FITNESS TRACKER
DISEÑO DEL SISTEMA Y BBDD

Realizado por Aaron Esono, Guillermo Quintanar y Hugo Pelayo

20 de abril de 2024

Índice

Arquitectura backend

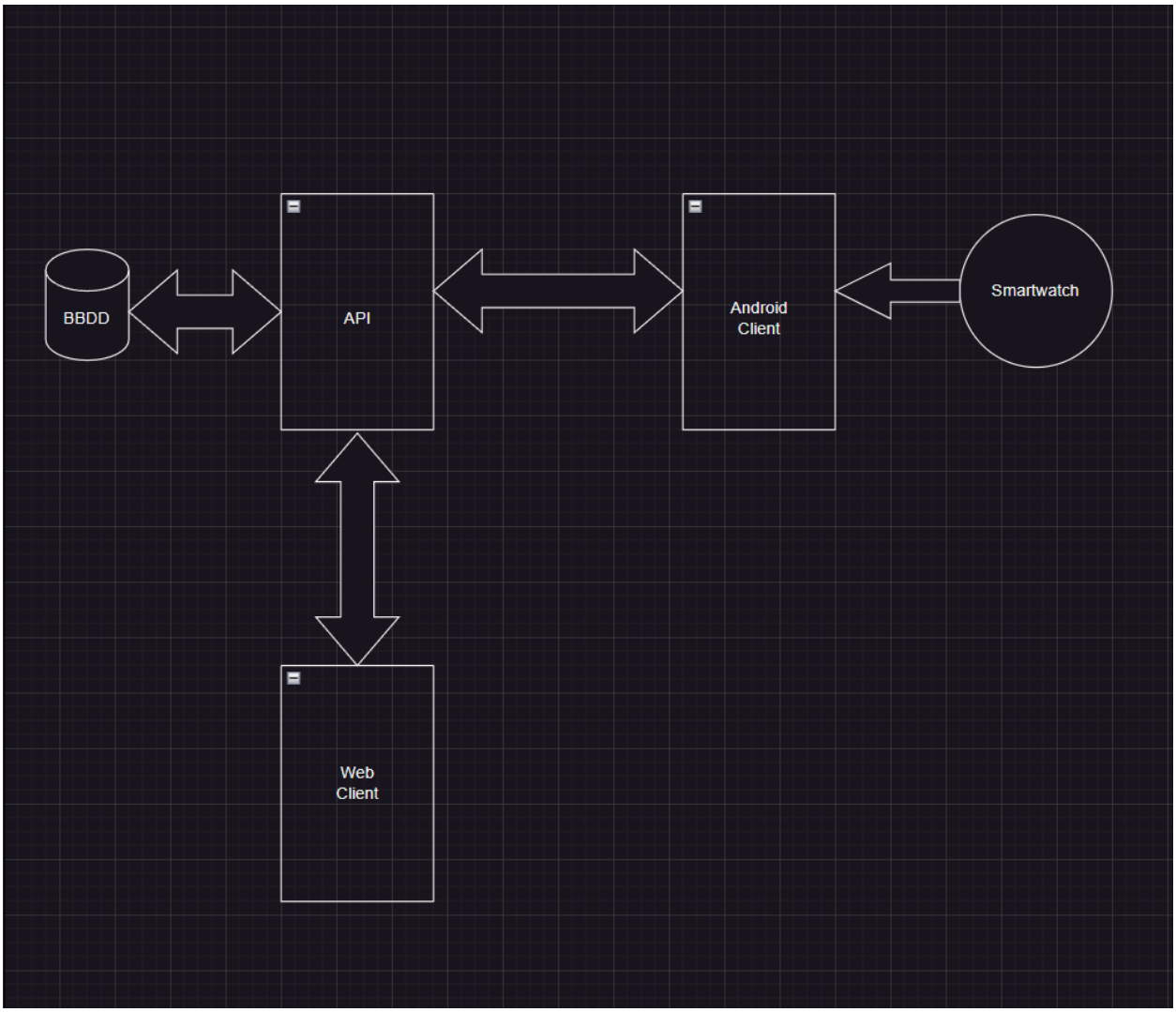
Nuestro sistema consta de cuatro componentes. Dos servicios REST con sus respectivos endpoints. Uno de los servicios REST es el Base que contiene la lógica de negocio relacionada con el manejo de cuentas de clientes así como su validación y autenticación al iniciar sesión; paralelo a este está el servicio REST Sistema External Services, que entre otras APIs, incluye la de la IA y una sobre información nutricional. El API de la IA es una 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.

