



KCALCONTROL

Ciclo Formativo de Grado Superior
Desarrollo de Aplicaciones Web

2024-2025

AUTOR: Yoannet Díaz Valdés

TUTOR: Paula Carolina Martínez Perea

RESUMEN

Este proyecto tiene como objetivo el desarrollo de una aplicación web orientada a la gestión de la nutrición de los usuarios, permitiéndoles llevar un control de su ingesta calórica. La aplicación cuenta con un sistema de autenticación de usuarios utilizando Firebase, que permite registrar, iniciar sesión y mantener la sesión activa mediante tokens de acceso, asegurando una experiencia de usuario fluida. Además, se ha implementado una API RESTful en Laravel para gestionar los datos de los perfiles de cada usuarios.

La aplicación permite a los usuarios calcular sus necesidades diarias de macronutrientes (proteínas, carbohidratos y grasas) en base a su edad, peso, sexo y objetivo personal, utilizando la fórmula de Mifflin-St Jeor. Para ello se ha diseñado una interfaz para que los usuarios puedan añadir, eliminar y editar alimentos en las tres comidas principales del día (desayuno, almuerzo y cena). Además, se incluye un indicador visual que muestra las calorías consumidas y las que faltan para alcanzar el objetivo diario, facilitando el seguimiento del progreso.

Esta aplicación web integra, además, una API externa que permite buscar y añadir alimentos comunes, proporcionando información sobre los macronutrientes y las calorías de cada alimento. También se ha desarrollado una funcionalidad para almacenar el historial de comidas diarias de los usuarios, con una vista que permite consultar y filtrar las ingestas pasadas por fecha. Estas características permiten a los usuarios tener un control sobre su nutrición contribuyendo al objetivo que quieren alcanzar.

ABSTRACT

This project aims to develop a web application focused on managing users' nutrition, allowing them to track their caloric intake. The application features a user authentication system using Firebase, enabling users to register, log in, and maintain an active session through access tokens, ensuring a smooth user experience. Additionally, a RESTful API has been implemented in Laravel to manage the profile data of each user.

The application allows users to calculate their daily macronutrient needs (proteins, carbohydrates, and fats) based on their age, weight, gender, and personal goal, using the Mifflin-St Jeor formula. To achieve this, an interface has been designed where users can add, remove, and edit foods in the three main meals of the day (breakfast, lunch, and dinner). Moreover, a visual indicator has been included to show the calories consumed and the remaining calories needed to reach the daily goal, facilitating progress tracking.

Furthermore, this web application integrates an external API that enables users to search and add common foods, providing information on the macronutrients and calories of each food item. A feature has also been developed to store the daily meal history of users, with a view that allows them to consult and filter past intakes by date. These features give users control over their nutrition, helping them achieve their desired goals.

RESUM

Aquest projecte té com a objectiu el desenvolupament d'una aplicació web orientada a la gestió de la nutrició dels usuaris, permetent-los portar un control de la seua ingesta calòrica. L'aplicació compta amb un sistema d'autenticació d'usuaris utilitzant Firebase, que permet registrar-se, iniciar sessió i mantenir la sessió activa mitjançant tokens d'accés, assegurant una experiència d'usuari fluida. A més, s'ha implementat una API RESTful en Laravel per gestionar les dades dels perfils de cada usuari.

L'aplicació permet als usuaris calcular les seues necessitats diàries de macronutrients (proteïnes, carbohidrats i greixos) en funció de la seua edat, pes, sexe i objectiu personal, utilitzant la fórmula de Mifflin-St Jeor. Per a això s'ha dissenyat una interfície perquè els usuaris puguin afegir, eliminar i editar aliments en les tres menjades principals del dia (esmorzar, dinar i sopar). A més, s'inclou un indicador visual que mostra les calories consumides i les que falten per aconseguir l'objectiu diari, facilitant el seguiment del progrés.

Aquesta aplicació web integra, a més, una API externa que permet cercar i afegir aliments comuns, proporcionant informació sobre els macronutrients i les calories de cada aliment. També s'ha desenvolupat una funcionalitat per emmagatzemar l'historial de menjars diaris dels usuaris, amb una vista que permet consultar i filtrar les ingestions passades per data. Aquestes característiques permeten als usuaris tenir un control sobre la seua nutrició, contribuint a l'objectiu que volen aconseguir.

