenarlos en una base de datos o enviarlos por correo electrónico.
- **Compatibilidad con una Amplia Gama de Servidores Web:** PHP es compatible con una amplia gama de servidores web, incluidos Apache, Nginx e IIS, lo que lo hace fácilmente adaptable a diferentes entornos de alojamiento web y sistemas operativos.

MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto ampliamente utilizado en el desarrollo web. Se utiliza para almacenar, gestionar y recuperar datos en aplicaciones web dinámicas y sitios web.

- **Bases de Datos Relacionales:** MySQL sigue el modelo de base de datos relacional, lo que significa que los datos se almacenan en tablas que tienen relaciones entre sí.
- **Lenguaje SQL:** MySQL utiliza el lenguaje de consulta estructurado (SQL) para interactuar con la base de datos. SQL es un lenguaje estándar que se utiliza para realizar consultas, insertar datos, actualizar registros y eliminar datos en una base de datos relacional.
- **Escalabilidad y Rendimiento:** MySQL es conocido por su escalabilidad y rendimiento, lo que lo hace adecuado para aplicaciones web de alto tráfico y grandes conjuntos de datos. Soporta técnicas de optimización como el índice de tablas y el almacenamiento en caché para mejorar el rendimiento de las consultas y reducir la latencia en la recuperación de datos.
- **Seguridad:** MySQL ofrece una variedad de características de seguridad para proteger los datos almacenados en la base de datos. Esto incluye la

autenticación de usuarios, la gestión de permisos y privilegios, el cifrado de datos, la auditoría de eventos y más.

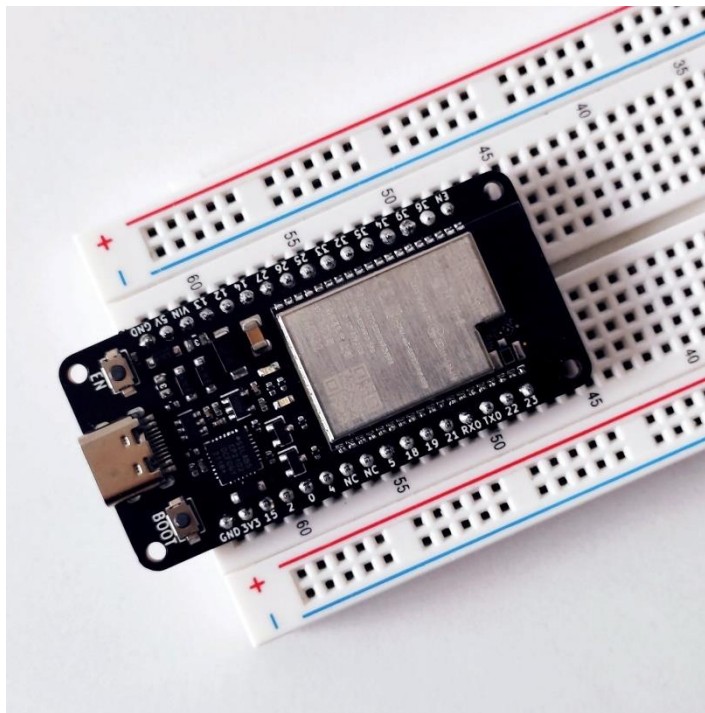
ESP32

El ESP32 es un microcontrolador de bajo costo y bajo consumo de energía que incluye Wi-Fi y Bluetooth integrados. Desarrollado por Espressif Systems, el ESP32 es una plataforma popular para una amplia gama de aplicaciones de IOT (Internet de las cosas) y proyectos embebidos debido a su versatilidad, potencia y facilidad de uso.

- **Microcontrolador de Doble Núcleo:** El ESP32 cuenta con un microcontrolador de doble núcleo Xtensa LX6 de 32 bits que puede operar a frecuencias de hasta 240 MHz. La capacidad de procesamiento de doble núcleo permite ejecutar múltiples tareas de forma simultánea y manejar aplicaciones más complejas con mayor eficiencia.
- **Wi-Fi y Bluetooth Integrados:** Una de las características más destacadas del ESP32 es su conectividad Wi-Fi y Bluetooth integrada. Esto permite al ESP32 conectarse a redes Wi-Fi existentes y comunicarse con otros dispositivos a través de Bluetooth, lo que lo hace ideal para aplicaciones de IOT que requieren conectividad inalámbrica.
- **Bajo Consumo de Energía:** A pesar de su potencia y capacidades, el ESP32 es conocido por su bajo consumo de energía, lo que lo hace adecuado para aplicaciones de batería y dispositivos portátiles. El ESP32 incluye modos de bajo consumo que permiten reducir significativamente el consumo de energía cuando el dispositivo no está en uso activo.
- **Amplia Gama de Periféricos:** El ESP32 cuenta con una amplia gama de periféricos integrados, incluyendo puertos GPIO, interfaces SPI, I2C, UART, ADC, DAC, entre otros. Esto permite conectar una variedad de sensores, actuadores y dispositivos externos al ESP32 para ampliar su funcionalidad y aplicaciones.
- **Facilidad de Programación y Desarrollo:** El ESP32 se puede programar utilizando el entorno de desarrollo Arduino, lo que lo hace accesible para una amplia comunidad de desarrolladores. Además, Espressif Systems proporciona

un SDK (kit de desarrollo de software) oficial y herramientas de desarrollo integradas que facilitan el desarrollo de aplicaciones para el ESP32.

- **Versatilidad y Flexibilidad:** Debido a sus capacidades y características, el ESP32 es adecuado para una amplia variedad de aplicaciones de IOT, incluyendo automatización del hogar, monitoreo ambiental, dispositivos portátiles, seguimiento de activos, control remoto, y mucho más. Su versatilidad y flexibilidad lo convierten en una opción popular para proyectos embebidos y de IOT.