Por otro lado tenemos dos clientes. Un cliente web y uno Android. Ambos podrán ser capaces de comunicarse con nuestro backend. 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.

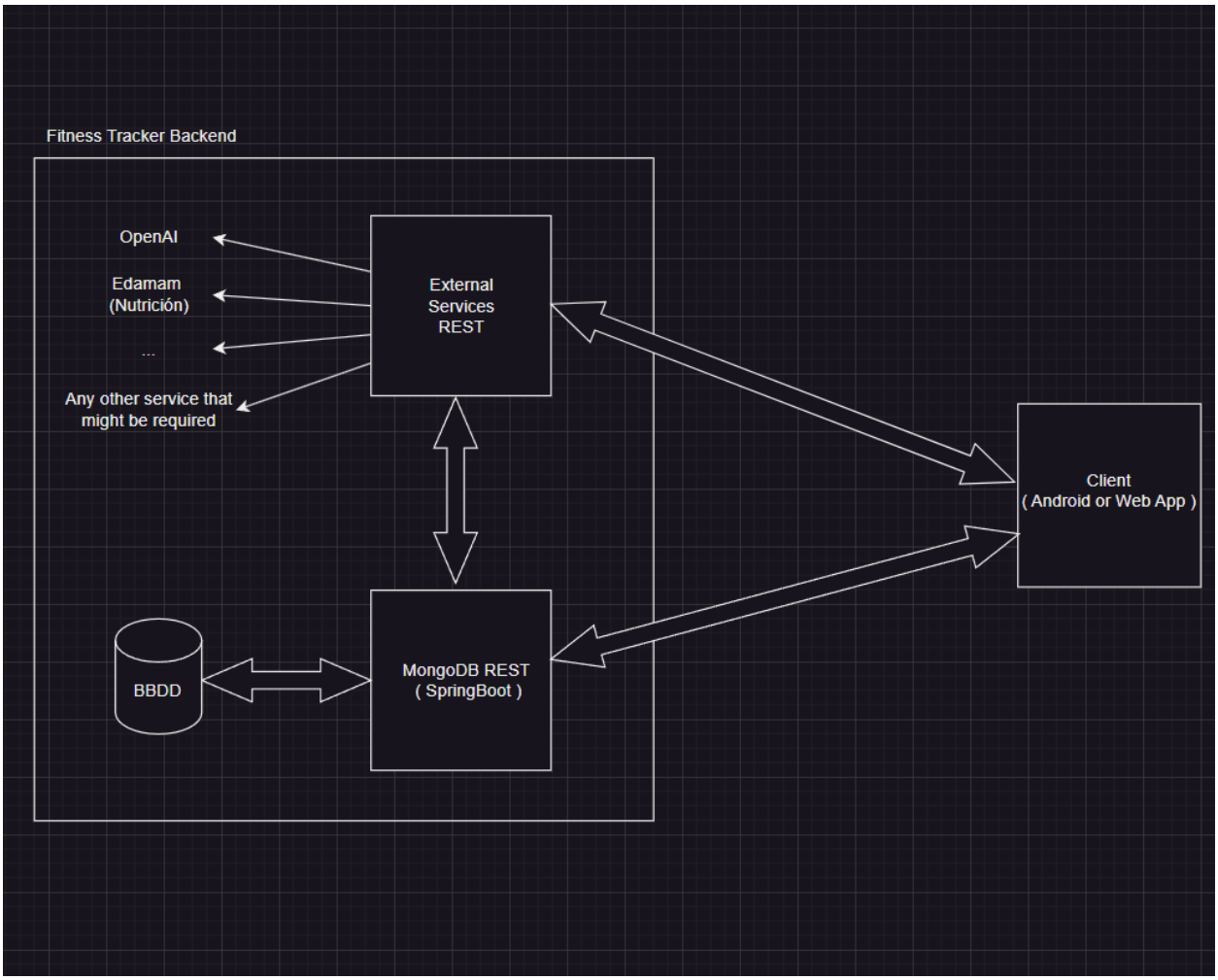
Por la naturaleza de los datos que se van a guardar en la base de datos y cómo están relacionados los modelos, estos se van a serializar a una base de datos documental, para ello se utilizará MongoDB que es un gestor de bases de datos NoSQL con muy eficiente para lecturas frecuentes de datos.

