

### Introducción

Con el avance de las tecnologías IoT, el número de hogares que implementan el Internet de las cosas no ha hecho sino aumentar exponencialmente, integrándose en los hogares y grandes sectores industriales, llegando a convertirse en un elemento indispensable para el funcionamiento de muchos ellos.



### Objeto

El objetivo de este TFG es el desarrollo software y hardware de un sistema IoT integrado a la vivienda convencional mediante enchufes, interruptores, conmutadores, control por voz y una aplicación móvil, así como el servidor central para la comunicación entre los distintos dispositivos. Se han aplicado una serie de requisitos como son la independencia de servicios de terceros, el desarrollo de un sistema modular que sea escalable, y el uso de herramientas de uso público y gratuito con el objetivo de ahorrar en gastos y ofrecer un producto final de bajo coste.

## Desarrollo del sistema IoT

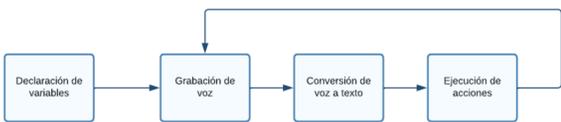
### Desarrollo del asistente de voz

#### Desarrollo hardware

El desarrollo hardware del asistente de voz no ha sido necesario al ir éste integrado directamente en el servidor local de la vivienda.

#### Desarrollo software

Debido a la alta adaptabilidad de un programa desarrollado en Python, así como el interés de desarrollar un sistema que no dependa de terceros, y para asegurar niveles de privacidad elevados, se ha optado por éste lenguaje.



### Esquema general

En el centro de las comunicaciones se ubica el *servidor central*. Éste será el encargado de gestionar a los distintos clientes y sus peticiones, además del control de la base de datos.

A su alrededor se encuentran los sistemas IoT periféricos. Estos están conformados por:

- Interruptor / Enchufe: Se programará un microcontrolador de la familia ESP. El desarrollo software se realizará en C++ en el entorno de Visual Studio Code.
- App Móvil: Se desarrollará una aplicación móvil que permita el control de todo el sistema.
- Asistente inteligente: Se desarrollará un asistente inteligente controlado por voz.

### Desarrollo del interruptor / enchufe

#### Desarrollo hardware

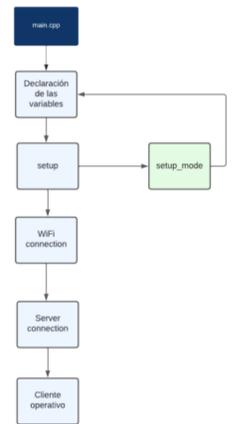
Para el conmutador se utilizará la familia de los ESP8266.

El ESP8266 es un SoC (system on chip) fabricado por la compañía china Espressif. Está conformado por un procesador de 32 bits y una shield WiFi con el protocolo TCP/IP integrado. Está orientado a aplicaciones IoT por lo que ofrece una conectividad WiFi completa.



#### Desarrollo software

El microcontrolador ha sido programado en C++ con Visual Studio Code y la extensión de PlatformIO. Está configurado con un protocolo de inicio de conexión, modo de funcionamiento normal y modo de configuración inicial, permitiendo introducir los parámetros necesarios como son el id y contraseña del servidor y el nombre y contraseña de la WiFi local para la conexión a red.



### Desarrollo de la aplicación móvil

#### Desarrollo hardware

La aplicación móvil se ha desarrollado de manera que cualquier tipo de dispositivo móvil pueda instalar y utilizar la aplicación desarrollada independientemente del sistema operativo.

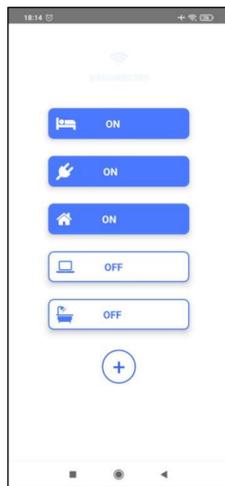
#### Desarrollo software

Debido a los problemas presentes en las aplicaciones nativas para correr en sistemas operativos diversos, se ha optó por desarrollar una aplicación multiplataforma. React Native es un framework de JavaScript creado por Facebook para escribir aplicaciones móviles orientado hacia iOS y Android.