1 Modulo ESP32

Leds

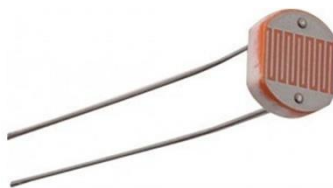
Los Leds son componentes electrónicos que emiten luz cuando se les aplica una corriente eléctrica. Podemos utilizar Leds para indicar el estado de diferentes dispositivos en nuestra casa inteligente, como luces encendidas o apagadas, alarmas activadas, etc.



2 Diodo LED

Sensores

Los sensores son dispositivos que detectan cambios en el entorno y los convierten en señales eléctricas. Podemos utilizar una variedad de sensores en nuestra maqueta de casa inteligente, como sensores de temperatura, humedad, movimiento, luz, gas, etc. Estos sensores nos permiten monitorear diferentes aspectos del entorno de la casa y tomar acciones en consecuencia.



3 Sensor LDR

Pantalla LCD

Las pantallas LCD (Liquid Crystal Display) son dispositivos de visualización utilizados para mostrar información en forma de texto, imágenes o gráficos. Son ampliamente utilizadas en una variedad de dispositivos electrónicos, desde teléfonos inteligentes y televisores hasta equipos médicos y sistemas de información pública.



4 Pantalla LCD

Metodología Utilizada

Para la gestión y ejecución de este proyecto, se ha optado por la metodología DevOps. DevOps es una filosofía de desarrollo de software que promueve la colaboración estrecha y la integración continua entre los equipos de desarrollo (Dev) y operaciones (Ops). Su objetivo principal es acelerar la entrega de software alineando los objetivos de negocio con los procesos de desarrollo y operación. En el contexto de este proyecto, la metodología DevOps se traduce en la adopción de prácticas que permiten la automatización de los procesos de desarrollo, pruebas, despliegue y monitorización. Esto implica una integración continua de cambios en el código, pruebas automatizadas para garantizar la calidad del software, despliegues frecuentes y confiables en entornos de producción, y una monitorización constante del rendimiento y la disponibilidad del sistema.

Planificación del Proyecto

En el caso del proyecto de desarrollo de software, se utiliza un diagrama de Gantt para representar visualmente estas actividades a lo largo del tiempo. Cada tarea se identifica por su nombre y se estima su duración en semanas. Esta representación permite una gestión eficiente del tiempo y recursos, facilitando el seguimiento del progreso y la identificación de posibles desviaciones en el plan inicial.

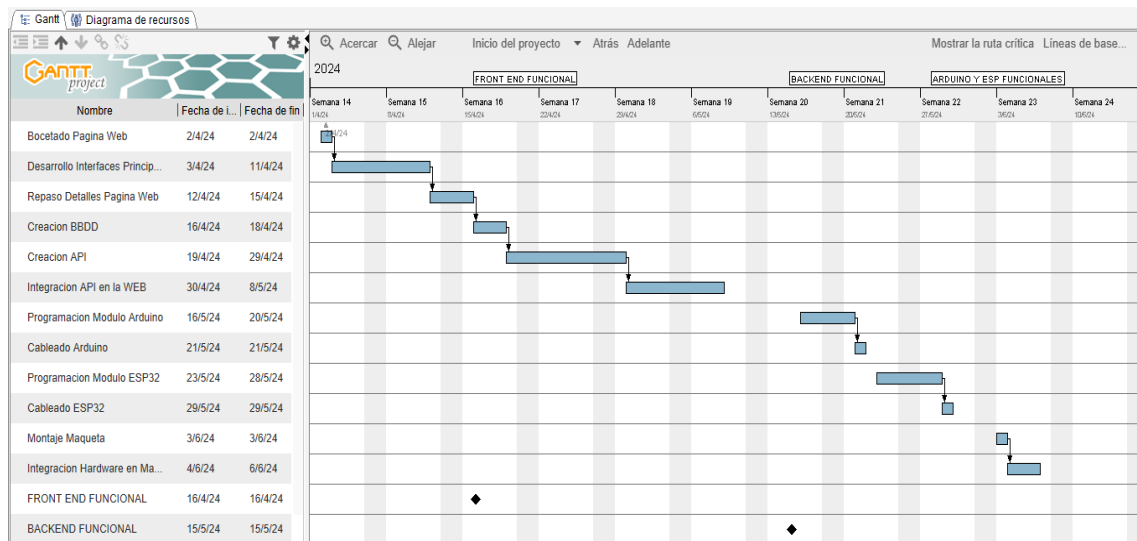
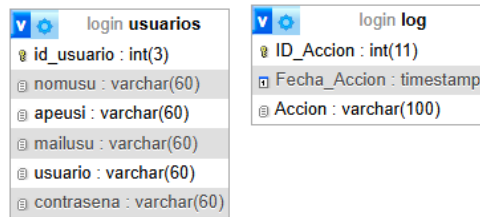


