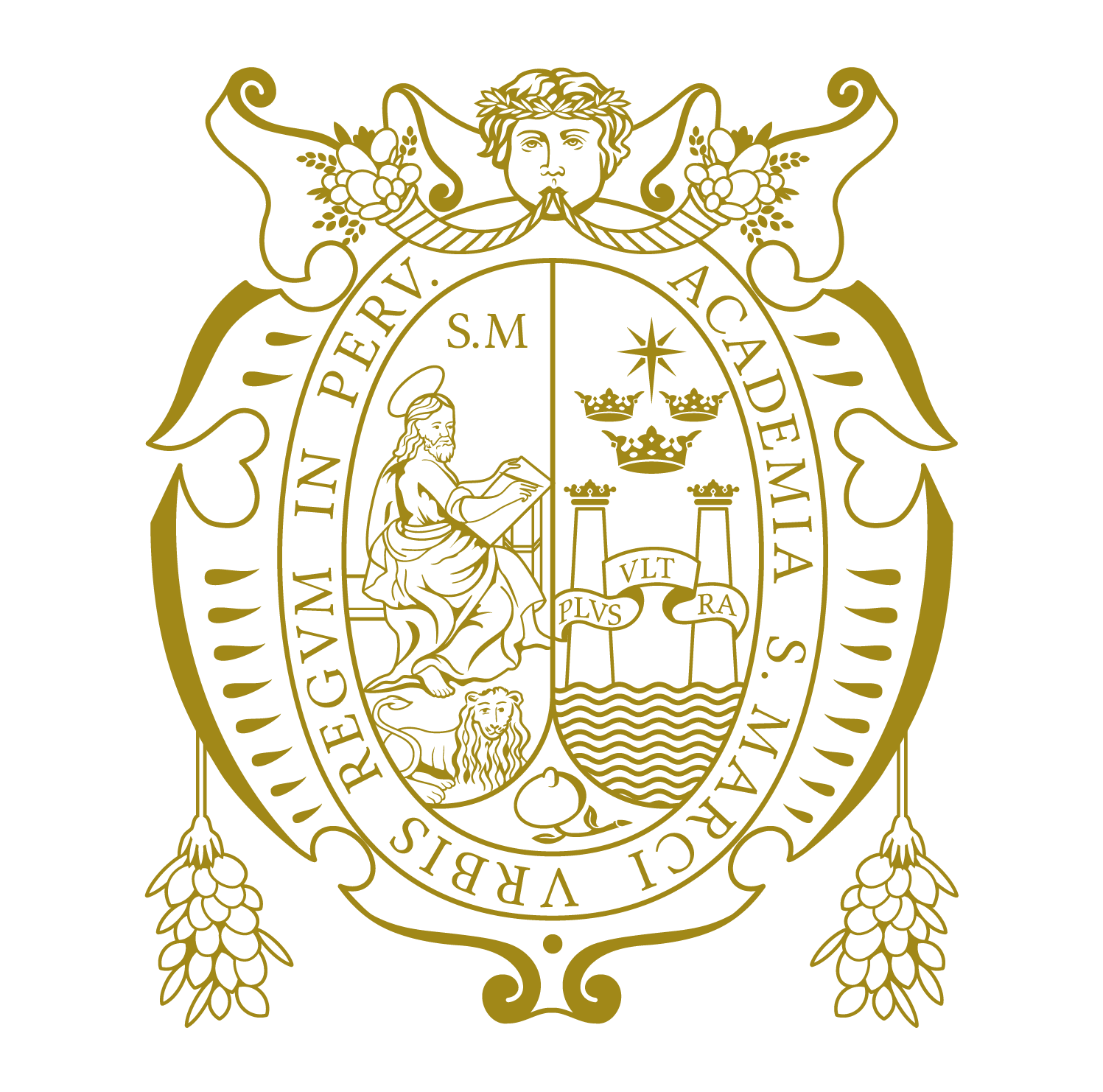
“Año de la recuperación y consolidación de la economía peruana”

**Universidad Nacional Mayor de San Marcos**

Universidad del Perú. Decana de América

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Software**



**Arquitectura del Proyecto**

**Internet de las Cosas**

**Grupo 2:**

Leonardo Ormeño Vasquez

Sergio Daniel Quiroz Ardiles

David Brian Sandoval Falcon

Ivan Tarazona Villar

Josue Montes Perez

**Docente:**

Yessica Rosas Cueva

**LIMA, PERÚ**

**2025**

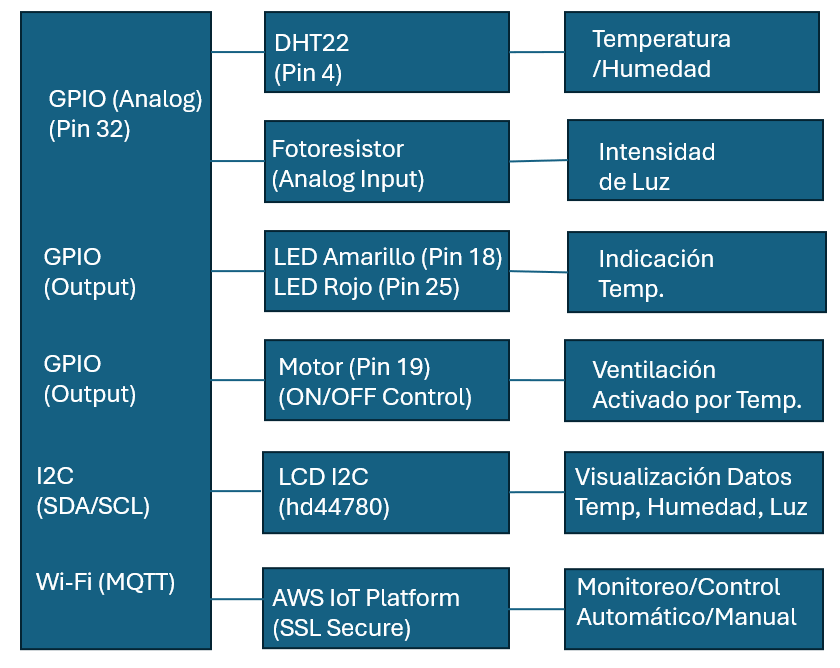
# Esquema de Arquitectura General

Este esquema conceptual describe cómo los componentes interactúan en el sistema:

