

Escuela Politécnica
Superior

Grupo (1 persona)
Nombre y apellidos:
1. Álvaro Sánchez



Proyecto
final:
Sistemas
Empotrados

ESP32-S3-WROOM1 (Arduino IDE)



Nombre y apellidos

Índice/Tabla de contenidos

<u>Índice/Tabla de contenidos</u>	2
<u>1. METODOLOGÍA Y FORMATO DE LA ACTIVIDAD</u>	3
1.1 Requisitos mínimos	3
1.2 Criterios de evaluación	4
<u>2. DESARROLLO DE LA ACTIVIDAD PRÁCTICA</u>	5
2.1 Fundamento	5
2.2 Diseño del sistema	5
2.3 Librerías utilizadas (obligatorio indicar nombre y versión)	6
2.4 Montaje (hardware) y esquema	5
2.6 Pruebas y validación	5
2.7 Métricas (opcional para bonus)	6
2.8 Resultados y discusión	6
2.9 Trabajo futuro	6
<u>3. Entregables</u>	6
<u>4. Presentación y defensa</u>	6
<u>APENDICE</u>	7
Ideas orientativas:	7

1. METODOLOGÍA Y FORMATO DE LA ACTIVIDAD

Trabajo por parejas. El desarrollo se realizará con Arduino IDE. Si se emplean librerías externas, es obligatorio indicarlas con nombre y versión en la sección correspondiente.

1.1 Requisitos mínimos

1. Captura de datos reales: mediante sensores físicos (p. ej., temperatura, luz, movimiento, sonido, IMU, etc.) o datos telemáticos (p. ej., MQTT/HTTP/BLE, tramas externas, beacons, etc.).
2. Procesado local: filtrado, fusión de sensores, umbrales, lógica de negocio, FSM, estadísticas básicas, etc. (no vale reenviar crudo sin tratamiento).
3. Respuesta en el mundo real: actuadores (LEDs, relés, servos, drivers, DAC, etc.) o respuesta telemática (publicar a MQTT/HTTP, BLE Notify, etc.).

Se acepta todo tipo de hardware externo. Si fuera costoso o difícil de conseguir, pueden simular parte del entorno siempre que justifiquen el criterio y entreguen el código y las herramientas de simulación.

1.2 Criterios de evaluación

Funcionalidad y requisitos (3,0):

- Ciclo completo captación → procesado local → respuesta.
- Casos de uso operativos y verificables (actuadores o telemáticos). Latencia razonable para el contexto.

Calidad técnica y diseño (2,0):

- Arquitectura modular (drivers, lógica, comunicación)
- Temporización sin bloqueos (FSM/millis()) Gestión correcta de periféricos.
- ISRs breves
- Sincronización adecuada
- Light/deep sleep/modem-sleep correctamente configurado.

Interrupciones y/o bajo consumo se valoran si aportan valor al caso, pero **no se penaliza** no usarlas **si se justifica** su no necesidad.

Robustez y pruebas (1,5):

- Casos de prueba representativos
- Manejo de errores
- Logs/telemetría para diagnóstico.
- Mediciones simples (ej., latencia evento→acción o consumo)

Documentación y repositorio (1,5): README.md

- claro con instrucciones
- Librerías con nombre y versión
- Diagrama de bloques Esquemas/fotos.
- Estructura del proyecto limpia y reproducible.

Gestión del proyecto (1,0):

- Uso de Git
- Commits significativos
- Ramas cuando corresponda Etiquetado de la versión defendida.

Presentación y defensa (1,0):

- Claridad expositiva
- Demo real
- Defensa individual de cada integrante.

2. DESARROLLO DE LA ACTIVIDAD PRÁCTICA

2.1 Fundamento

Describid el propósito del sistema, el problema que resuelve y el contexto de uso. Incluid un diagrama de bloques que muestre entradas (sensores o datos telemáticos), procesado local y salidas (actuadores o telemetría).

El sistema desarrollado tiene como objetivo implementar un **mecanismo básico de control físico**, donde un microcontrolador ESP32-S3 gestiona el encendido y apagado de un LED externo mediante dos pulsadores. Este tipo de lógica es habitual en sistemas empotrados sencillos como mandos, paneles de control o prototipos de interacción humana.

El proyecto cumple el ciclo completo de un sistema empotrado:

- **Captura de datos:** los botones funcionan como entradas digitales.
- **Procesado local:** el ESP32 interpreta el estado de cada pulsador y decide la acción.
- **Actuación:** un LED externo responde encendiéndose o apagándose.

Adicionalmente, se intentó integrar un sensor DHT11 para capturar temperatura y humedad. Sin embargo, durante las pruebas se detectaron lecturas inestables y pérdida de comunicación, lo que sugiere un fallo de hardware en el módulo. Aun así, se documenta como parte del diseño original del sistema.