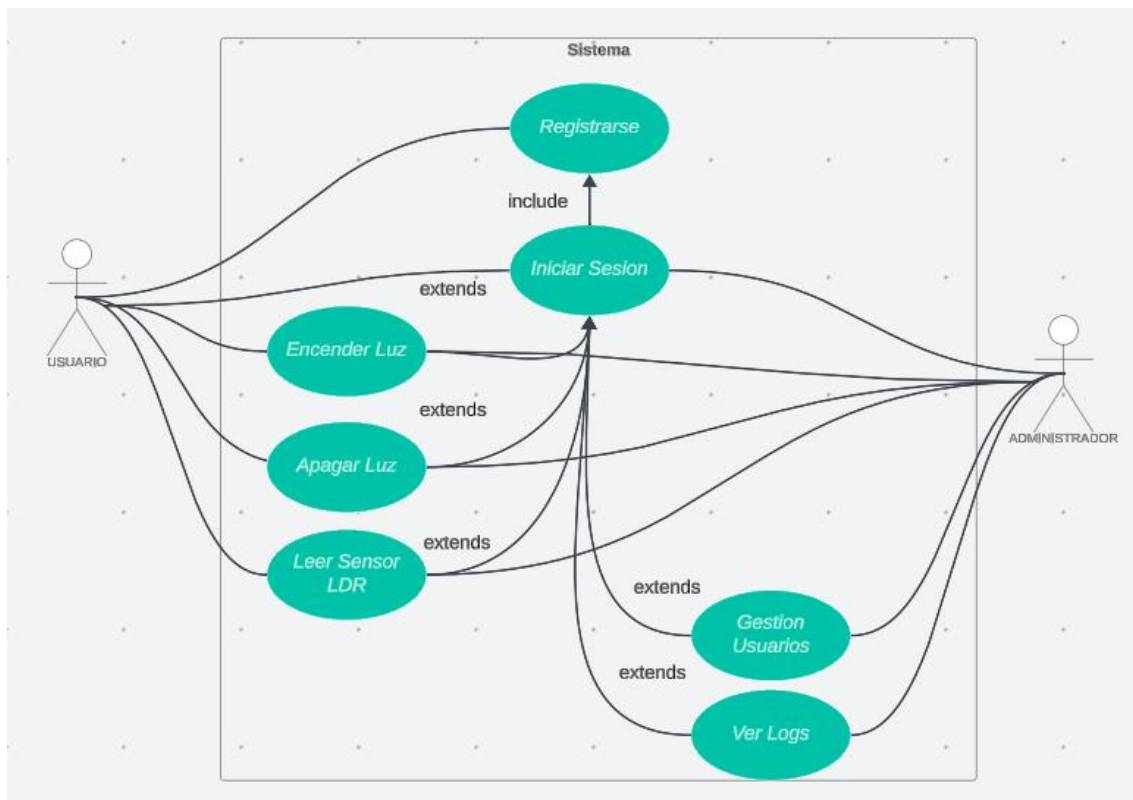
Diagrama BBDD



La tabla "usuarios" almacena la información básica de los usuarios registrados, incluyendo nombre, apellido, correo electrónico, nombre de usuario y contraseña. Es fundamental para gestionar el acceso al sistema de control domótico, permitiendo la autenticación de usuarios durante el inicio de sesión y la asignación de roles y privilegios.

También tenemos una tabla llamada "log" en la cual guardaremos todos los eventos que se registren en la base de datos (Registrar Usuario, Eliminar Usuario) .

Diagrama de Casos de Uso



Formato 2 Columnas

Iniciar Sesión

Actor Principal: Usuario

Precondiciones: El usuario se encuentra registrado en el sistema

Garantía de Éxito: Se inicia sesión correctamente y se redirige a la página principal

Escenario Principal de Éxito:

Usuario	Sistema
1- El caso de uso inicia cuando el usuario accede al portal web	2- El sistema muestra el formulario para que el usuario introduzca sus datos de autenticación
3- El cliente introduce los datos	4- El sistema verifica la información mediante una consulta a la BBDD
	5- El sistema redirige al usuario a la página principal.

Alternativas

4a – Los datos introducidos son incorrectos:

- Se le muestra un PopUp al cliente y se le reenvía al formulario de inicio de Sesión.

Registrarse

Actor Principal: Usuario

Precondiciones: Ninguna

Garantía de Éxito: Se registra correctamente el usuario en el sistema

Escenario Principal de Éxito

Actor	Sistema
1- El caso de uso se inicia cuando el usuario pulsa en la opción de registrarse	2- El sistema redirige al usuario a la página de registro
4- El usuario introduce los datos pedidos por el sistema.	3- El sistema pide los datos necesarios para registrar al usuario
	5- El sistema verifica que todos los datos sean correctos
	6- El sistema muestra un Pop Up indicando que se ha registrado con éxito.

Alternativas

5a – Los datos introducidos son incorrectos.

- Se le muestra una advertencia al usuario.

5b – El usuario ya está registrado

- Se le informa al usuario y se le reenvía a la página de login.

Encender Luz

Actor Principal: Usuario

Precondiciones: Tener la sesión iniciada en el sistema

Garantía de Éxito: Se enciende la luz que el usuario ordena cuando el usuario lo ordena

Escenario Principal de Éxito:

Actor	Sistema
1- El caso de uso inicia cuando el usuario hace click en uno de los botones destinados al control de luces	2- El sistema comprueba cual es el botón que el usuario ha presionado 3- El sistema comprueba el estado de la luz (Encendido/Apagado) 4- El sistema enciende la luz.

Alternativas:

3a – La luz ya se encuentra encendida

- El sistema apaga la luz

Apagar Luz

Actor Principal: Usuario

Precondiciones: Tener la sesión iniciada en el sistema

Garantía de Éxito: Se apaga la luz que el usuario ordena cuando el usuario lo ordena

Escenario Principal de Éxito:

Actor	Sistema
1- El caso de uso inicia cuando el usuario hace click en uno de los botones destinados al control de luces	2- El sistema comprueba cual es el botón que el usuario ha presionado 3- El sistema comprueba el estado de la luz (Encendido/Apagado) 4- El sistema apaga la luz.

Alternativas:

3a – La luz ya se encuentra apagada

- El sistema enciende la luz

Leer Sensor LDR

Actor Principal: Usuario

Precondiciones: Tener la sesión iniciada en el sistema

Garantía de Éxito: Se muestran los datos de luminosidad del entorno en la pantalla LCD

Escenario Principal de Éxito:

Actor	Sistema
1- El caso de uso inicia cuando el usuario se autentica en el sistema	2- El sistema comienza un bucle de lectura del sensor LDR 3- El sistema muestra los datos de luminosidad en la pantalla LCD

Gestión Usuarios

Actor Principal: Administrador

Precondiciones: Tener la sesión iniciada en el sistema como administrador

Garantía de Éxito: Se listan los usuarios correctamente con opción a eliminar el usuario deseado

Escenario Principal de Éxito:

Actor	Sistema
1- El caso de uso inicia cuando el administrador pincha en la ventana de administración	2 El sistema muestra la pagina con los usuarios existentes
3- El administrador pulsa en eliminar sobre el usuario deseado	4- El sistema elimina al usuario de la BBDD

