

## Resumen

En este documento se detalla el desarrollo, resultados y futuros pasos del trabajo práctico integrador de la materia **Taller de Sistemas Embebidos**. El proyecto elegido es un controlador de ventilador de techo y luces de línea (220 VAC) por medio de un control físico dedicado y una aplicación de celular conectada por *bluetooth*.

El sistema es una propuesta propia inspirada en otros productos que ya cumplen esta función. Por un lado, integra un control físico, indispensable para la experiencia del usuario debido a que sería absurdo que tenga que usar el celular cada vez que quiere prender una luz. Por otro, le da la libertad al usuario a controlar el celular de forma remota, para esas situaciones donde uno no puede o quiere acercarse a los controles de pared, como bien podría ser alguien acostado en la cama.

Otra parte importante es que el proyecto nos permite a los alumnos practicar con sistemas de control de potencia con tensión de línea, una habilidad demandada en la industria pero que no es parte del currículo universitario actual.

Ahora bien, la implementación de todo lo recién enunciado se realizó siguiendo los preceptos enseñados a lo largo de la materia, partiendo de diagramas de estados y luego construyendo un código limpio, entendible y funcional. La base de *hardware* es un núcleo F103RB junto con 2 placas de desarrollo propio: un *shield* con el control físico y una placa con el control de potencia.

## Índice General

### Índice

1	Introducción general	2
1.1	Análisis de necesidad y objetivos . . . . .	2
1.2	Módulos e Interfaces . . . . .	5
2	Introducción específica	5
2.1	Requisitos . . . . .	5
2.2	Casos de uso . . . . .	5
2.3	Descripción de los Módulos del sistema . . . . .	5
2.3.1	Alimentación . . . . .	5
2.3.2	Microcontrolador . . . . .	5
2.3.3	Motor . . . . .	5
2.3.4	Lector RFID . . . . .	5
2.3.5	Teclado Matricial . . . . .	5
2.3.6	Sensor Magnético . . . . .	5
2.3.7	Parlante . . . . .	5
2.3.8	Comunicación Wi-Fi . . . . .	5
3	Diseño e Implementación	5
3.1	Diseño del Hardware . . . . .	5
3.1.1	Diseño de la alimentación . . . . .	5
3.1.2	Diseño de los indicadores e interruptores . . . . .	5
3.1.3	Diseño del lector RFID . . . . .	5
3.1.4	Diseño del circuito para controlar el parlante . . . . .	5
3.1.5	Diseño del circuito para controlar el motor . . . . .	5
3.1.6	Diseño del circuito para controlar el teclado matricial . . . . .	5
3.1.7	Diseño del circuito para controlar el Módulo Wi-Fi . . . . .	5
3.1.8	Diseño del hardware con la placa NUCLEO-F429ZI . . . . .	5

3.2	Firmware del Smartlock . . . . .	5
3.2.1	Módulo Access Keys . . . . .	5
3.2.2	Módulo Keypad . . . . .	5
3.2.3	Módulo Motors . . . . .	5
3.2.4	Módulo MQTT . . . . .	5
3.2.5	Módulo RFID . . . . .	5
3.2.6	Módulo Speaker . . . . .	5
3.2.7	Módulo System . . . . .	5
3.2.8	Módulo UART Communications . . . . .	5
3.3	Firmware del ESP32 DEVKIT V1 . . . . .	5
3.4	Mosquitto broker . . . . .	5
3.5	Diseño de la aplicación y manejo de paquetes . . . . .	5
4	Ensayos y Resultados . . . . .	5
4.1	Pruebas funcionales de funcionamiento . . . . .	5
4.2	Cumplimiento de requisitos . . . . .	5
4.3	Comparación con otros sistemas similares . . . . .	5
4.4	Documentación del desarrollo realizado . . . . .	5
5	Conclusiones . . . . .	5
5.1	Resultados obtenidos . . . . .	5
5.2	Próximos pasos . . . . .	5