A continuación se muestra un esquema del sistema:



El backend consta de una, por una parte, de una API REST desarrollada sobre el framework de ASP.NET Core, este es un framework que facilita la creación de un sistema a modo de pasarela entre nuestro backend y servicios externos como el de nutrición y el de OpenAI. ASP.NET Core es un framework con que es sencillo tener una API REST robusta y escalable.

Por otra parte tenemos la API REST que gestiona la lógica de negocio de los clientes así como almacenar datos relativos a estos, incluimos aquí las dietas, registros del reloj inteligente, entre otros. Esta API es la que tiene conexión directa con la base de datos y se desarrolla en Java utilizando el framework de Spring Boot. A continuación, un esquema ilustrativo.



La razón detrás de esta arquitectura yace en que queremos diferenciar los servicios que son propiamente nuestros de los que son dependientes de servicios externos. MongoDB REST gestiona, como se ha mencionado anteriormente, la lógica y negocio relacionados con la manipulación de datos relativos a nuestros clientes. Paralelo a ello, External Services es un módulo que ofrece nuestro backend para ciertas funcionalidades que requieren nuestros clientes pero que no son servicios completamente dependientes de nosotros.

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.

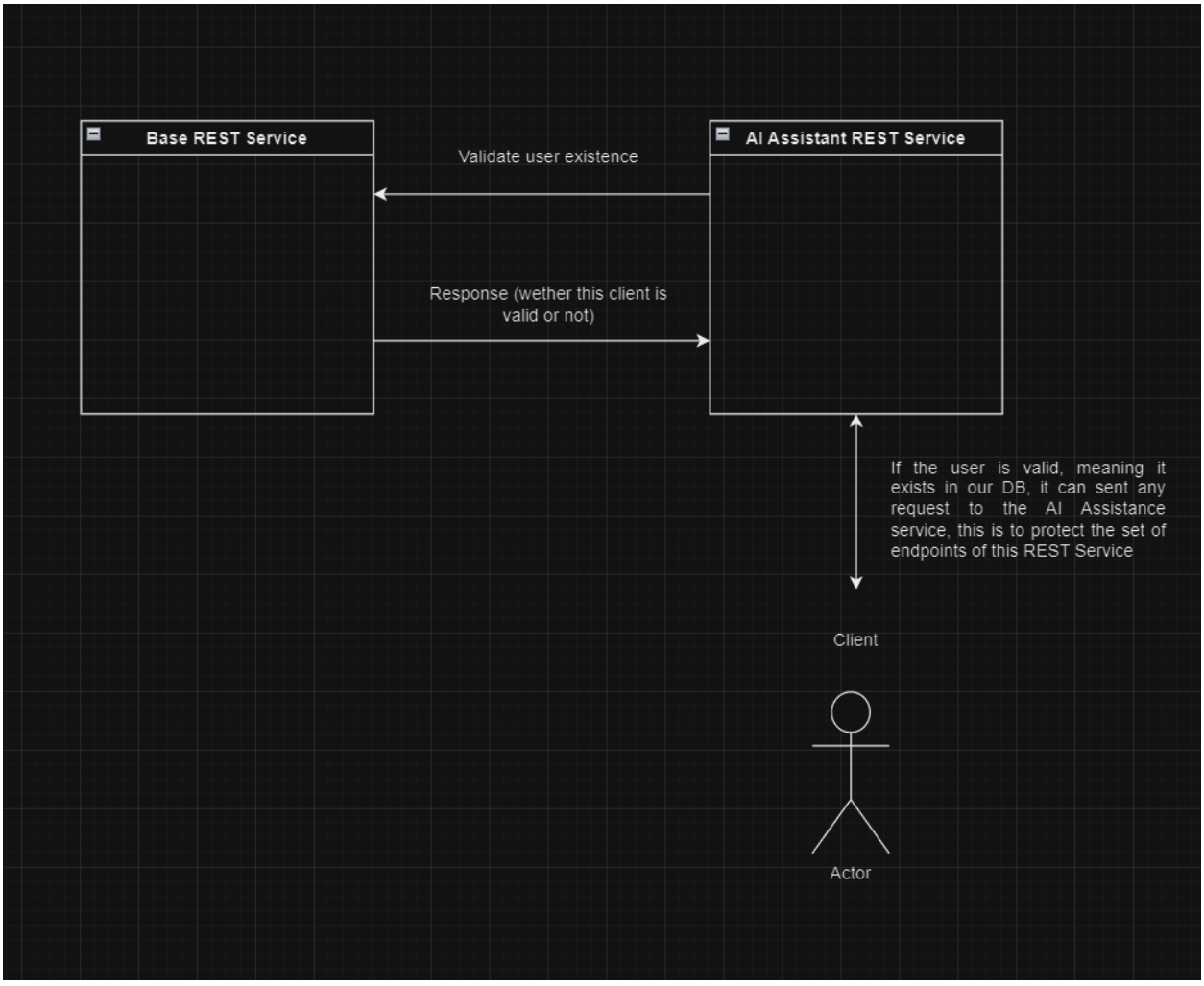