Alternativas

2a – No hay usuarios registrados

- Se mostrará una tabla vacía

Ver Logs

Actor Principal: Administrador

Precondiciones: Tener la sesión iniciada en el sistema como administrador

Garantía de Éxito: Se listan los logs de la BBDD correctamente

Escenario Principal de Éxito:

Actor	Sistema
1- El caso de uso inicia cuando el administrador pincha en la ventana de logs	2- El sistema muestra la página con registros existentes

Alternativas

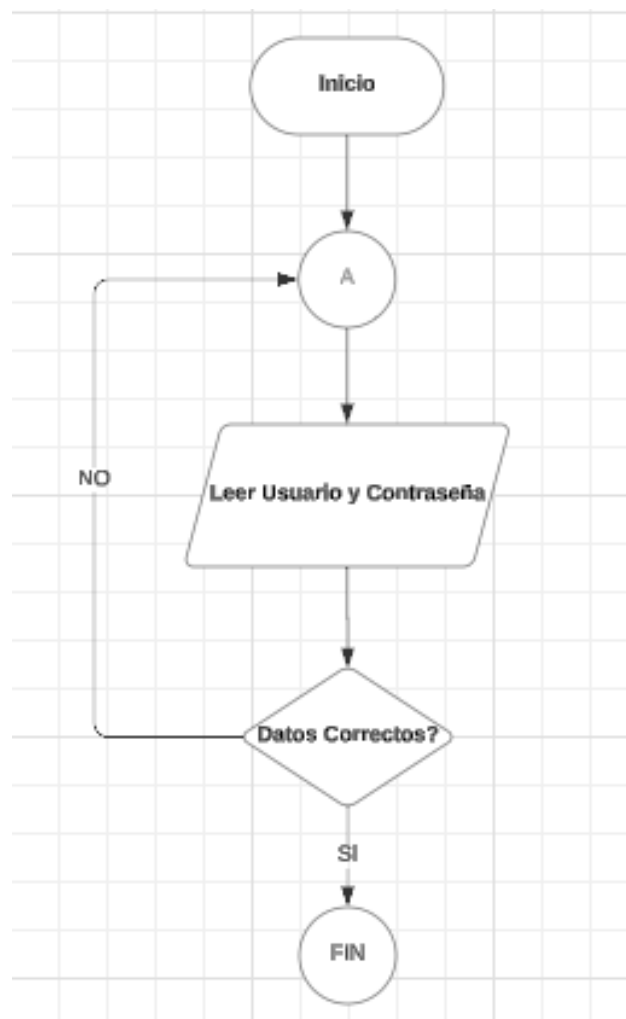
2a – No hay registros

- Se mostrará una tabla vacía

Diagramas de Flujo

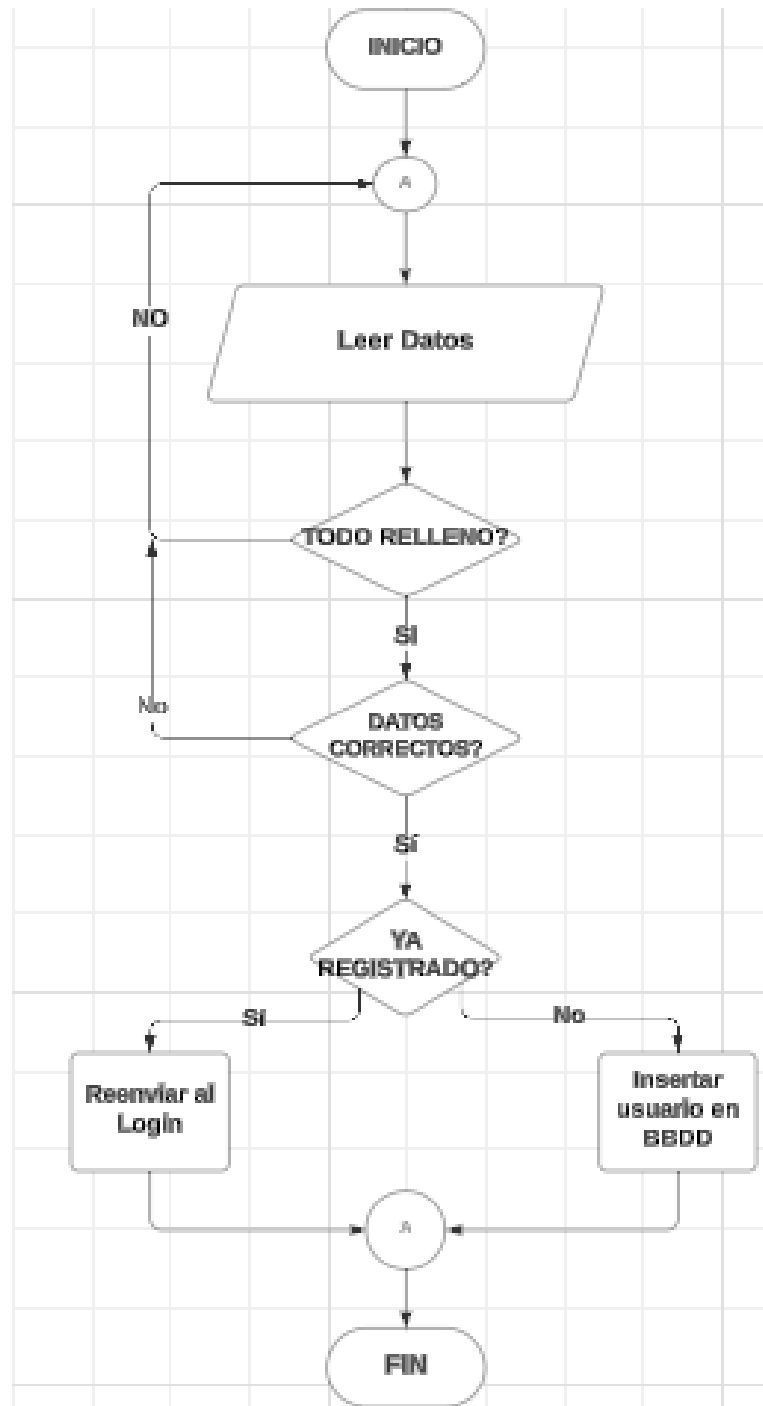
Iniciar sesión

Este diagrama representa el camino que el software sigue para que un usuario pueda iniciar sesión en nuestro portal web.



