

ARQUITECTURA DE SISTEMA

1. Capa de Dispositivos

Esta es la capa física que interactúa directamente con la persona. Su única función es recolectar los datos biométricos.

- **Sensores:**
 - **ECG (Electrocardiograma):** Mide el pulso cardíaco.
 - **DS18B20:** Mide la temperatura corporal superficial.
- **Microcontrolador (El Cerebro del Dispositivo):**
 - **El ESP32:** se encargará de leer los datos de los sensores varias veces por segundo.

2. Capa de Red

Una vez que el ESP32 recolecta los datos, necesita una forma segura y eficiente de enviarlos a internet para su procesamiento.

- **Conectividad:** El módulo **Wi-Fi** del ESP32 se conectará a una red local.
- **Protocolo de Comunicación:** El protocolo ideal para IoT es **MQTT (Message Queuing Telemetry Transport)**. Es extremadamente ligero y eficiente, perfecto para enviar pequeños paquetes de datos de sensores.

3. Capa de Procesamiento

Los datos que llegan desde el dispositivo no tienen mucho valor hasta que se procesan, almacenan y analizan.

- **Broker MQTT:** Es el servidor intermediario que recibe los mensajes del ESP32 y los dirige a donde necesiten ir.
- **Base de Datos:** Una vez que el broker recibe los datos, una regla los enviará a una base de datos para su almacenamiento. Una base de datos de MYSQL
- **Lógica de Análisis (El Algoritmo):**
 1. Recupera los datos del sensor.
 2. Calcula la **línea base** de la persona (calibración inicial).
 3. Compara las nuevas lecturas con la línea base para detectar desviaciones significativas.
 4. Genera un "índice de estrés" o una alerta.
 5. Almacena el resultado del análisis de vuelta en la base de datos.

4. Capa de Aplicación

Esta es la capa final con la que interactúa el usuario. Su propósito es presentar los datos biométricos y los resultados del análisis de una manera clara y visualmente atractiva. Usaremos un flujo de trabajo que va del diseño a la implementación.

Paso 1: Diseño y Prototipado con Figma

Antes de escribir una sola línea de código, se diseña la experiencia de usuario. Figma es la herramienta perfecta para este propósito.

- **Función de Figma:**

- **Diseño de la Interfaz (UI):** Aquí se definirán colores, tipografías, y la disposición de los elementos como los gráficos, los indicadores de estado, y los valores numéricos del pulso, temperatura.
- **Prototipado (UX):** Puedes crear un prototipo interactivo para simular cómo funcionará el dashboard. Por ejemplo, puedes diseñar cómo se verá la pantalla durante la fase de "calibración" y cómo cambiará a la vista de "monitoreo en vivo". Esto te permite probar la usabilidad antes de la programación.

Paso 2: Desarrollo del Backend con Python y Flask

Flask es un micro-framework de Python ideal para crear el "motor" de tu aplicación web. Este backend no será visible para el usuario, pero es el que hace todo el trabajo pesado.

- **Función de Flask:**

- **Crear una API:** El backend de Flask expondrá una API que el frontend podrá consultar para obtener los datos. Por ejemplo, podría tener un endpoint como `/api/live_data`.
- **Conexión a la Base de Datos:** Tu aplicación Flask se conectará a la base de datos
- **Procesar Peticiones:** Cuando el frontend pida datos, la aplicación Flask consultará la base de datos, formateará la información y se la enviará de vuelta al dashboard.
- **Servir la Aplicación:** Flask también se encargará de servir los archivos HTML, CSS y JavaScript que componen la interfaz visible del dashboard.