**YouLumiere: Monitoreo Inteligente de la Salud**

YouLumiere es una innovadora plataforma diseñada para el monitoreo remoto de la salud de pacientes de la tercera edad, aprovechando tecnologías emergentes como el Internet de las Cosas Médicas (IoMT) y la Inteligencia Artificial.

(Aquí va un video)

A través del IoMT, YouLumiere conecta sensores de signos vitales con certificación médica para medir con precisión parámetros esenciales como temperatura corporal, niveles de glucosa en sangre, presión arterial, oxigenación, peso y otros indicadores clave. Esto permite al personal médico monitorear la salud de sus pacientes en tiempo real, sin importar la distancia.

(Aquí van imágenes reales)

Asistente Virtual con Inteligencia Artificial

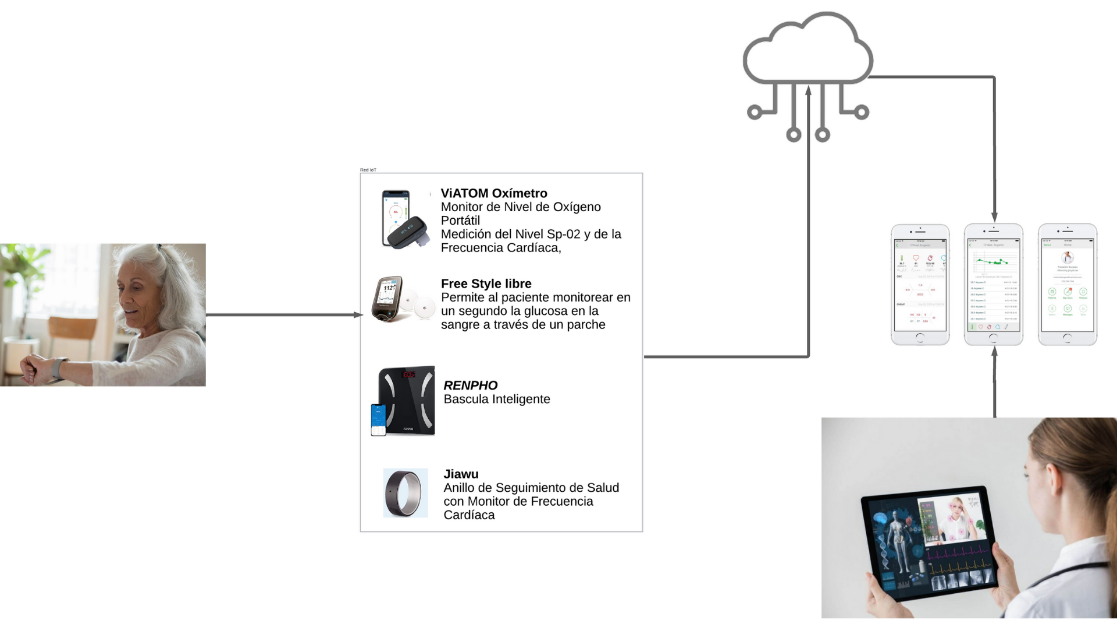
Además del monitoreo de signos vitales, YouLumiere busca mejorar la calidad de vida de los pacientes mediante un asistente virtual basado en inteligencia artificial. Este asistente proporcionará acompañamiento a personas de la tercera edad que viven solas, brindándoles apoyo en su rutina diaria y facilitando la interacción con el sistema de monitoreo.

Gracias a su integración con la red de sensores, el asistente podrá ofrecer recordatorios personalizados sobre la toma de medicamentos, hábitos saludables y alertas médicas cuando detecte anomalías en los signos vitales. Asimismo, servirá como un compañero interactivo que fomentará el bienestar emocional de los pacientes.

**Arquitectura del Sistema**

En la siguiente imagen se ilustra la arquitectura de YouLumiere. En el extremo izquierdo, un paciente utiliza distintos sensores para la medición de sus signos vitales. Estos sensores transmiten los datos a un **Gateway**, el cual se encarga de recolectar, procesar y enviar la información a través de **Bluetooth**. Posteriormente, el Gateway envía los datos procesados a una base de datos en la nube, donde quedan almacenados de forma segura.