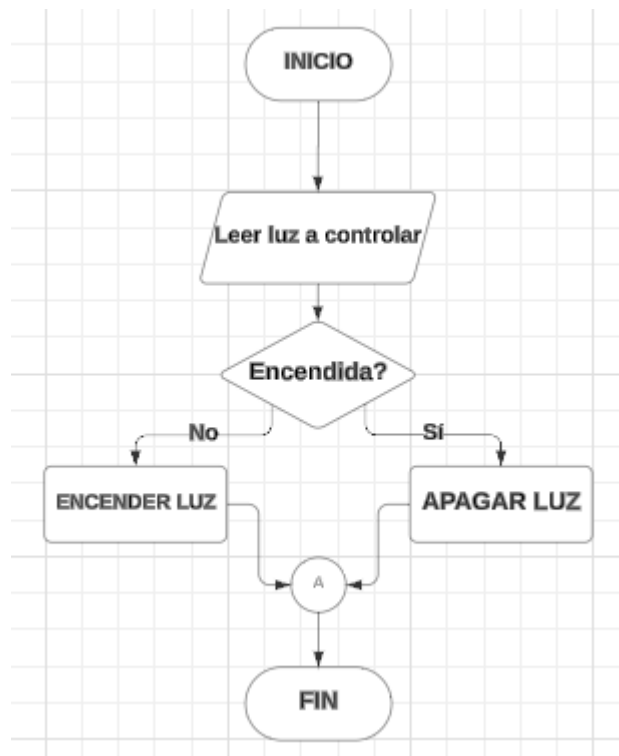
Registrarse

El siguiente diagrama de flujo muestra como actuará el software a la hora de dar de alta a un usuario nuevo en la BBDD



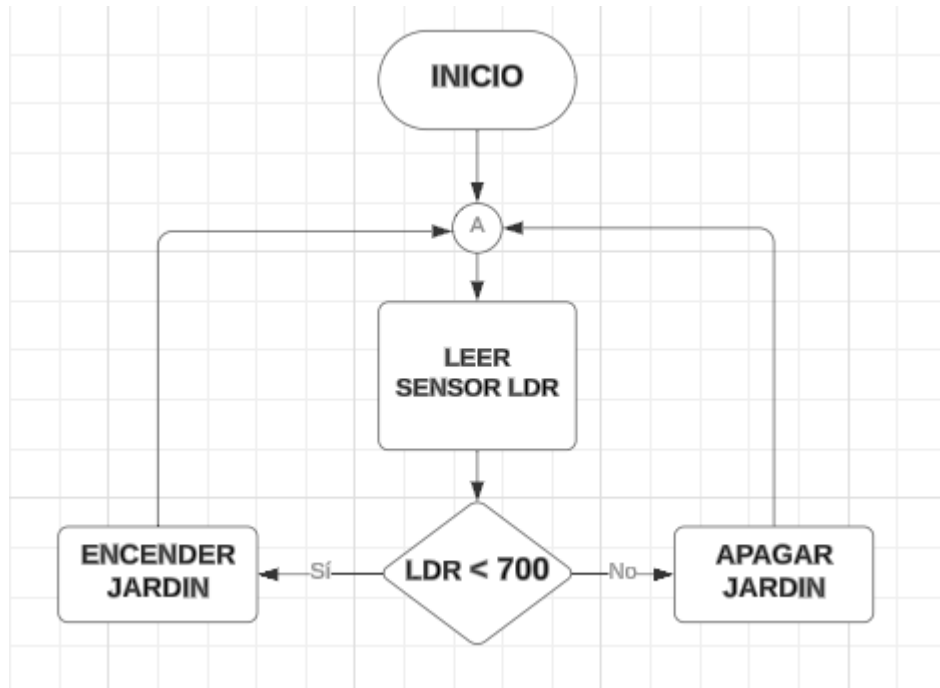
Control Luces

Este diagrama indica el flujo del programa a la hora de gestionar las luces de la casa.



Luces Automáticas

Este otro diagrama explica la funcionalidad que sirve para controlar las luces de manera automática en base al nivel de luz del entorno.



Explicación Código

En este apartado vamos a revisar las que considero que son las partes más interesantes del código desarrollado.

Verificación Administrador

```
// Verificar si el usuario tiene permisos de administración
$query1 = mysqli_query($conn, "SELECT * FROM Usuarios WHERE usuario = '$logeado'");
$filas1 = $query1->fetch_all(MYSQLI_ASSOC);
foreach ($filas1 as $fila) {
    if ($fila["id_usuario"] != 1) {
        header("Location: index.php");
    }
}
```

El código verifica si un usuario tiene permisos de administrador mediante una consulta a la base de datos que busca al usuario en la tabla "Usuarios". Luego, recorre los resultados y verifica si el ID del usuario es diferente de 1. Si no es administrador, se redirige al usuario a la página "index.php".

Conexión BBDD

```
function conectar(){
    $conn = new mysqli("localhost","root","","Login",3306);
    return $conn;
}
```

El código PHP establece una conexión a una base de datos MySQL mediante la función conectar, que devuelve una instancia de la clase mysqli con los parámetros de configuración de la conexión, incluyendo la dirección del servidor, nombre de usuario, contraseña, nombre de la base de datos y puerto.