Los límites de uso se expresan en diferentes métricas, que incluyen solicitudes por minuto (RPM o Request Per Minute), solicitudes por día (RPD o Request Per Day), tokens por minuto (TPM o Token Per Minute), tokens por día (TPD o Tokens Per Day) e imágenes

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.



Arquitectura de la base de datos

Dietas

Una dieta se puede definir como el conjunto de alimentos y bebidas que una persona consume regularmente para mantener su salud y nutrición. Sin embargo, en nuestra aplicación consideramos la dieta como un plan estructurado y personalizado de ingesta de alimentos diseñado con objetivos específicos, como perder peso, ganar masa muscular, mejorar la salud cardiovascular, entre otros aspectos.

Para cada usuario se mantiene constancia de las dietas que lleva a lo largo del tiempo. Un usuario puede tener varias dietas a lo largo del tiempo en donde puede incluir todo tipo de comidas, las cuales incluso pueden repetirse entre dietas.

Las dietas se generan con la ayuda del módulo de External Services, a través del soporte de la inteligencia artificial. Luego esta se almacena en MongoDB REST que es nuestra API REST que se encarga de manipular todos los datos e información relativa al cliente.

Para generar una dieta, el usuario deberá cumplir un formulario similar al siguiente:

Ejercicios físicos

El módulo de Ejercicios Físicos de nuestra aplicación complementa el aspecto dietético del plan de salud y bienestar del usuario. Este módulo proporciona un enfoque integral para ayudar a los usuarios a alcanzar sus objetivos de fitness y mejorar su calidad de vida.