Finalmente, los médicos pueden acceder a esta información mediante una interfaz gráfica intuitiva, ya sea a través de una página web o una aplicación móvil, lo que les permite visualizar los signos vitales de sus pacientes de manera remota y en tiempo real.



Desde un enfoque técnico, YouLumiere se basa en una arquitectura **IoMT (Internet de las Cosas Médicas)** compuesta por hardware y software, dividiéndose en los siguientes componentes clave:

1. **Sensores de signos vitales:** Dispositivos médicos encargados de medir parámetros como temperatura corporal, glucosa en sangre, presión arterial, oxigenación, peso y otros signos vitales relevantes.
2. **Gateway:** Un nodo de comunicación que recibe los datos de los sensores, los procesa y los envía a la nube para su almacenamiento y análisis.
3. **Almacenamiento:** Una base de datos segura donde se guardan los registros de cada paciente, permitiendo el acceso histórico y el análisis de tendencias en su salud.
4. **Interfaz de usuario:** Plataforma accesible desde una página web y una aplicación móvil, donde médicos y profesionales de la salud pueden visualizar los datos de los pacientes en tiempo real.

**Sensores en la Red IoMT**

En la red **IoMT** de YouLumiere, se integran sensores certificados para la medición de signos vitales, como:

* Termómetros digitales
* Glucómetros
* Tensiómetros
* Pulsioxímetros
* Básculas inteligentes
* Otros sensores médicos según las necesidades del personal de salud

Para garantizar la conectividad con el sistema, todos los sensores utilizados deben contar con un **módulo Bluetooth**, ya que esta tecnología permite la transmisión de datos en tiempo real hacia el **Gateway**, optimizando la comunicación y asegurando la precisión de los registros médicos.

### ****Gateway: Procesamiento y Transmisión de Datos desde los Sensores IoMT****

El **Gateway** es un componente fundamental en la arquitectura de **YouLumiere**, ya que actúa como intermediario entre los **sensores IoMT** y la **base de datos en la nube**. Su función principal es recibir los datos de los dispositivos biomédicos a través de **Bluetooth**, procesarlos y enviarlos en tiempo real a **Firebase Firestore** para su almacenamiento y posterior visualización en la interfaz de usuario.

## **1️⃣ Hardware del Gateway**

El Gateway está implementado en una **Raspberry Pi**, un dispositivo de bajo consumo y alta flexibilidad, ideal para entornos IoT.

### ****Especificaciones del Hardware****

* **Raspberry Pi 4 Model B** con 4GB de RAM
* **Módulo Bluetooth 5.0** integrado para la comunicación con los sensores
* **Conexión WiFi/Ethernet** para el envío de datos a la nube
* **Almacenamiento en microSD** con sistema operativo **Raspberry Pi OS**

## **2️⃣ Software del Gateway**

El software del Gateway está desarrollado en **Python**, utilizando librerías específicas para la comunicación con los sensores y la gestión de datos.

### ****Tecnologías utilizadas:****

* **PyBluez** para la comunicación Bluetooth con los sensores IoMT
* **Firebase Admin SDK** para el envío de datos a Firestore
* **Pandas y NumPy** para el procesamiento y validación de datos biomédicos
* **MQTT (opcional)** para transmisión de datos en arquitecturas más escalables

## **3️⃣ Proceso de Adquisición de Datos**

1. **Recepción de datos desde los sensores:**
   * Los sensores envían los valores biomédicos vía **Bluetooth Low Energy (BLE)** o **Bluetooth clásico**.
   * La Raspberry Pi escanea y se empareja con los dispositivos mediante **PyBluez**.
   * Se establece una conexión estable y se reciben los datos en formato **byte** o **string**.
2. **Procesamiento y validación de datos:**
   * Se convierten los valores crudos en **unidades médicas estándar** (°C, mmHg, mg/dL, %SpO2).
   * Se validan los datos para evitar valores erróneos o inconsistentes.
   * Se estructuran los datos en formato **JSON** para su posterior almacenamiento.
3. **Envío de datos a la nube:**
   * Se establece una conexión segura con **Firebase Firestore** usando el **Firebase Admin SDK**.
   * Los datos se envían en **tiempo real** con la estructura adecuada.
   * Se asegura que cada paciente tenga su historial de signos vitales almacenado correctamente.