Leer Tarjeta

```
// A la espera de tarjeta
if ( ! mfrc522.PICC_IsNewCardPresent() ) {
return false;
}
if ( ! mfrc522.PICC_ReadCardSerial() ) { // Leer ID Tarjeta
return false;
}
tagID = "";
for ( uint8_t i = 0; i < 4; i++) { // Recorre el ID
//readCard[i] = mfrc522.uid.uidByte[i];
tagID.concat(String(mfrc522.uid.uidByte[i], HEX)); // Concatena todo en una sola String
}
tagID.toUpperCase();
mfrc522.PICC_HaltA();
acceso = 1;
return true;
```

El código en el fragmento es una función que busca una tarjeta RFID, lee su ID, lo convierte a un string y lo guarda en la variable tagID. Después, para de leer la tarjeta y la función devuelve true, indicando que se ha encontrado la tarjeta.

Control Luces desde Arduino

```
// Ruteo para las habitaciones interiores
server.on("/h1", []() { controlLuces(15); });
server.on("/h2", []() { controlLuces(4); });
server.on("/h3", []() { controlLuces(16); });
server.on("/h4", []() { controlLuces(17); });

// Ruteo para las zonas exteriores
server.on("/z1", []() { controlLuces(5); });
server.on("/z2", []() { controlLuces(19); });
server.on("/z3", []() { controlLuces(21); });
server.on("/z4", []() { controlLuces(22); });
```

Define rutas para recibir solicitudes HTTP y, según la ruta recibida, llama a la función controlLuces con un parámetro numérico que indica qué luces controlar. Por ejemplo, la ruta /h1 controla las luces de la habitación "h1" y la ruta /z1 controla las luces de la zona exterior "z1". La función controlLuces no se muestra en el código, pero se supone que se encarga de enviar señales a los dispositivos de iluminación correspondientes para encender o apagar las luces.

Control Luces JavaScript

```
function encenderLuz(habitacion) {  
    if (habitacion == "habitacion1") {  
        var newWindow = window.open("http://192.168.0.41/h1");  
        setTimeout(() => newWindow.close(), 7);  
    } else if (habitacion == "habitacion2") {  
        var newWindow = window.open("http://192.168.0.41/h2");  
        setTimeout(() => newWindow.close(), 7);  
    } else if (habitacion == "habitacion3") {  
        var newWindow = window.open("http://192.168.0.41/h3");  
        setTimeout(() => newWindow.close(), 7);  
    } else if (habitacion == "habitacion4") {  
        var newWindow = window.open("http://192.168.0.41/h4");  
        setTimeout(() => newWindow.close(), 7);  
    }  
    console.log("Se ha encendido la luz de la habitación: " + habitacion);  
}
```

El código define una función llamada `encenderLuz` que recibe un parámetro `habitación`. Dentro de la función, se utiliza una sentencia `if-else` para abrir una ventana emergente con una URL específica en función del valor de `habitación`. La URL corresponde a una dirección IP y un identificador de habitación. Después de abrir la ventana emergente, se establece un temporizador de 7 milisegundos para cerrarla automáticamente. Finalmente, se imprime un mensaje en la consola indicando que se ha encendido la luz de la habitación correspondiente.

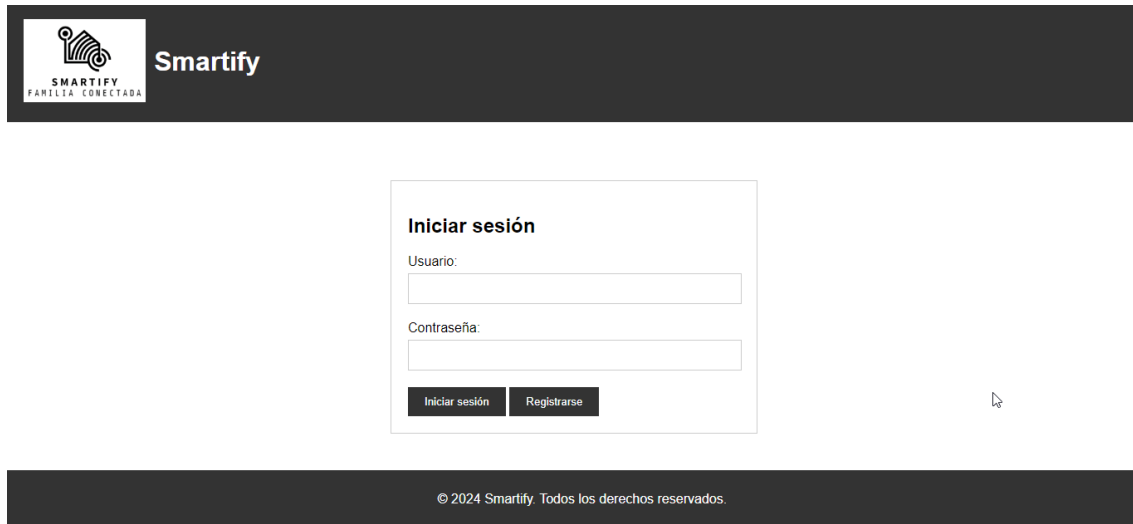
Resultados Y Análisis

El desarrollo de este proyecto me ha servido para aprender muchísimas cosas acerca del mundo tanto de la programación como del hardware, a lo largo de este proyecto he conseguido lo siguiente, implementar un sistema inteligente de control del hogar enfocado en la autenticación segura de usuarios, el manejo eficiente de luces y la lectura precisa de sensores. También he desarrolló una interfaz intuitiva y responsiva para dispositivos móviles y PC, asegurando una experiencia de usuario óptima. La integración de medidas de seguridad robustas como la encriptación y la gestión de sesiones garantiza la protección de datos, mientras que el diseño sencillo de la interfaz facilitó la navegación y el uso del sistema. Además, la implementación de sensores ambientales proporcionó datos en tiempo real, esenciales para la automatización y mejora del confort en el hogar.

