FITNESS TRACKER

DISEÑO DEL SISTEMA Y BBDD

**Realizado por Aaron Esono, Guillermo Quintanar y Hugo Pelayo**

**20 de abril de 2024**

**Índice**

1. **Arquitectura de la base de datos**
2. **Sistema de inteligencia artificial**
3. **Arquitectura de la base de datos**
4. **Arquitectura de la base de datos**

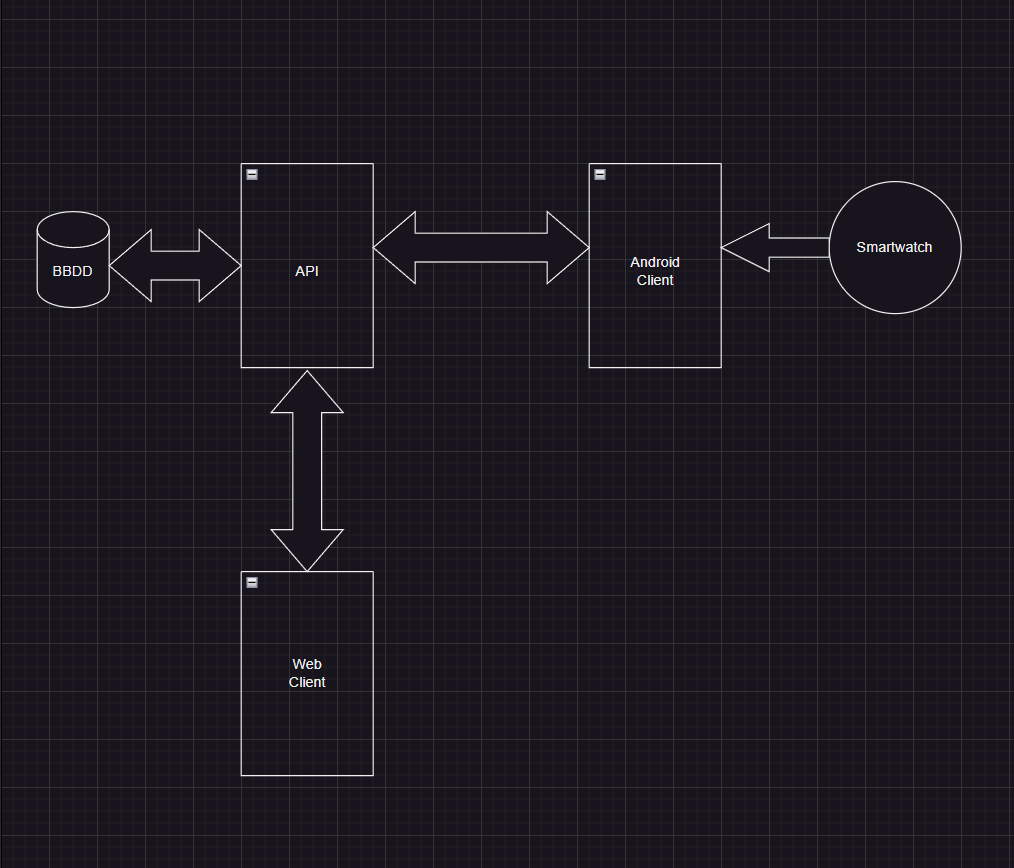
Nuestro sistema consta de cuatro componentes. Dos servicios REST con sus respectivos endpoints. Uno de los servicios REST es el Base que contiene la lógica de negocio relacionada con el manejo de cuentas de clientes así como su validación y autenticación al iniciar sesión; paralelo a este está el servicio REST AI Assistant. Es una API que maneja la lógica de negocio respecto la generación de itinerarios relativos a dietas o rutinas y que también puede asistir al cliente en las necesidades que este pueda tener en relación con sus rutinas de ejercicios físicos diaria, entre otros aspectos.

Para validar que el usuario que realiza peticiones a AI Assistant es válido, este deberá comunicarse con este servicio usando su token de inicio de sesión, AI Assistant se comunicará con el REST Base para validar el usuario, si este es un usuario válido para poder utilizar los servicios de AI Assistant, este responderá con un token al cliente con que se puede establecer todo tipo de peticiones de forma más segura y eficiente.