ÍNDICE

RESUMEN	1
ABSTRACT	2
RESUM	3
JUSTIFICACIÓN	5
OBJETIVOS	6
PLANIFICACIÓN DEL PROYECTO	7
Propuesta plan de trabajo inicial	7
Diagrama de Gantt del proyecto	8
Subtareas	9
HERRAMIENTAS UTILIZADAS	11
DESCRIPCIÓN DEL PROYECTO.....	12
Análisis de funcionalidades claves	12
Autenticación de usuarios	12
Distribución de macronutrientes.....	12
Diseño	16
Implementación.....	20
Pruebas	21
Documentación	21
AMPLIACIONES FUTURAS	22
CONCLUSIONES	23
BIBLIOGRAFÍA Y WEBGRAFÍA.....	24
ANEXOS	26

JUSTIFICACIÓN

Este proyecto surge con el motivo principal de crear una herramienta de control de calorías dirigida a personas que están en un proceso de transformación física, ya sea para perder peso, ganar masa muscular o mejorar su salud en general. Durante este tipo de procesos, es fundamental tener un control de la ingesta calórica así como los macronutrientes, para la consecución de objetivos físicos. Sin embargo, muchas veces las personas carecen de los conocimientos necesarios para administrar correctamente sus ingestas, lo que hace de esto, un proceso un poco engorroso. Y es aquí donde kcalcontrol ofrece una solución sencilla, accesible y personalizada

Para el sistema de cómputo calórico y macronutrientes se ha empleado la fórmula de Mifflin-St Jeor, que es una de las fórmulas más aceptadas y precisas para estimar la Tasa Metabólica Basal (BMR). La misma fue publicada en 1990 por **Mifflin y St Jeor** en un estudio titulado: *"A new predictive equation for resting energy expenditure in healthy individuals"*.