Capturas del Proyecto

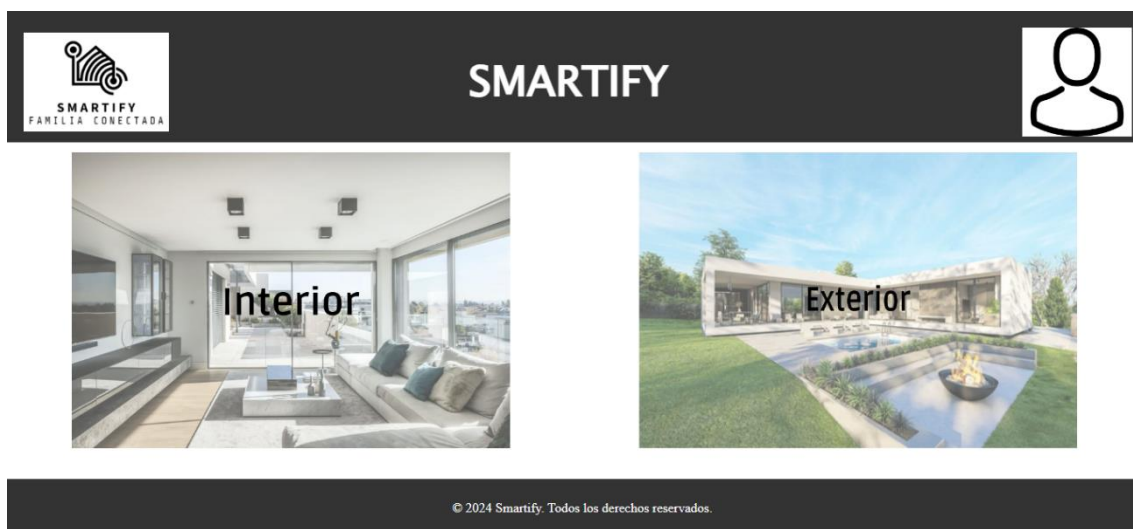
Pantalla de Login.

Esta pantalla será la primera que vea el usuario a la hora de entrar a la WEB.



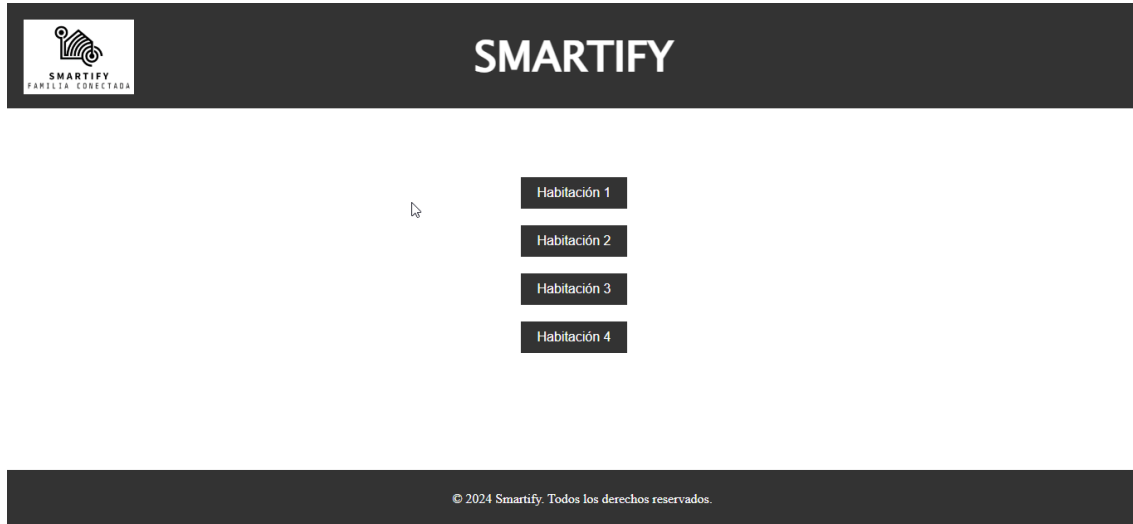
Pantalla Principal

Desde esta pantalla el usuario puede elegir controlar las luces interiores o las exteriores.



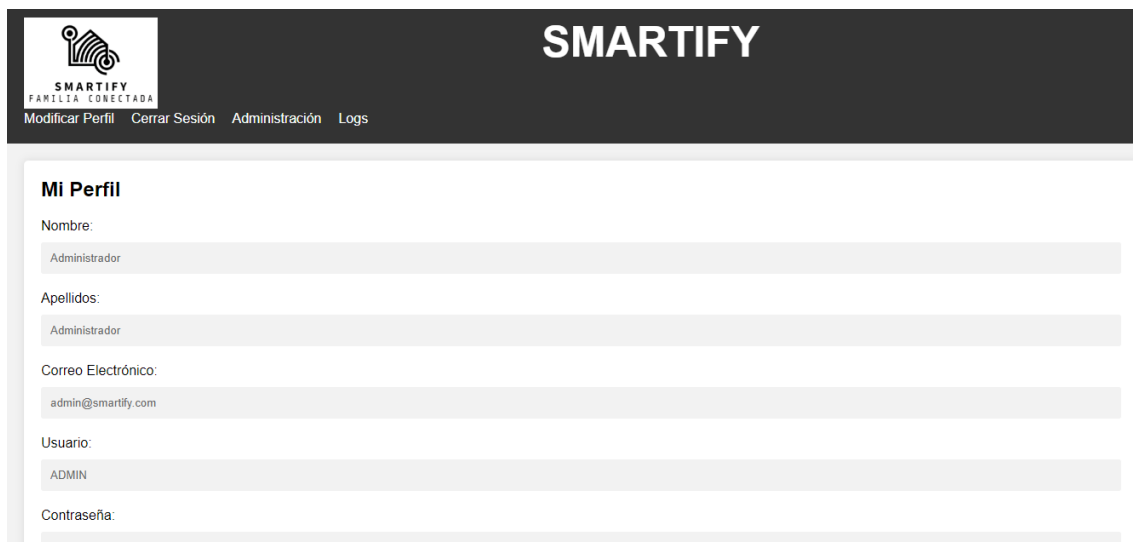
Control de Luces

En esta pantalla se muestra un botón por cada habitación/zona desde los cuales controlaremos la iluminación.



