

## Trabajo Práctico: IOT DOG - Sistema de Monitoreo Ambiental

**Objetivo General:** Desarrollar un prototipo funcional montado en un case desarrollado en pp que mida y reporte variables ambientales (temperatura, humedad, luz, gases), alerte sobre condiciones anómalas y permita la configuración de sus parámetros a través de un menú físico.

### Resumen de Funcionalidades:

- Leer los diferentes sensores junto a la hora de la muestra.
  - Evaluar si los sensores superan el umbral y alertar.
  - Enviar por Telegram y MQTT la información.
  - Guardar en la EEPROM los valores de configuración.
  - Mostrar la información en la pantalla LCD.
- 

### 1. Requisitos Generales

Requisito	Descripción
<b>Grupos</b>	Equipos de 3 a 4 integrantes. La conformación es libre, pero debe estar definida al final de la primera clase. Si no se asignará aleatoriamente o se creará un grupo con dicha persona q no tiene un grupo.
<b>Repositorio Git</b>	Crear un repositorio para el proyecto. El link debe ser entregado en el campus virtual.
<b>Identificación</b>	La primera línea de cada archivo de código (.ino) debe contener los apellidos de los integrantes.
<b>Gestión de Proyecto</b>	Crear y mantener un cronograma de tareas con roles y responsables. Habrá un seguimiento semanal con calificación individual y grupal.

---

### 2. Módulos del Proyecto y Requisitos Técnicos

#### 2.1 Programación (Software)

- **Desarrollo Modular:** Realizar códigos individuales para probar cada sensor y actuador antes de la integración.
- **Máquina de Estados:** El código final debe estar estructurado como una máquina de estados. Se debe entregar el diagrama de estados correspondiente.
- **Menú de Configuración:** Implementar un menú accesible por botones para configurar:
  - Offset GMT para la hora.
  - Valores umbral de los sensores para las alertas.

- Intervalo de envío de datos por MQTT.
- **Memoria No Volátil:** Almacenar los valores de configuración en la memoria EEPROM del ESP32.
- **Visualización Local:** Mostrar lecturas y estados en la pantalla LCD.
- **Indicadores Visuales:** Usar LEDs para indicar el estado del ambiente (Normal, Alerta, Peligro).

## 2.2 Diseño de PCB (Hardware)

Característica	Requisito
<b>Dimensiones</b>	Tamaño máximo de 10x10 cm.
<b>Alimentación</b>	1. Conector USB-B para 5V. 2. Bornera para 12V (incluir diodo de protección de polaridad inversa).
<b>Análisis Requerido</b>	Justificar por escrito la necesidad (o no) de un regulador de tensión de 5V a 3.3V. En caso de necesitarlo incluir regulador con su circuito
<b>Montaje ESP32</b>	Utilizar una tira de pines hembra. <b>No soldar el ESP32 directamente a la PCB.</b>
<b>Conectores</b>	Usar conectores Molex o IDC para todos los sensores, LEDs y botones. No se permiten cables de protoboard soldados.
<b>Salidas de Tensión</b>	Incluir una bornera con salidas de 3.3V y GND.
<b>Circuito de Actuador</b>	Implementar un circuito con transistor para controlar el relé. La salida del relé debe ser a una bornera.
<b>Puntos de Prueba</b>	Incluir <b>test points</b> para mediciones clave (VCC, GND, salidas analógicas de sensores).
<b>Fijación</b>	El diseño debe incluir perforaciones para sujetar la PCB al <b>case</b> con tornillos.

## 2.3 Conectividad (IoT)

- **Configuración Wi-Fi:** Crear una página web hosteada en el ESP32 (usando **WiFiManager** o similar) que permita cambiar las credenciales de Wi-Fi sin reprogramar el microcontrolador.
- **Alertas Remotas:** Enviar alertas a un chat de Telegram cuando un sensor supere su umbral configurado. También se puede preguntar el estado por medio de un msj.

- **Telemetría MQTT:** Publicar los datos de los sensores en un broker MQTT. El formato del mensaje (payload) debe ser **JSON**, incluyendo el valor y la fecha/hora de la muestra.

## 2.4 Diseño Mecánico (Case)

- **Integración:** El **case** debe alojar la PCB, sensores y pantalla de forma segura, con tornillos.
- **Accesibilidad:** Garantizar el acceso al conector micro-USB del ESP32 y a los conectores de los periféricos.
- **Mantenimiento:** El **case** debe poder abrirse para acceder a los componentes internos.

## 3. Cronograma de Entregas

El incumplimiento en las fechas de entrega impactará en la calificación final.

Entrega	Contenido a Entregar	Fecha Límite (Semana de)
<b>1: Pruebas Iniciales</b>	Códigos de cada sensor/actuador funcionando por separado (subidos a Git).	[ dd/mm/aaaa ]
<b>2: Diseño Electrónico</b>	Capturas del esquemático y layout de la PCB (Gerbers opcional).	[ dd/mm/aaaa ]
<b>3: Integración</b>	Video mostrando el prototipo ensamblado (PCB + Case) con todos los componentes funcionando en conjunto.	[ dd/mm/aaaa ]
<b>4: Proyecto Final</b>	Código final en Git, video de la presentación y demostración de la comunicación IoT (MQTT y Telegram).	[ dd/mm/aaaa ]

## 4. Evaluación y Calificación

- **Colaboración:** El trabajo en equipo es mandatorio. La falta de colaboración resultará en una nota grupal **inferior a 6**.
- **Nota Máxima:**
  - **7 (siete):** Cumpliendo con todos los requisitos del TP y las entregas.
  - **7 a 10 :** Presentando el proyecto funcional en la "**Expo Mecatrónica**". Se verificará en los días del evento que algún integrante del grupo está presentando.
- **Presentación Final:** Exposición oral o video (5-10 minutos) explicando el proceso de diseño y construcción.

---

## 5. Lista de Materiales

Componente	Cantidad
ESP32 DevKitC	1
PCB Virgen 10x10cm	1
Sensor de Temperatura y Presión (BMP280)	1
Fotorresistencia (LDR)	1
Sensor de Gas (MQ-2 o similar)	2
Pantalla LCD I2C 16x2	1
LEDs 5mm (diferentes colores)	3
Pulsadores para menú 6x6 tth	5
Relé 5V, Transistor	1 de c/u
Conectores (Molex, IDC, Tiras de pines hembra)	Varios
Conector USB-B hembra para PCB	1
Borneras de 2 pines	3

---

## 6. Recursos Útiles

- **Control de Relé con Transistor:**  
<https://www.inventable.eu/controlar-rele-con-transistor/>
- **Solución de problemas con BME280:**  
<https://randomnerdtutorials.com/solved-could-not-find-a-valid-bme280-sensor/>