## **4️⃣ Seguridad y Fiabilidad**

Para garantizar la integridad de los datos y la seguridad en la transmisión, se implementan las siguientes estrategias:

* **Autenticación con claves API** para conectarse a Firebase de manera segura.
* **Cifrado de datos sensibles** antes del envío a la nube.
* **Validación de dispositivos Bluetooth** para evitar conexiones no autorizadas.
* **Registro de logs** en archivos locales para auditoría y depuración de errores.
* **Reconexión automática** en caso de pérdida de conexión con la nube.

## **5️⃣ Flujo Completo de Datos**

1. **El sensor IoMT mide un signo vital** y lo transmite vía **Bluetooth**.
2. **El Gateway (Raspberry Pi) recibe los datos** y los procesa.
3. **Los datos se validan y estructuran** en un formato estándar.
4. **El Gateway envía los datos** a **Firebase Firestore** a través de una conexión segura.
5. **La interfaz de usuario obtiene los datos en tiempo real** para su visualización.

**Almacenamiento**

El almacenamiento de datos es un pilar fundamental en **YouLumiere**, ya que permite gestionar y estructurar la información de los signos vitales de los pacientes de manera eficiente, segura y escalable. Para este propósito, se ha elegido **Firebase Firestore**, una base de datos NoSQL en la nube que ofrece **sincronización en tiempo real, alta disponibilidad y escalabilidad automática**.

## **1️⃣ Motivos para Usar Firebase Firestore**

Firebase Firestore ha sido seleccionado por los siguientes beneficios clave:

✅ **Sincronización en tiempo real:** Los datos de los signos vitales de los pacientes pueden ser consultados instantáneamente desde la aplicación móvil y web.  
✅ **Escalabilidad automática:** Puede manejar desde unos pocos registros hasta millones sin necesidad de configurar servidores adicionales.  
✅ **Seguridad avanzada:** Se pueden definir reglas de acceso con autenticación basada en Firebase Authentication.  
✅ **Soporte para múltiples plataformas:** Se integra fácilmente con aplicaciones **Flutter, Next.js y Python**.  
✅ **Almacenamiento estructurado y optimizado:** Al ser una base de datos NoSQL, permite consultas rápidas sin comprometer el rendimiento.

## **2️⃣ Estructura de Almacenamiento**

En **Firebase Firestore**, los datos se organizan en **colecciones** y **documentos**.

📂 **Colección:** pacientes  
📄 **Documentos:** Cada paciente tiene un documento con su ID único (id\_paciente).

## **3️⃣ Flujo de Almacenamiento de Datos**

1. **Recepción de datos desde el Gateway:**
   * La **Raspberry Pi** capta los valores de los sensores mediante **Bluetooth**.
   * Los datos son procesados y estructurados en formato **JSON**.
   * Se establecen reglas de validación para evitar registros erróneos o duplicados.
2. **Envío de datos a Firebase Firestore:**
   * Se realiza una **conexión segura** con Firebase a través del SDK de Python.
   * Los datos se **almacenan en la subcolección signos\_vitales** del paciente correspondiente.
   * Se guarda un **timestamp UTC** para mantener un historial médico ordenado.
3. **Acceso y consulta de datos desde el frontend:**
   * La interfaz web y la aplicación móvil consumen los datos en **tiempo real**.
   * Se aplican filtros para mostrar tendencias y gráficas de evolución de signos vitales.
   * Se generan **alertas médicas automáticas** si los valores están fuera de los rangos normales.

## **4️⃣ Seguridad y Protección de Datos**

Para garantizar la seguridad de la información médica de los pacientes, se implementan las siguientes medidas:

🔐 **Reglas de acceso en Firebase Firestore:** Solo usuarios autenticados pueden leer o escribir datos.  
🔐 **Cifrado de datos en tránsito y en reposo:** Se usa **SSL/TLS** en la comunicación con la base de datos.  
🔐 **Autenticación basada en Firebase Authentication:** Solo médicos autorizados pueden acceder a los datos de los pacientes.  
🔐 **Reglas de integridad de datos:** Se evitan registros erróneos mediante validaciones previas antes de almacenar la información.