Por otro lado tenemos dos clientes. Un cliente web y uno Android. Ambos podrán ser capaces de comunicarse con nuestras API REST. El cliente Android por su lado consta de otra fuente de datos que es el reloj inteligente, este le proporciona datos como el número de pasos realizados al día, calorías quemadas en un período de tiempo, entre otros.

Por la naturaleza de los datos que se van a guardar en la base de datos y cómo están relacionados los modelos, estos se van a serializar a una base de datos documental, para ello se utilizará MongoDB que es un gestor de bases de datos NoSQL con muy eficiente para lecturas frecuentes de datos.

A continuación se muestra un esquema del sistema:



1. **Sistema inteligencia artificial**

El sistema de inteligencia artificial es un módulo tanto para la aplicación Web como el cliente Android que sirve de asistente virtual para el usuario. Permite al usuario realizar consultas a la hora de construir rutinas de ejercicios diarias o semanales.

Empezando por la inteligencia artificial que se utiliza en este módulo. Se va a utilizar el modelo de lenguaje GPT 4 Turbo con Vision, la versión de GPT capaz de aceptar imágenes y trabajar sobre éstas. Es una versión mejorada de GPT 3.5 Turbo, con la mejora de que puede procesar y generar imágenes, juntamente con la capacidad de poder procesar aún más palabras con más fluidez y precisión. Open AI expone una serie de endpoints a través de los cuales se puede realizar peticiones a estos modelos. Para ello es necesario primero una clave API que se puede generar desde su página.

Este servicio no es gratuito y está sujeto a unos límites de uso. Los límites de uso de la API son restricciones cruciales que la plataforma impone para regular el acceso y garantizar una experiencia equitativa para todos los usuarios. Estos límites, también conocidos como "Rate Limits", definen la cantidad máxima de veces que un usuario o cliente puede acceder a los servicios dentro de un período de tiempo específico.

Las razones principales para imponer estas restricciones están recogidas en la guía de desarrollo y son:

**Protección contra el Abuso:** La API se protege contra posibles abusos o malos usos que podrían sobrecargar los servidores o interrumpir el servicio. Esto se logra limitando la cantidad de solicitudes que un usuario puede realizar en un período de tiempo dado.

**Equidad de acceso:** los límites de uso garantizan que todos los usuarios tengan acceso justo a nuestros servicios. Al limitar el número de solicitudes que un usuario puede hacer, evitamos que un individuo o entidad monopolice los recursos disponibles, lo que podría ralentizar la experiencia para otros usuarios.

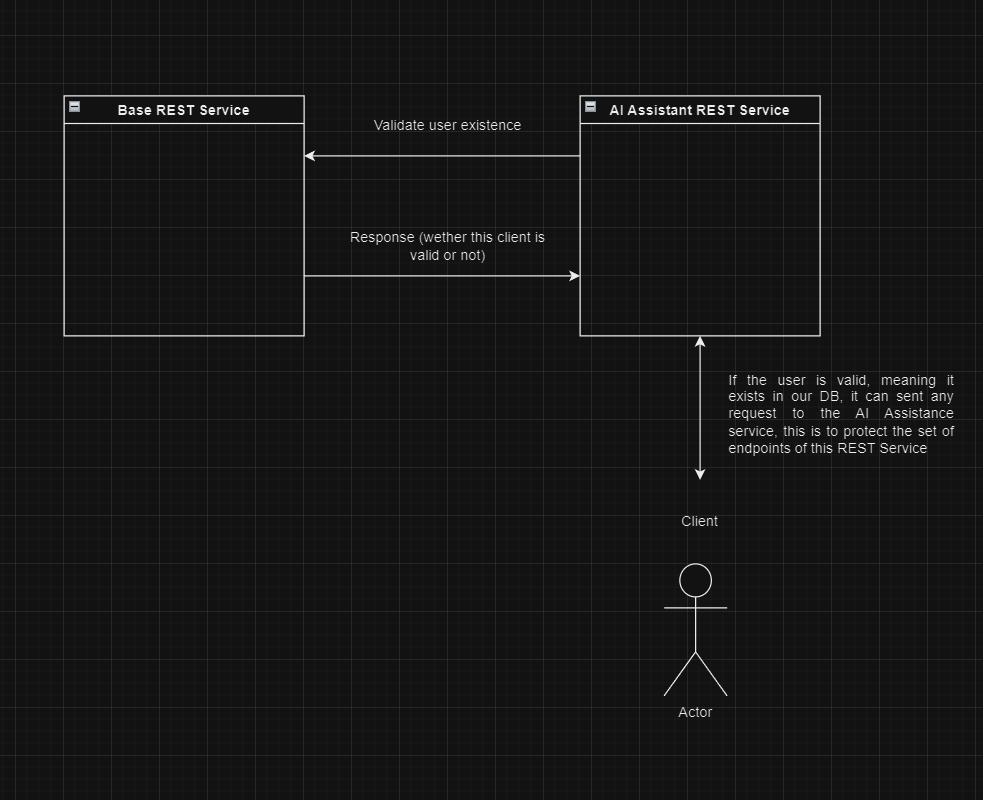
**Gestión de la carga de infraestructura:** Los límites de uso ayudan a administrar la carga total en nuestra infraestructura. Al controlar la cantidad de solicitudes que recibimos, podemos garantizar un rendimiento estable y consistente de nuestros servidores, incluso durante períodos de alta demanda.