## Registro de versiones

Revisión	Cambios realizados	Fecha
1.0	Creación del documento	9/12/2025

## 1. Introducción general

Recordando del resumen, este proyecto implementa un módulo de control de ventilador y luces de línea (220 V) que permite operar:

- Desde la pared, por medio de un **\*\*botón físico\*\*** y un **\*\*potenciómetro\*\*** - Desde un dispositivo móvil mediante **Bluetooth (MIT App Inventor)**

Resultados esperados:

- Control de velocidad del ventilador. - Encendido/apagado de luces y ventilador. - Feedback mediante LEDs y buzzer para estados y alertas. - Permitir deshabilitar o configurar el BT mediante un DIP switch. - Guardado en memoria del último estado usado. - Funcionamiento seguro sobre cargas de 220 VAC.

### 1.1. Análisis de necesidad y objetivos

Un controlador de ventilador cubre la necesidad básica de controlar un ventilador y, opcionalmente, sus luces, valga la redundancia. Es decir, todos los controladores buscan satisfacer el mismo requisito, pero lo interesante es cómo encaran esto, y que canales de control le ofrecen al usuario.

En este proyecto se eligieron dos avenidas de control contrastantes, derivadas de haber estudiado otros dispositivos comercialmente disponibles. Estas son el control físico y el control por *bluetooth*.

El control físico, que estaría en la pared, es, de cierta manera, lo mínimo que espera el cliente con una instalación domiciliaria típica (ej.: un dormitorio). Cuando ingresa o egresa a una habitación tiene la oportunidad de ejercer control sobre el ventilador y sus luces, sin mayor esfuerzo que acercar la mano y girar una perilla, o tocar un botón. Es la mínima expresión de lo que entendemos como un control, y es sorprendente que ciertas alternativas comerciales no lo integren. Es evidente que tienen otra necesidad en mente, como que el control esté oculto por razones estéticas o de la propia instalación.

Este último punto nos lleva al segundo modo de control implementado: el *bluetooth*. Con el auge de el IOT y domótica, cada vez más gente quiere sentir el poder de controlar un dispositivo desde la palma de su mano, i.e. su celular. En el caso del ventilador, hemos identificado una verdadera necesidad de un control a distancia, que es el caso en el que el usuario no se quiere acercar al control físico. Esto sucede cuando: está sentado lejos o está en la cama descansando. Afortunadamente, la gente interesada en la domótica tiende a ser proficiente con el celular, y esta misma población generalmente lo lleva con su persona. Dada esta serie de postulados, es natural decidir integrar el control en el celular de forma inalámbrica.

Ahora bien, ¿Por qué *bluetooth* y no otro método? La realidad es que se eligió este por sobre WI-FI o un control RF dedicado debido a que existía un requerimiento del proyecto de integrar este método en la solución. En una solución real y no condicionada, se elegiría un *framework* armado por WI-FI que conecte con alguna aplicación de domótica (ej.: *Google Home*) o, para no alienar a aquellos menos tecnológicamente dispuestos, un control RF con los mismos controles que el físico.

En otros documentos se entra en detalle de puntos en contra del uso de WI-FI, que incluyen:

1. WI-FI presenta mayor complejidad en configuración, seguridad y mantenimiento de la conexión.
2. Requiere infraestructura adicional (router, red doméstica).
3. Demanda más recursos del microcontrolador.
4. Implica un ciclo de desarrollo más largo.
5. Un dispositivo mal diseñado podría volverse una vulnerabilidad en la red.

Estos son válidos pero completamente resolubles por un equipo preparado.