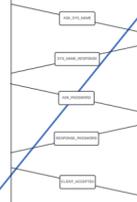
React Native es como React, pero utiliza componentes nativos en lugar de componentes web como bloques de construcción. React Native utiliza JSX, una sintaxis que permite escribir elementos dentro de JavaScript.

Los botones de la app se crean con JSX, elementos similares a HTML en diseño web.

Para la conexión sockets con el servidor, se utilizó la misma librería que en Python adaptada a JavaScript: socket.io, una biblioteca que permite la comunicación de baja latencia, bidireccional y basada en eventos entre un cliente y un servidor. Está construida sobre el protocolo WebSocket y proporciona garantías adicionales como el fallback a HTTP long-polling o la reconexión automática.



### Protocolo de conexión e identificación



### Desarrollo del servidor

#### Desarrollo hardware

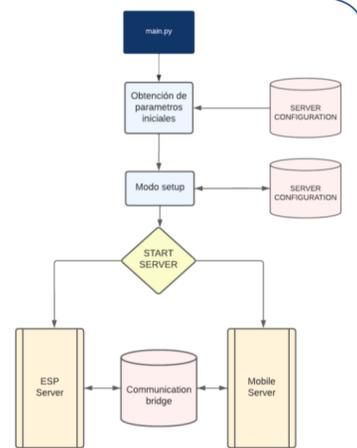
El servidor central se alojará en una Raspberry Pi 4, la última versión de un miniordenador de pequeñas dimensiones, bajo coste y consumo reducido. Este cuenta con una RAM duplicada de 8GB, conexiones Bluetooth 4.1 y Wifi 802.11n, además de dos puertos USB 3.0, dos USB 2.0, un puerto de un Gigabit Ethernet y dos entradas micro HDMI.



#### Desarrollo software

El servidor se ha desarrollado en Python, un lenguaje de programación versátil multiplataforma y multiparadigma que se destaca por su código legible y limpio. Está formado por tres grandes bloques, uno inicial y dos procesos paralelos:

1. Proceso inicial: Arranque del servidor. Permite la edición de parámetros del servidor como el ID, contraseña de acceso, puertos, etc...
2. Procesos paralelos: Una vez configurados los parámetros iniciales se inicializa el servidor con dos escuchas activas.
  1. Escucha de los dispositivos IoT del sistema
  2. Escucha de la app móvil.



## Resultados y conclusiones

- Enchufes / interruptores: Los enchufes e interruptores inteligentes cumplieron las expectativas previstas, funcionando en tiempo real, tanto como actuadores como sensores de estado.
- Servidor central: El servidor central ofrece el rendimiento adecuado para sus posibles aplicaciones, gestionando de manera eficiente las operaciones necesarias de cara a los clientes conectados a éste. Asimismo, al ser un servidor local ubicado en la propia red, preserva la privacidad del cliente frente a terceros, cumpliendo los requisitos especificados al inicio del proyecto.
- App móvil: La app móvil ha conseguido cumplir con sus funciones básicas de visualización y control de estado. Además, se ha diseñado de manera modular, dejando abierto su desarrollo a la introducción de nuevas aplicaciones como pueden ser alarmas, temporizadores, etc...
- Asistente de voz: De manera adicional, se ha creado un asistente de voz que permite una vía de control adicional además de la aplicación móvil. Sin embargo, ha sido necesario su apoyo en servicios externos para la traducción de audio a texto, incumpléndose el requisito de la privacidad frente a terceros.

En conclusión, a nivel global se han cumplido satisfactoriamente los objetivos propuestos: desarrollar un sistema IoT que ofrece el control de los sistemas de manera rápida, eficaz, y a bajo coste, permitiendo la integración del mismo en hogares y preservando la privacidad frente a terceros.