## **5️⃣ Consultas y Recuperación de Datos**

Las aplicaciones pueden recuperar información de la base de datos de manera eficiente utilizando consultas optimizadas.

Ejemplo de flujo de consulta en la interfaz web:

1️⃣ El médico inicia sesión y selecciona un paciente.  
2️⃣ Se consulta la colección pacientes y se recuperan sus datos personales y signos vitales.  
3️⃣ Se generan gráficos interactivos con tendencias de temperatura, glucosa, presión arterial y oxígeno en sangre.  
4️⃣ Si un signo vital está fuera del rango normal, se activa una **alerta médica**.

### ****Interfaz de Usuario (Frontend) y Consumo de Datos desde Firebase Firestore****

La interfaz de usuario de **YouLumiere** está diseñada para ofrecer una experiencia fluida e intuitiva a médicos y cuidadores, permitiéndoles visualizar y analizar en tiempo real los signos vitales de los pacientes.

La interfaz está disponible en **dos plataformas**:

1. **Aplicación web**, desarrollada con **Next.js** para acceso desde navegadores.
2. **Aplicación móvil**, construida con **Flutter** para dispositivos Android e iOS.

Ambas interfaces se conectan a **Firebase Firestore** para obtener y actualizar la información de los pacientes de manera eficiente y segura.

**Arquitectura Front End (Interfaz de usuario)**

El **frontend** sigue una arquitectura basada en componentes y estado global, utilizando tecnologías modernas para garantizar rendimiento y escalabilidad:

* **Next.js** (para la web), que permite renderizado en el servidor y optimización en la carga de datos.
* **Flutter** (para la app móvil), ofreciendo una experiencia nativa y fluida en distintos dispositivos.
* **Firebase Firestore SDK**, para la conexión en tiempo real con la base de datos.
* **React Query** y **Provider**, utilizados para la gestión del estado y almacenamiento en caché de datos.
* **Tailwind CSS** y **Material Design**, que proporcionan una interfaz profesional y moderna.

## 2.Consumo de datos desde Firebase

Los datos de los signos vitales se obtienen en tiempo real desde **Firebase Firestore**, lo que permite a los médicos visualizar información actualizada sin necesidad de recargar la página o la aplicación.

El sistema se basa en suscripciones a los datos, lo que significa que cualquier cambio realizado en la base de datos se refleja automáticamente en la interfaz del usuario.

## 3. **Visualización de Datos**

Los datos obtenidos se presentan en la interfaz mediante gráficos interactivos, tablas dinámicas y alertas visuales.

* **Gráficos de línea** para la evolución de la presión arterial, temperatura corporal y nivel de glucosa.
* **Tablas interactivas** con filtros y búsqueda rápida de pacientes.
* **Notificaciones en tiempo real** si un signo vital supera los valores normales establecidos.

**Seguridad en el Acceso a Datos**

Para garantizar la privacidad de los pacientes, la plataforma implementa autenticación con **Firebase Authentication** y reglas de seguridad en Firestore.

Cada usuario tiene permisos específicos:

* **Pacientes** solo pueden acceder a su información personal.
* **Médicos autorizados** pueden visualizar los signos vitales de sus pacientes asignados.
* **Administradores** tienen acceso completo para gestionar la plataforma.

## **5. Flujo Completo de Datos**

1. **El paciente utiliza los sensores IoMT** para medir signos vitales.
2. **Los datos se transmiten al Gateway** mediante Bluetooth.
3. **El Gateway procesa y envía los datos** a Firebase Firestore.
4. **El frontend (web y móvil) obtiene los datos** en tiempo real.
5. **Los datos son visualizados** en gráficos interactivos, permitiendo un monitoreo eficiente.

PREGUNTAR A CHAT GPT   
Necesito que s eintegre dentro de mi pagina, aunque puedes hacerlo de forma independiente y luego yo le pido a una IA que tiene conocimeinto de mi pagina que integre el fragmento de codigo, solo dame un prompt que me permita agregar de forma correcta el fragmento de codigo a mi pagina

Estructura HTML

Si prefiero una estructura con secciones bien definidas

Estilo y diseños

Si incluye css

Si quiero que el html sea responsivo

Elementos multimedia

Si quiero que utilices esas etiquetas para espacios reservados

Obvia las imagenes que estan ahi luego las pondre yo