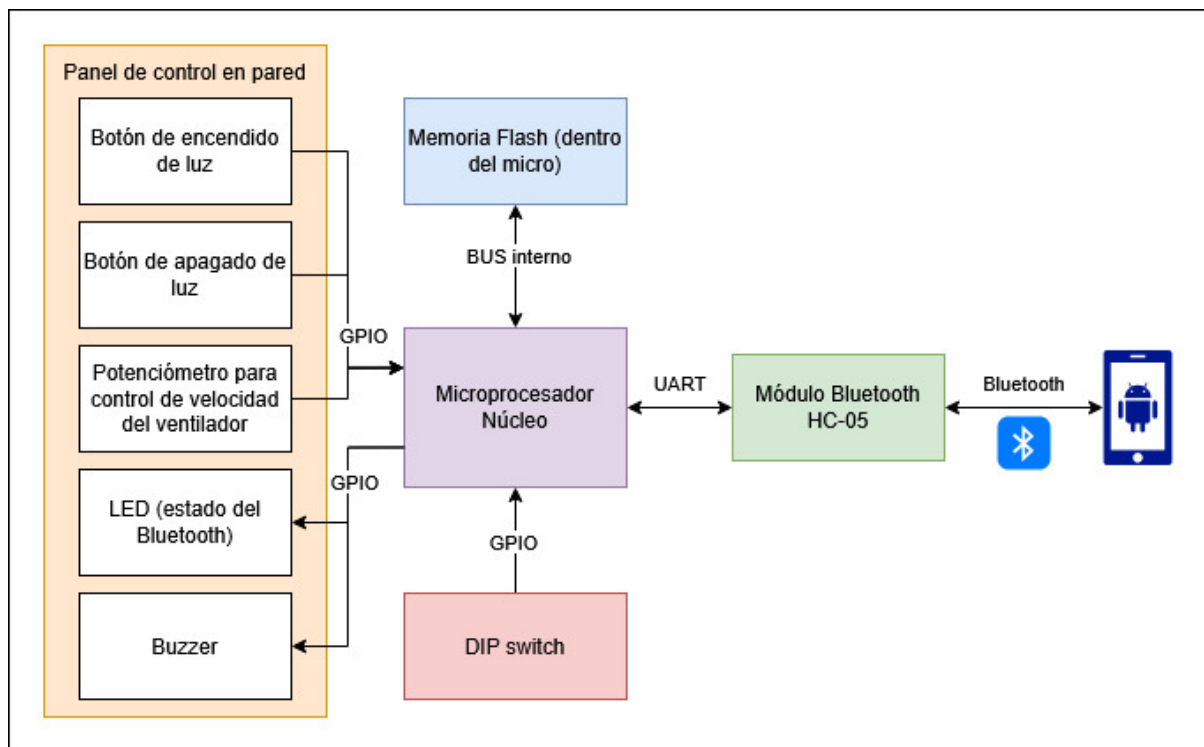


Figura 1: Diagrama en bloques de módulos e interfaces

## 1.2. Módulos e Interfaces

# 2. Introducción específica

## 2.1. Requisitos

## 2.2. Casos de uso

## 2.3. Descripción de los Módulos del sistema

### 2.3.1. Alimentación

### 2.3.2. Microcontrolador

### 2.3.3. Motor

### 2.3.4. Lector RFID

### 2.3.5. Teclado Matricial

### 2.3.6. Sensor Magnético

### 2.3.7. Parlante

### 2.3.8. Comunicación Wi-Fi

# 3. Diseño e Implementación

## 3.1. Diseño del Hardware

### 3.1.1. Diseño de la alimentación

### 3.1.2. Diseño de los indicadores e interruptores

### 3.1.3. Diseño del lector RFID

### 3.1.4. Diseño del circuito para controlar el parlante

### 3.1.5. Diseño del circuito para controlar el motor

### 3.1.6. Diseño del circuito para controlar el teclado matricial

### 3.1.7. Diseño del circuito para controlar el Módulo Wi-Fi

### 3.1.8. Diseño del hardware con la placa NUCLEO-F429ZI

## 3.2. Firmware del Smartlock

### 3.2.1. Módulo Access Keys

### 3.2.2. Módulo Keypad

### 3.2.3. Módulo Motors

### 3.2.4. Módulo MQTT

### 3.2.5. Módulo RFID

### 3.2.6. Módulo Speaker

### 3.2.7. Módulo System

### 3.2.8. Módulo UART Communications

## 3.3. Firmware del ESP32 DEVKIT V1

## 3.4. Mosquitto broker

## 3.5. Diseño de la aplicación y manejo de paquetes

# 4. Ensayos y Resultados