Este módulo permite a los usuarios registrar sus sesiones de ejercicio, incluyendo el tipo de actividad, la duración y la intensidad. También muestra a los usuarios un resumen de su actividad física a lo largo del tiempo, lo que les permite visualizar su progreso y establecer metas alcanzables.

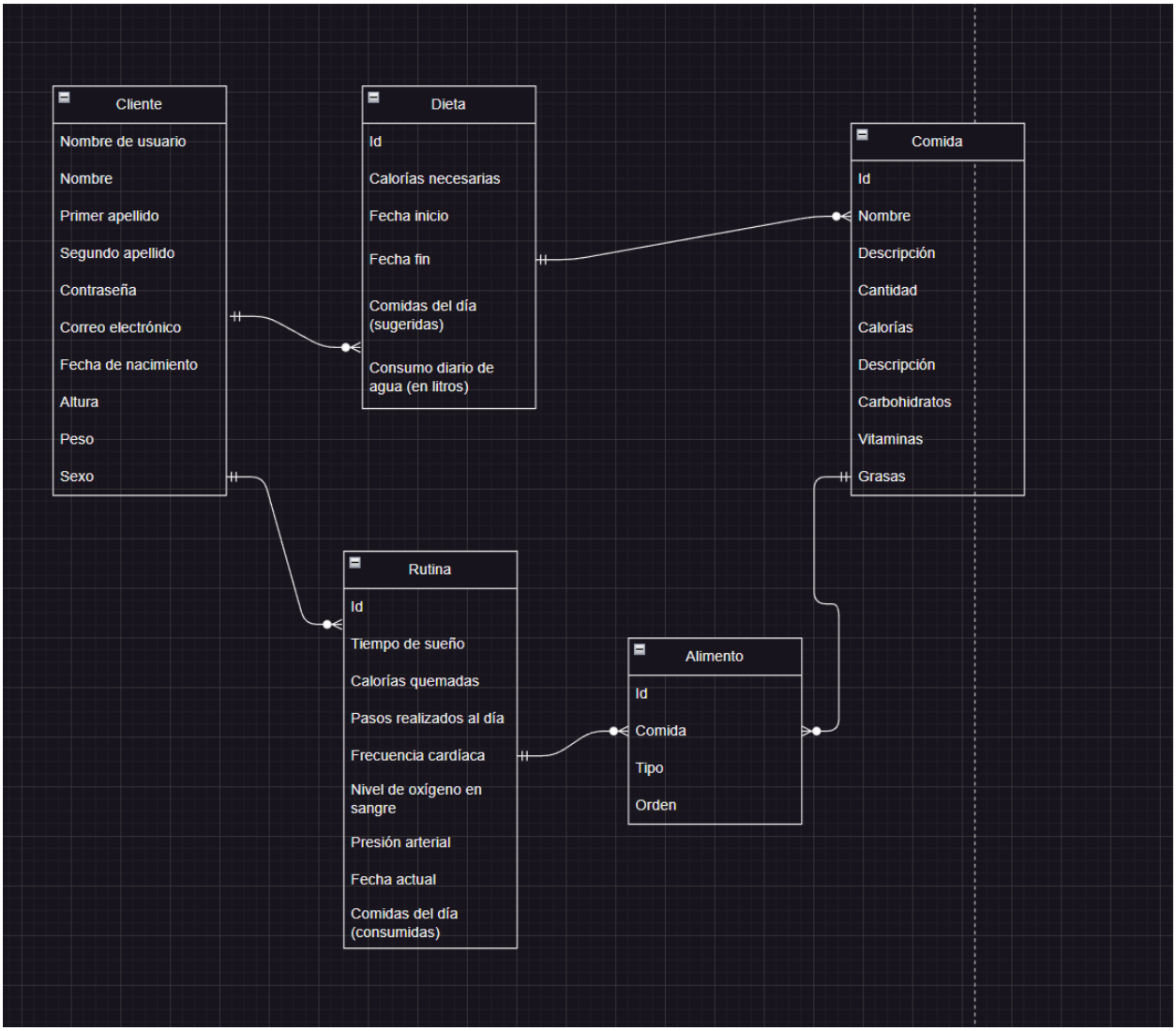
Consejos de alimentación y ejercicio

El módulo de Consejos de Alimentación y ejercicio proporciona a los usuarios orientación y apoyo adicional para ayudarles a adoptar hábitos de vida más saludables.