2.2 Diseño del sistema

Compleudad la siguiente tabla con los elementos principales.

Elemento	Rol	Interfaz/Pin/Bus	Notas
Botón ON	Entrada	GPIO4	Activa el LED cuando se pulsa. Pull-down interno.
Botón OFF	Entrada	GPIO5	Apaga el LED cuando se pulsa. Pull-down interno.
DHT11 (fallido)	Entrada	GPIO15	Se intentó usar para capturar temperatura, pero no funcionó por error del módulo.

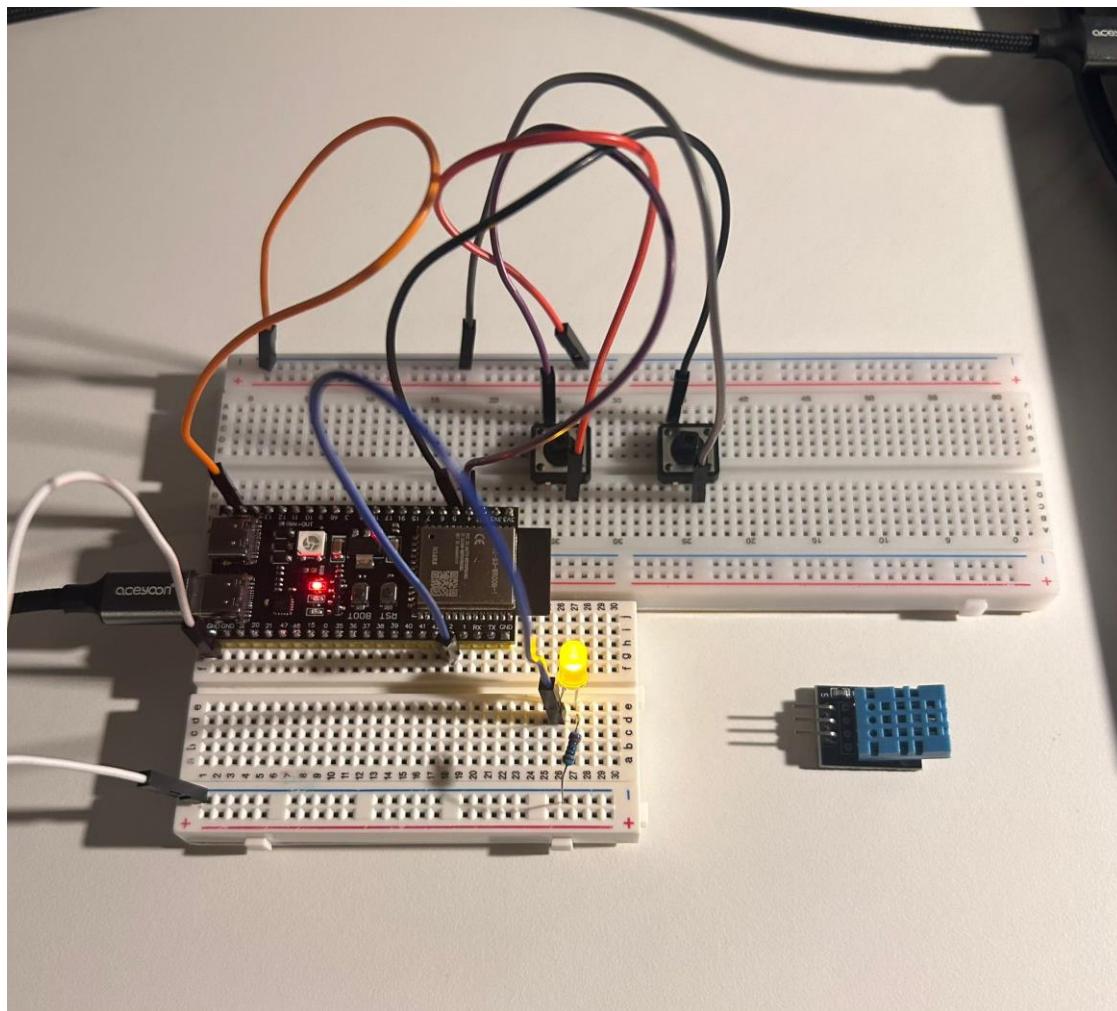
LED Externo	Actuador	GPIO 2	LED con resistencia serie (220 Ω) para protección.
Alimentación	Fuente	USB 5V → Regulador interno 3.3V	La placa trabaja a 3.3 V pero se alimenta desde USB.

2.3 Librerías utilizadas (obligatorio indicar nombre y versión)

Librería	Versión	Origen/URL	Uso
ESP32 Arduino Core	2.0.14	https://github.com/espressif/arduino-esp32	Soporte de GPIO, digitalRead, digitalWrite.
DHTesp (intentada)	1.0.0	https://github.com/beegee-tokyo/DHTesp	Lectura del DHT11 (no funcional por fallo del sensor)

2.4 Montaje (hardware) y esquema

Pegad aquí el esquema eléctrico (KiCad/Fritzing/Wokwi) o fotos nítidas del cableado y prototipo. Describid consideraciones de alimentación, protección y seguridad eléctrica.