Los límites de uso se expresan en diferentes métricas, que incluyen solicitudes por minuto (RPM o Request Per Minute), solicitudes por día (RPD o Request Per Day), tokens por minuto (TPM o Token Per Minute), tokens por día (TPD o Tokens Per Day) e imágenes por minuto (IPM o Images Per Minute). Estos límites se aplican según lo que ocurra primero en cada caso.

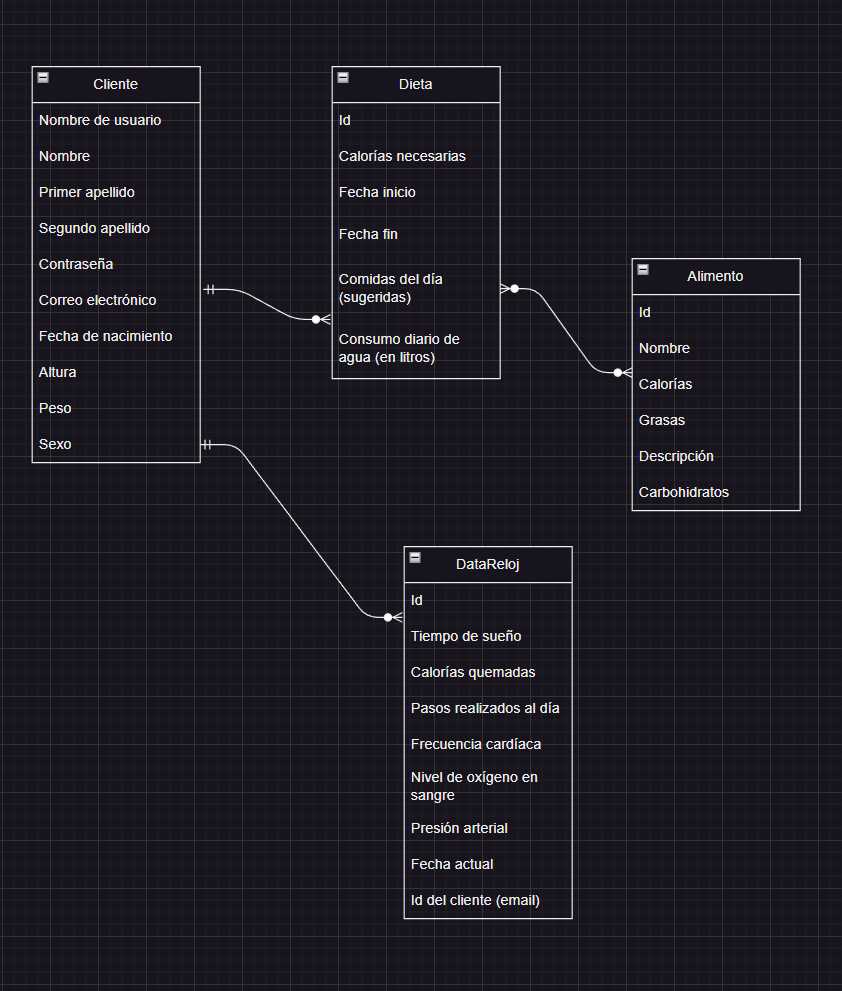
Es importante destacar que los límites de la API se definen tanto a nivel de organización como a nivel de proyecto, no a nivel de usuario individual. Además, los límites pueden variar según el modelo específico que se esté utilizando en la API.