Perfil

En esta pantalla mostramos los datos de perfil del usuario.



Modificar Perfil

Aquí el usuario puede modificar tanto sus datos personales como sus datos de acceso.

Modificar Perfil

Nombre:

Administrador

Apellidos:

Administrador

Correo Electrónico:

admin@smartify.com

Usuario:

ADMIN

Contraseña:

Confirmar Contraseña:

Guardar Cambios

Administrar Usuarios

Esta pantalla solamente la puede ver el Administrador y es desde donde podremos eliminar los usuarios.

SMARTIFY - Administración

[Inicio](#) - [Cerrar Sesión](#)

Usuarios

ID	Nombre	Correo Electrónico	Acción
2	Patrick	patrick.petterson@stortrec.com	Eliminar
3	Gabriel	gabriel.valles@stortrec.com	Eliminar
4	Arturo	arturo.pozas@stortrec.com	Eliminar
7	Luis Miguel	cancerbero.x.98@gmail.com	Eliminar
8	Alejandro	alejandro.rico@campus.es	Eliminar

© 2024 Smart Home. Todos los derechos reservados.

Ver Logs BBDD

Al igual que la anterior esta pantalla solamente la puede ver el Administrador y sirve para ver el registro de actividad que se ha registrado en la Base de Datos.

SMARTIFY - Administración		
Inicio Cerrar Sesión		
Registro de Acciones		
ID Acción	Fecha Acción	Acción
1	2024-06-03 09:38:18	USUARIO ELIMINADO
2	2024-06-03 09:39:13	USUARIO INSERTADO
3	2024-06-03 09:39:13	USUARIO INSERTADO
4	2024-06-03 09:42:28	USUARIO INSERTADO
5	2024-06-03 09:42:28	USUARIO INSERTADO
6	2024-06-03 09:44:02	USUARIO ACTUALIZADO
7	2024-06-03 11:48:14	USUARIO INSERTADO
8	2024-06-03 11:48:14	USUARIO INSERTADO
9	2024-06-03 11:49:04	USUARIO ELIMINADO
10	2024-06-03 11:52:45	USUARIO INSERTADO
11	2024-06-03 11:52:45	USUARIO INSERTADO
12	2024-06-03 11:53:28	USUARIO ELIMINADO

5.CONCLUSION

En conclusión, el desarrollo de este sistema de domótica basado en la placa de desarrollo ESP32 ha sido un proyecto desafiante y gratificante que me ha permitido integrar eficazmente hardware y software para crear una solución completa de control remoto para una maqueta de casa. A lo largo de este proyecto, he diseñado e implementado un sistema que abarca control de iluminación, climatización y seguridad, integrando diversos sensores y actuadores para automatizar múltiples aspectos del entorno doméstico. He gestionado un servidor web, desarrollando una interfaz de usuario web responsiva que proporciona una experiencia fluida y accesible en dispositivos variados. La comunicación en tiempo real entre el servidor web y los dispositivos físicos se ha logrado mediante protocolos eficientes, asegurando confiabilidad y seguridad. Las pruebas exhaustivas realizadas han garantizado la funcionalidad del sistema en diversos escenarios, y el análisis de datos recopilados ha permitido identificar áreas de mejora.

6.LÍNEAS FUTURAS DE INVESTIGACIÓN

1. **Optimización de Protocolos de Comunicación:** Investigar y desarrollar nuevos protocolos de comunicación que mejoren la eficiencia, velocidad y seguridad en la interacción entre el servidor web y los dispositivos conectados.
2. **Integración con Asistentes Virtuales:** Explorar la integración del sistema con asistentes virtuales como Alexa, Google Assistant y Siri para permitir el control por voz de la maqueta de la casa.
3. **Ampliación de Funcionalidades:** Investigar y añadir nuevas funcionalidades al sistema, como el control de electrodomésticos, gestión avanzada de la energía, y sistemas de riego automatizado.
4. **Inteligencia Artificial y Machine Learning:** Implementar técnicas de inteligencia artificial y aprendizaje automático para predecir y automatizar comportamientos del sistema basado en las preferencias y rutinas del usuario.
5. **Seguridad Avanzada:** Desarrollar e implementar medidas de seguridad avanzadas para proteger el sistema contra posibles ciberataques, garantizando la privacidad y la integridad de los datos del usuario.
6. **Interfaz de Usuario Mejorada:** Investigar nuevas tecnologías y metodologías de diseño para mejorar la usabilidad y accesibilidad de la interfaz de usuario, asegurando una experiencia más intuitiva y amigable.

7.BIBLIOGRAFÍA

Autor	Fecha	Título	Fuente
Programar Fácil	2024	Servidor web con ESP32	ENLACE
Open AI	2024	Chat GPT	ENLACE
Alejandro Alomar	2024	ESP32 desde Cero - Tutorial 7	ENLACE
Luis Llamas	2024	Cómo servir una página web con el ESP8266 o ESP32	ENLACE
W3Schools	2024	W3Schools	ENLACE
StackOverflow	2024	StackOverflow	ENLACE
PHP	2024	PHP	ENLACE
Víctor Colomo Gómez	2024	Documentación de Víctor Colomo Gómez 1ºDam	NO HAY
Maker Pro	2024	How to Use an LDR Sensor With Arduino	ENLACE
NayLamp	2024	Tutorial lcd con i2c, controla un lcd con solo dos pines	ENLACE
NayLamp	2024	Tutorial módulo lector rfid rc522	ENLACE
GeekFactory	2024	Botón o pulsador con Arduino: Todo lo que debes saber	ENLACE
ArgarCorp	2024	Ventana nueva con Javascript	ENLACE
MDN Web Docs	2024	Window.close()	ENLACE