OBJETIVOS

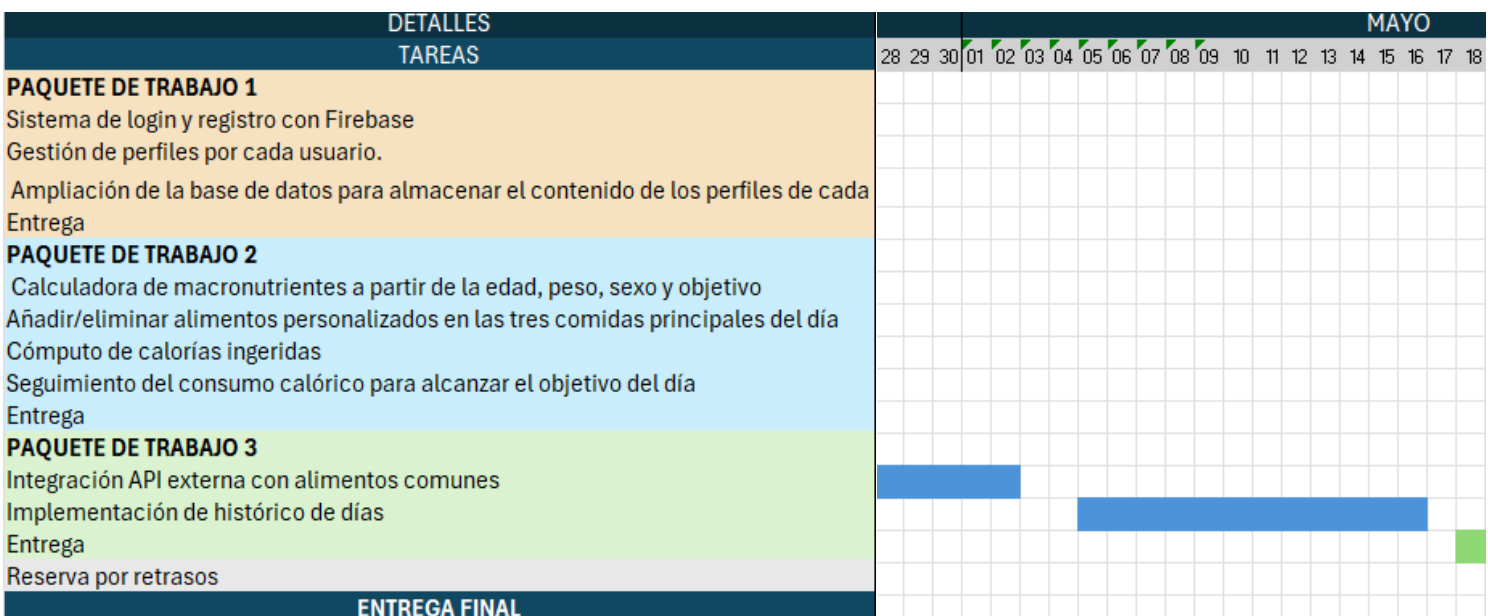
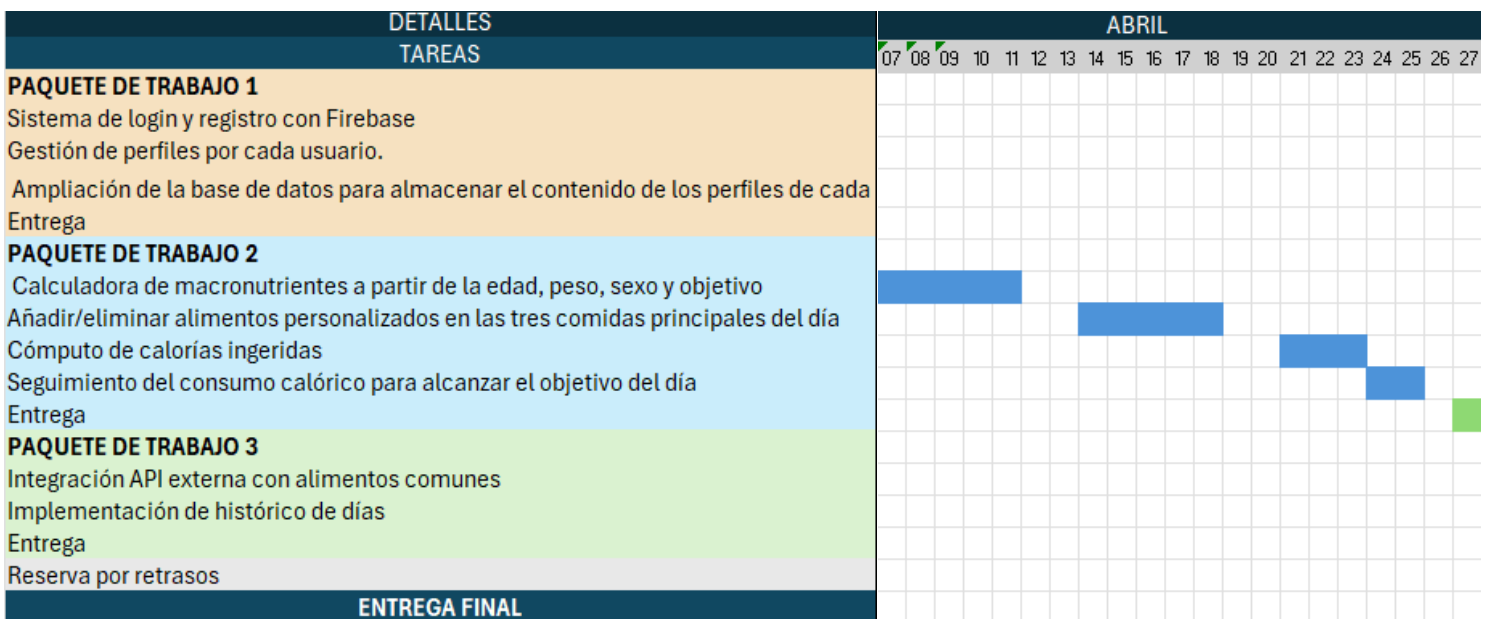
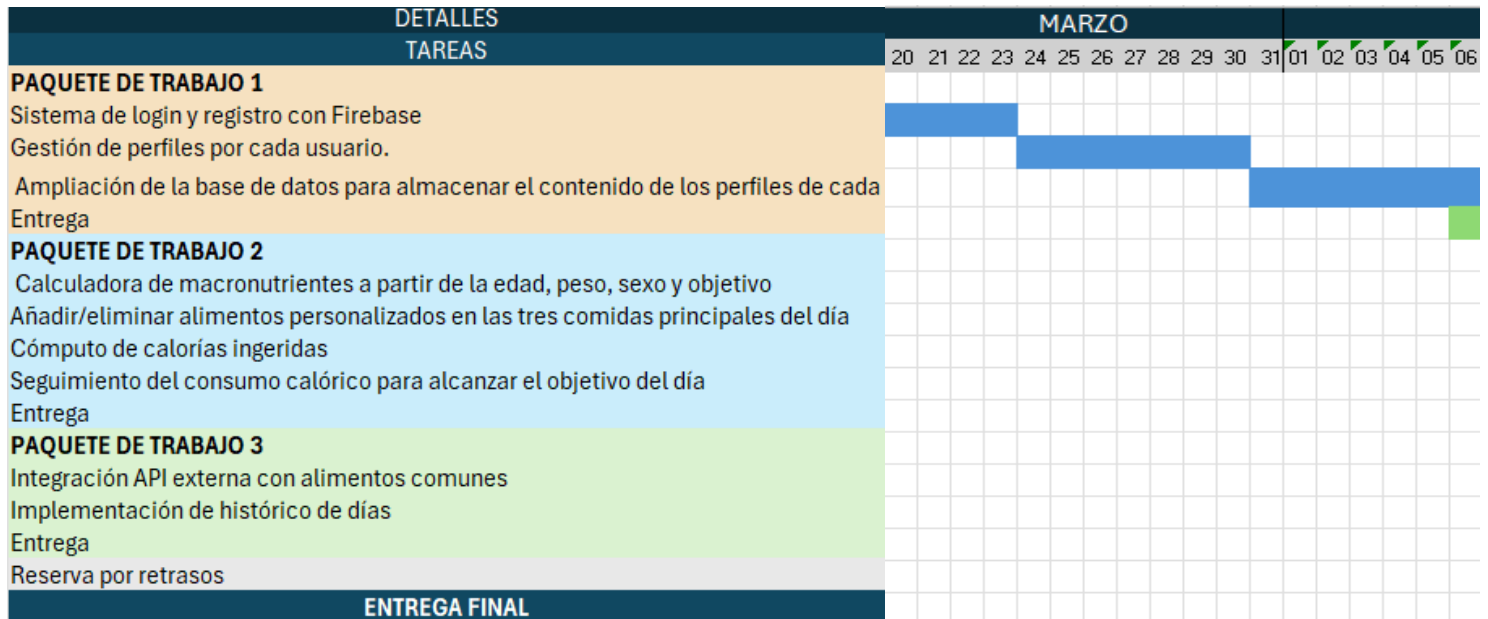
1. **Facilitar el acceso y gestión de cuentas de usuario:** El objetivo principal es permitir que los usuarios creen y gestionen sus cuentas proporcionando funcionalidades de registro, inicio de sesión y cierre de sesión.
2. **Personalizar la experiencia del usuario:** La aplicación permitirá que cada usuario pueda crear y gestionar su propio perfil, almacenando y actualizando información básica, así como datos relacionados con su progreso y objetivos.
3. **Calcular las necesidades nutricionales individuales:** La aplicación permitirá a los usuarios calcular sus necesidades diarias de macronutrientes (proteínas, carbohidratos y grasas) en función de su edad, peso, sexo y objetivo personal.
4. **Realizar un seguimiento de las calorías ingeridas:** La aplicación proporcionará a los usuarios una vista clara de las calorías consumidas en cada comida y el total del día. Esto les permitirá hacer ajustes en su dieta en tiempo real para asegurarse de que cumplen con sus objetivos de ingesta calórica.
5. **Consultar el historial de ingestas diarias:** Los usuarios podrán filtrar por fecha para ver cómo ha sido su evolución a lo largo del tiempo, brindándoles un panorama claro de su trayectoria nutricional.

