

# SISTEMA DE MONITOREO DE SALUD

**Integrantes:** Alejandro Juncal, Christian Ojeda, Tenoch Moreno

## **Descripción:**

El sistema de monitoreo de la salud personal permite a los usuarios registrar, visualizar y analizar métricas de salud (peso, presión arterial, nivel de glucosa, temperatura corporal), proporcionando un seguimiento detallado de su evolución. Además, genera gráficos interactivos, permite configurar alertas personalizadas, visualizar un historial de mediciones, y exportar datos para consultas médicas.

## **Objetivos:**

1. Permitir a los usuarios registrar sus métricas de salud.
2. Visualizar y analizar las tendencias de salud a través de gráficos interactivos.
3. Configurar alertas personalizadas para notificar cambios fuera de los rangos normales.
4. Ofrecer un historial detallado de mediciones para su revisión y comparación.
5. Facilitar la exportación de los datos en formatos CSV o PDF.
6. Proveer una plataforma segura y confiable para la protección de los datos de los usuarios.

## **Requerimientos funcionales:**

1. Los usuarios pueden registrar manualmente sus datos de salud, incluyendo la fecha y hora de cada medición.
2. El sistema genera gráficos interactivos con las métricas de salud.
3. Los usuarios pueden configurar alertas personalizadas para ciertos valores fuera de rango.

4. Los usuarios pueden acceder a su historial de mediciones y compararlo con valores recomendados.
5. El sistema permite la exportación de los datos a formatos CSV o PDF.
6. El sistema permite la autenticación de usuarios y la gestión de cuentas personales.
7. Los usuarios pueden establecer objetivos personalizados relacionados con sus métricas de salud.
8. Los usuarios pueden compartir sus datos con su médico mediante reportes digitales.

### **Requerimientos no funcionales:**

1. El sistema debe ser seguro y garantizar la privacidad de los datos.
2. Implementación de HTTPS para la comunicación segura entre el cliente y el servidor.
3. Aplicación de políticas de encriptación para proteger los datos almacenados.
4. La base de datos debe ser eficiente en el manejo de registros de salud (uso de PostgreSQL o SQL Server).
5. El sistema debe permitir un acceso rápido y eficiente a los datos del usuario.

### **Principales casos de uso:**

- **Análisis y seguimiento del estado de salud**  
Los usuarios pueden registrar y analizar métricas como peso, presión arterial, glucosa y temperatura corporal. El sistema detecta patrones, resalta valores fuera de rango y permite establecer objetivos personalizados. Se generan alertas en caso de valores críticos y, si hay pocos datos, se sugiere registrar más mediciones para un análisis preciso.
- **Exportación de datos para consultas médicas**  
Los usuarios pueden exportar su historial de salud en formatos CSV o PDF para compartirlo con médicos. Se elige el rango de fechas y el sistema genera automáticamente el archivo. Si faltan registros, se sugiere completar más mediciones antes de exportar.

- **Visualización de datos y gráficos**  
Un panel interactivo permite analizar métricas de salud a través de gráficos dinámicos con líneas de tendencia y promedios. Los usuarios pueden filtrar datos por fecha o tipo de medición y comparar con valores de referencia. También pueden guardar o imprimir los gráficos para un mejor seguimiento.
- **Configuración de alertas personalizadas**  
Los usuarios pueden programar alertas para recibir notificaciones cuando una medición esté fuera de los valores normales o cuando sea momento de realizar un nuevo registro. Las alertas pueden recibirse por correo electrónico o dentro de la plataforma, ayudando a un monitoreo más preciso y constante.

## Tecnologías Sugeridas:

### Backend

- **Django con Python (Alternativa: Node.js con Express.js)**
  - Framework robusto y seguro para la gestión de usuarios y datos.
  - Buen soporte para autenticación y ORM para bases de datos.

### Frontend

- **React.js (Alternativa: Vue.js)**
  - Framework moderno para crear interfaces dinámicas e interactivas.
  - Facilita la visualización de datos con librerías gráficas.

### Base de Datos

- PostgreSQL (Alternativa: SQL Server)
- Base de datos relacional eficiente y escalable para almacenar registros médicos.
- Soporte para consultas avanzadas y transacciones seguras.

### Visualización de Datos

- **Chart.js o D3.js**
  - Librerías para gráficos interactivos y visualización de tendencias.

### Seguridad

- **Autenticación con JWT o OAuth 2.0**
  - Para gestionar sesiones y permisos de acceso.
- **HTTPS**
  - Para encriptar la comunicación entre cliente y servidor.

#### **Exportación de Datos**

- Librerías para generación de PDF (ReportLab en Python o pdfkit en JavaScript).
- Módulos de exportación CSV nativos en Python y JavaScript.

#### **Notificaciones y Alertas**

- WebSockets con Django Channels o Socket.io en Node.js.
- Emails y notificaciones push con Firebase o Twilio.

#### **Consideraciones de Seguridad:**

- **Autenticación de usuarios:** El sistema debe permitir que cada usuario tenga una cuenta personal protegida por credenciales seguras.
- **HTTPS:** Utilización de HTTPS para proteger la comunicación entre el cliente y el servidor.
- **Encriptación de datos:** Aplicación de encriptación para proteger la información almacenada en la base de datos.
- **Roles y permisos de usuario:** Los usuarios solo pueden acceder y modificar sus propios datos, estableciendo permisos claros.
- **Validaciones de entrada de datos:** Implementar validaciones para evitar registros incorrectos o manipulaciones de datos.