* **Sensores (Entradas):**
  + **DHT22:** Monitorea temperatura y humedad del aire.
  + **LDR:** Detecta niveles de luz en el invernadero.
* **Microcontrolador:**
  + **ESP32:** Controlador principal
* **Actuadores (Salidas):**
  + **Motor DC:** Activa la ventilación
  + **2 leds**
  + **Pantalla LCD:** Muestra la lectura de la temperatura y humedad medidas por el DHT22.
* **Plataforma IoT:**

1. Diagrama Eléctrico Simplificado

Este diagrama muestra cómo se conectarán los sensores y actuadores al ESP32.



## Notas técnicas:

* **DHT22**: Se conecta al ESP32 mediante un pin digital (DHT\_PIN = 4).
* **Fotoresistor**: Se conecta a un pin analógico (PHOTOSENSOR\_PIN = 32).
* **LEDs**: Se utilizan dos LEDs (amarillo y rojo) en los pines 18 y 25 para indicar la temperatura.
* **Motor**: Controlado desde el pin 19, activado manualmente o automáticamente según la temperatura.
* **LCD I2C**: Muestra la temperatura, humedad e intensidad de luz en pantalla.
* **Wi-Fi MQTT**: Se usa para enviar datos a AWS IoT y recibir comandos de control remoto.
* **Modos de control**:
  + **Automático**: Basado en temperatura, enciende o apaga LEDs y motor.
  + **Manual**: AWS IoT puede encender/apagar LEDs y motor a través de MQTT.

1. Flujo Lógico del Software:

**1. Inicialización de Librerías y Definición de Variables**

* Se incluyen las librerías necesarias para manejar WiFi, comunicación segura con AWS IoT, sensores (DHT22), LCD y MQTT.
* Se definen los tópicos MQTT para publicar y suscribirse a mensajes de AWS IoT.
* Se configuran los pines del **ESP32** para sensores y actuadores.
* Se inicializan variables para controlar **luces LED, motor y modo de operación** (automático o manual).
* Se crea una instancia de:
  + **DHT22** (sensor de temperatura y humedad).
  + **LCD 16x2**.
  + **WiFiClientSecure** para comunicación con AWS IoT.
  + **PubSubClient** para gestionar MQTT.

**2. Conexión a AWS IoT**

* Se configura el **modo WiFi** y se conecta a la red WiFi.
* Se establecen **certificados de seguridad** para comunicación con AWS IoT.
* Se configura el **broker MQTT** en AWS IoT.
* Se suscribe a los tópicos:
  + iotfrontier/sub → Modo de operación.
  + iotfrontier/sub-motor → Control del motor.
  + iotfrontier/sub-led → Control de los LEDs.
  + iotfrontier/sub-config → Configuración de parámetros.

**3. Recepción de Mensajes desde AWS IoT**

* Cuando llega un mensaje MQTT:
  + Se identifica el tópico en el que se recibió.
  + Se procesa el mensaje en formato JSON.
  + Dependiendo del tópico, se ejecuta una acción:
    - iotfrontier/sub → Cambia entre **modo automático/manual**.
    - iotfrontier/sub-config → Actualiza **umbrales** para activar luces/motor.
    - iotfrontier/sub-motor → Enciende/apaga el motor en modo **manual**.
    - iotfrontier/sub-led → Enciende/apaga los LEDs en modo **manual**.

**4. Configuración de Sensores y Actuadores**

* Se configuran los **pines** de los LEDs y el motor como salida.
* Se inicializa la pantalla **LCD 16x2** y muestra un mensaje de "Iniciando...".

**5. Ejecución en el Bucle Principal (loop)**

Cada 3 segundos, el ESP32 realiza las siguientes acciones:

1. **Lee sensores**:
   * Temperatura (DHT22).
   * Humedad (DHT22).
   * Iluminación (Photosensor).
2. **Muestra los valores en la pantalla LCD**:
   * Línea 1: Temperatura y humedad.
   * Línea 2: Nivel de iluminación.
3. **Control de LEDs**:
   * **Modo Automático**:
     + Si la temperatura es **mayor o igual al umbral del LED rojo**, enciende el **LED rojo**.
     + Si la temperatura es **mayor o igual al umbral del LED amarillo**, enciende el **LED amarillo**.
     + Si la temperatura es baja, apaga ambos LEDs.
   * **Modo Manual**:
     + Controla los LEDs según los valores recibidos desde AWS IoT.
4. **Control del Motor**:
   * **Modo Automático**:
     + Si la temperatura supera el umbral del motor, se enciende.
   * **Modo Manual**:
     + El motor se enciende/apaga según los comandos de AWS IoT.
5. **Publicación de Datos en AWS IoT**:
   * Se envían los valores de temperatura, humedad e iluminación en formato JSON.
6. **Mantiene Conexión con AWS IoT**:
   * Llama a client.loop() para gestionar la conexión MQTT.
7. **Espera 3 segundos y repite el proceso**.