Este módulo ofrece consejos específicos de alimentación adaptados a las necesidades y objetivos de cada usuario, teniendo en cuenta factores como edad, género, nivel de actividad y preferencias alimenticias. Además, proporciona ideas y recetas de comidas

nutritivas y equilibradas, ayudando a los usuarios a diversificar su dieta y disfrutar de opciones más saludables. Facilita la planificación de comidas diarias de acuerdo con los objetivos de salud y las preferencias personales de los usuarios, y proporciona recomendaciones de ejercicios específicos y estrategias de entrenamiento adaptadas a sus objetivos de fitness.

A continuación, se muestra un esquema del modelo de datos de nuestra base de datos.



En la arquitectura de la base de datos de nuestra aplicación, hemos identificado cuatro modelos principales: Cliente, Dieta, Ejercicio físico (Rutina) y Comida. Cada uno de estos modelos tiene su propio propósito y están interrelacionados de acuerdo con las funcionalidades de nuestra aplicación.

Modelos

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, la cuenta mantiene constancia de los datos personales del cliente. A continuación, se listan los datos personales del cliente que se consideran relevantes para la aplicación:

- **Nombre de usuario:** Campo opcional con que el usuario puede indicar cómo quiere ser dirigido en la aplicación, si no se especifica, se dirige a él por su nombre.
- **Nombre:** Nombres del cliente.
- **Primer apellido:** Primer apellido del cliente.
- **Segundo apellido:** Segundo apellido del cliente, no obligatorio en caso de que el cliente no lo tenga.
- **Contraseña:** Contraseña que se guarda encriptada en nuestra base de datos.
- **Correo electrónico:** Correo electrónico del cliente.
- **Fecha de nacimiento:** Fecha de nacimiento del cliente, con que se puede deducir la edad donde es necesario.
- **Altura:** Altura en centímetros del cliente.
- **Peso:** Peso en kilogramos del cliente.
- **Sexo:** Sexo del cliente. Donde puede especificar mujer u hombre. Este campo hace referencia al sexo biológico ya que se considera un factor importante para luego poder determinar las dietas.

Modelo de Dieta

El modelo de Dietas representa los planes de dieta disponibles en nuestra aplicación. Un cliente podrá tener varias dietas a lo largo del tiempo, tantas como quiera y, obviamente podamos almacenar en nuestro servidor. Las dietas son planes de comida semanales que se ajustan a las necesidades nutricionales y alimenticias que quiere llevar el cliente en un intervalo de tiempo determinado, por ende, son recomendaciones, y no necesariamente lo que el usuario ha consumido en concreto. La dieta una vez creada se puede modificar. A continuación, se listan las propiedades de una dieta:

- **Id:** Identificador único para cada comida, generado automáticamente por el sistema.
- **Calorías objetivo:** Cantidad de calorías a quemar acorde a la dieta, medidas en Kcal o Julios.
- **Fecha de inicio:** Fecha de inicio de la dieta.
- **Fecha de fin:** Fecha de fin de la dieta.
- **Comidas del día (sugeridas):** Comidas sugeridas para consumir en esta dieta.
- **Consumo de agua diario:** Consumo de agua diario mínimo recomendado.