Apuntes: No consigo implementar el DHT11. Posible error de fabricación

Lista de componentes:

- ESP32-S3 Dev Module
- 2 botones
- 1 LED rojo/amarillo
- 1 resistencia de $220\ \Omega$
- Cables Dupont
- Protoboard
- Cable USB

Conexiones

Botón ON (GPIO4):

- Un lado del botón → GPIO4
- Otro lado → GND
- Se usa INPUT_PULLDOWN, por lo que no requiere resistencia externa

Botón OFF (GPIO5):

- Un lado → GPIO5
- Otro lado → GND
- Igual: INPUT_PULLDOWN

LED:

- Ánodo → GPIO2
- Cátodo → Resistencia 220 Ω → GND

DHT11 (fallido):

- VCC → 3.3V
- GND → GND
- DATA → GPIO15

(El sensor no ofrecía datos válidos, posiblemente por un defecto de fábrica.)

2.5 Pruebas y validación

Definid casos de prueba que cubran las rutas principales del sistema. Registrad resultados y observaciones.

ID	Caso de prueba	Esperado	Resultado
T1	Pulsar botón ON	LED se enciende	Correcto
T2	Pulsar botón OFF	LED se apaga	Correcto
T3	Mantener ON pulsado	LED permanece encendido	Correcto
T4	Mantener OFF pulsado	LED permanece apagado	Correcto
T5	Lectura del DHT11	Datos estables de temperatura	Lecturas fallidas. Se descarta del prototipo.

2.6 Métricas (opcional para bonus)

Si optáis al bonus, medid o estimad consumo en reposo/actividad y/o latencia cuando usáis interrupciones frente a polling. Indicad herramienta de medición (USB meter, multímetro, logs de tiempo, etc.).

Métrica	Método	Valor	Comentarios
Consumo LED encendido	Estimación	20–30 mA	Típico ESP32 con un LED externo
Latencia botón → LED	millis()	< 5 ms	Respuesta prácticamente instantánea

2.7 Resultados y discusión

Resumid los principales resultados, limitaciones encontradas y justificaciones de diseño (por qué interrupciones/bajo consumo o por qué no: compromisos).

El sistema implementado cumple correctamente la funcionalidad principal: el usuario puede encender y apagar un LED mediante dos botones físicos. La respuesta es inmediata y el circuito es estable.

La principal limitación encontrada fue el mal funcionamiento del sensor DHT11. Se realizaron varias pruebas de alimentación y cambio de GPIO sin éxito, por lo que se concluye que el módulo estaba defectuoso.

El diseño es sencillo pero representativo del ciclo básico de un sistema empotrado.

2.8 Trabajo futuro

Indicad mejoras previstas, riesgos mitigados parcialmente y siguientes pasos realistas.

- Sustituir el DHT11 defectuoso por uno funcional.
- Añadir un indicador sonoro usando un buzzer.
- Añadir comunicación BLE para encender/apagar desde el móvil.
- Integrar protección física y una caja para el montaje.

1. Entregables

- 1) PDF único con este guion cumplimentado, esquemas/fotos y enlaces.
- 2) Enlace al repositorio con código Arduino, diagramas y documentación.
- 3) Demo funcional en la defensa (ver sección Presentación).
- 4) Vídeo de 1–2 min con demo funcional.

2. Presentación y defensa

Cuándo: última semana de clase (orientativo: martes 2 de diciembre; la fecha exacta la confirmará la persona que sustituye al profesor).

Formato: 8–10 min de presentación + 5 min de preguntas. La presentación NO es suficiente por sí sola: debe haber defensa oral y demo.

Defensa individual: se realizarán preguntas dirigidas a cada integrante sobre partes distintas del proyecto (código, hardware, diseño, pruebas...).

Entrega previa: enlazar el repositorio y subir el vídeo 48 h antes de la defensa.

APÉNDICE

Ideas orientativas:

Por supuesto, si tenéis otra idea exponedla, pero si estáis cortos de ideas:

- Estación ambiental con detección de eventos (ruido, CO₂, luz) y actuación (ventilación/avisos).
- Sistema de aforo con IR/ToF e iluminación adaptativa.
- Cerradura o control de acceso con BLE/NFC y registro telemático.
- Monitor de energía simple con relé inteligente y políticas de ahorro.
- Telemetría de un juego físico (dianas, sensores) con marcador local MQTT/HTTP.

- OCR de matrícula/código y apertura de puerta/servo (Puerta de aparcamiento o cinta de paquetería)