La API REST que se comunicará con los servicios de OpenAI juntamente con los servicios de autenticación de usuarios se va a realizar con el lenguaje de programación C# usando ASP.NET Core, un framework que facilita mucho la creación de servicios REST.

Para realizar peticiones a la API de OpenAI se va a utilizar la librería de .NET OpenAI-API-dotnet. Una librería de código abierto que facilita la tarea de realizar peticiones contra la API de OpenAI, agilizando el trabajo de procesar las respuestas recibidas de los endpoints de Open AI en clases e interfaces más sencillas de usar.



1. **Arquitectura de la base de datos**



En la arquitectura de la base de datos de nuestra aplicación, hemos identificado cuatro modelos principales: Cliente, Dietas, Alimentos y Datos de Reloj. Cada uno de estos modelos tiene su propio propósito y están interrelacionados de acuerdo con las funcionalidades de nuestra aplicación.

**Modelo de Cliente**

El modelo de Cliente representa a los usuarios de nuestra aplicación, cada uno de los cuales tiene una cuenta registrada en nuestra base de datos. Cada registro de cliente contiene el nombre de usuario (un nombre identificativo de la cuenta que es opcional), el nombre del cliente, primero y segundo apellido, la contraseña (esta se va aguardar encriptada en nuestra base de datos), el correo electrónico, la fecha de nacimiento (con que se puede deducir la edad siempre que sea necesaria, la altura, el peso y el sexo.

**Modelo de Dietas**

El modelo de Dietas representa los planes de dieta disponibles en nuestra aplicación. Cada dieta viene determinada por un identificador único que genera el sistema gestor de la base de datos automáticamente, las calorías a quemar necesarias para cumplir con la dieta, la fecha de inicio y fin de la dieta, las comidas del día (aquí se incluye desayuno, comida, cena, etcétera) y el consumo de agua diario o por unidad de tiempo.

Un cliente puede estar asociado con una o varias dietas a lo largo del tiempo, dependiendo de sus objetivos de salud y preferencias personales.

**Modelo de Alimentos**

El modelo de Alimentos representa los alimentos específicos que forman parte de cada dieta. Cada alimento puede tener atributos como nombre, descripción, valor nutricional, etcétera. Los alimentos están relacionados con las dietas, ya que cada alimento específico solo se relaciona con una dieta en particular. Es el cliente quien puede seleccionar los alimentos dentro de una dieta, en función de sus necesidades y preferencias dietéticas.

**Modelo de Datos de Reloj**

El modelo de Datos de Reloj representa la información recopilada de dispositivos de seguimiento de actividad física, como relojes inteligentes o bandas de fitness. Estos datos se almacenan en documentos que vienen identificados por un identificador único que proporciona la base de datos de manera automática, paralelamente con atributos como el tiempo de sueño, calorías quemadas en un día específico, pasos realizados al día, frecuencia cardíaca, nickel de oxígeno en sangre, presión arterial, el día del que se está haciendo seguimiento y una referencia al email del cliente para señalizar a qué cliente pertenece el documento. Los datos de reloj pueden estar asociados con un cliente específico para realizar un seguimiento de su actividad física y progreso en sus objetivos de salud y fitness.

**Interrelaciones**

Cliente - Dietas: Un cliente puede tener una relación de uno a muchos con las dietas, lo que significa que un cliente puede estar asociado con varias dietas a lo largo del tiempo.

Dietas - Alimentos: Existe una relación de muchos a muchos entre las dietas y los alimentos, lo que significa que cada alimento específico está asociado con varias dietas, al igual que varias dietas pueden tener asociados alimentos varios alimentos.

Cliente - Datos de Reloj: Un cliente tiene una relación de uno a muchos con los datos de reloj, lo que significa que un cliente puede tener múltiples registros de datos de reloj asociados con su actividad física.