Modelo de Comida

El modelo de Comidas representa los alimentos específicos o recetas específicas que forman parte de cada dieta o de una rutina en concreto, es decir, se diferencia entre las comidas que se sugieren para seguir una dieta en concreto y las que el usuario ha consumido. A continuación, se listan sus propiedades:

- **Id:** Identificador único para cada comida, generado automáticamente por el sistema.
- **Nombre:** Nombre de la comida (puede ser una receta completa).
- **Descripción:** Descripción de la comida.
- **Cantidad:** Cantidad en unidades de esta comida.
- **Calorías:** Cantidad de calorías proporcionadas por esta comida.
- **Carbohidratos:** Carbohidratos proporcionados por esta comida.
- **Vitaminas:** Tipo de vitaminas proporcionadas por esta comida.
- **Grasas:** Valor en grasas proporcionado por esta 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 Ejercicio Físico (Rutina)

El modelo de rutina representa la información recopilada de dispositivos del seguimiento de la actividad física del cliente, incluyendo datos recopilados de relojes inteligentes y las comidas consumidas al día.

- **Id:** Identificador único para rutinas, generado y asignado automáticamente por el sistema gestor de la base de datos.
- **Tiempo de sueño:** Indica el tiempo de sueño invertido para inhibir la somnolencia durante el día siguiente, representado en minutos.
- **Calorías quemadas:** Cantidad de calorías quemadas por ejercicio físico realizado durante el día, medido en kcal o Julios (J)
- **Pasos realizados al día:** Número de pasos realizados al día.
- **Frecuencia cardíaca:** Frecuencia de las pulsaciones del corazón por minuto. Medido en BPM o *Beats Per Minute*.
- **Nivel de oxígeno en sangre:** Porcentaje de oxígeno en sangre.

- **Presión arterial:** Presión con que circula la sangre por nuestro organismo. Medido en milímetros de mercurio, ya que resulta más informativo de cara al cliente.
- **Comidas del día:** Alimentos consumidos al día, ver el modelo de Alimento.

Modelo de Alimento

Este modelo nace como consecuencia de que el usuario no necesariamente va a ceñirse a las restricciones de la dieta y acabará consumiendo algún alimento del día. Este modelo mantiene constancia de ello y siempre es parte de una rutina, no existe por sí mismo, es decir, tiene una relación de agregación con el modelo de rutina.

Los alimentos se obtienen del servicio externo de FT - Alimentos. Estos luego se serializan a nuestra base de datos cuando se genera la dieta ya que la dieta se serializa a la base de datos y esta debe mantener constancia de los alimentos que la componen. En esencia esto implica que tenemos datos duplicados y parte de ellos en una base de datos totalmente ajena a nosotros. Esto es así porque esto nos libera, como desarrolladores, de la carga de mantener actualizada una base de datos con información nutricional lo suficientemente extensa como para asistir las necesidades de un cliente potencial.

A continuación, se listan sus atributos:

- **Id:** Identificador único generado automáticamente por la base de datos para identificar los diferentes alimentos.
- **Comida:** Comida consumida en una fecha específica.
- **Tipo de comida:** Especifica el tipo de comida, que puede ser el desayuno, almuerzo, merienda o la cena.
- **Orden:** Especifica el orden de plato, que puede ser primer plato, segundo plato, tercer plato.

Interrelaciones

Cliente y Dietas: Un cliente puede tener una relación de uno (obligatorio) a muchos (opcional) con las dietas, lo que significa que un cliente puede tener asociadas varias dietas a lo largo del tiempo, los clientes pueden obtener dietas similares pero estas son únicas por cada uno de ellos.

Dietas y Comidas: Existe una relación de muchos (opcional) a muchos (opcional) 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 y Rutina: Un cliente tiene una relación de uno a muchos con la rutina, lo que significa que un cliente puede tener múltiples registros de datos diarios relativos a su actividad física, consumo de alimentos, entre otros aspectos.

Alimento y Comida: Como se ha explicado anteriormente, los alimentos son las comidas que el usuario ha consumido, no necesariamente los mismos que sugiere la dieta para un día en concreto. La relación es de muchos (opcional) a muchos (opcional).