PLANIFICACIÓN DEL PROYECTO

Propuesta plan de trabajo inicial

Tareas	Fecha de inicio	Fecha de fin
Paquete de Trabajo 1		
PT1.1. Sistema de login y registro con Firebase	20/03/2025	23/03/2025
PT1.2. Gestión de perfiles por cada usuario.	24/03/2025	30/03/2025
PT1.3. Ampliación de la base de datos para almacenar el contenido de los perfiles de cada usuario.	31/03/2025	06/04/2025
Entrega parcial		06/04/2025
Paquete de Trabajo 2		
PT2.1. Calculadora de macronutrientes a partir de la edad, peso, sexo y objetivo	07/04/2025	11/04/2025
PT2.2. Añadir/eliminar alimentos personalizados en las tres comidas principales del día	14/04/2025	18/04/2025
PT2.3. Cómputo de calorías ingeridas	21/04/2025	23/04/2025
PT2.4. Seguimiento del consumo calórico para alcanzar el objetivo del día	24/04/2025	25/04/2025
Entrega parcial		25/04/2025
Paquete de Trabajo 3		
PT3.1. Integración API externa con alimentos comunes	28/04/2025	02/05/2025
PT3.2. Implementación de histórico de días	05/05/2025	16/05/2025
Entrega parcial		16/05/2025
Reserva por atrasos	16/05/2025	29/05/2025
Entrega Final		30/05/2025

Diagrama de Gantt del proyecto



Subtareas

Paquete de trabajo 1

PT1.1. Sistema de login y registro con Firebase

- Configurar un proyecto en Firebase.
- Implementar la autenticación de usuario con correo electrónico y contraseña.
- Diseño e implementación de la página de Login y Registro
- Manejar el cierre de sesión.
- Gestionar los tokens de acceso para mantener la sesión activa.

PT1.2. Gestión de perfiles de usuarios.

- Implementar una API RESTful en Laravel con la base una base de datos en MySQL para gestionar los datos básicos de los usuarios.
- Crear rutas y controladores en Laravel para las operaciones CRUD (crear, leer, actualizar, eliminar).
- Integrar la API de Laravel con Firebase para autenticar usuarios.

PT1.3. Ampliación de la base de datos para almacenar el contenido de los perfiles de cada usuario.

- Definir las tablas necesarias para almacenar la información del usuario.
- Crear un esquema relacional de base de datos.
- Implementar migraciones y seeders en Laravel para la creación y población inicial de la base de datos.
- Crear procedimientos almacenados para optimizar consultas específicas.

Paquete de trabajo 2

PT2.1. Calculadora de macronutrientes a partir de la edad, peso, sexo y objetivo

- Definir la fórmula para el cálculo de macronutrientes (por ejemplo, fórmula de Mifflin-St Jeor).
- Diseño e implementación de la interfaz para recibir los datos del usuario (edad, peso, sexo, objetivo).
- Crear una lógica para calcular los macronutrientes (proteínas, carbohidratos y grasas).

PT2.2. Añadir/eliminar alimentos personalizados en las tres comidas principales del día

- Diseñar formularios de entrada para que los usuarios puedan agregar alimentos de forma personalizada.
- Implementar lógica para añadir, eliminar, editar alimentos personalizados en la base de datos.

- Crear una interfaz para que los usuarios puedan gestionar los alimentos por cada comida (desayuno, almuerzo, cena).

PT2.3. Cómputo de calorías ingeridas

- Diseñar una lógica para calcular las calorías totales consumidas basadas en los alimentos ingresados.
- Mostrar el total de calorías ingeridas para cada comida y el total del día.

PT2.4. Seguimiento del consumo calórico para alcanzar el objetivo del día

- Implementar indicador visual que muestre las calorías ingeridas y las que le faltan para alcanzar el objetivo.

Paquete de trabajo 3

PT3.1. Integración API externa con alimentos comunes

- Seleccionar una API externa de alimentos (por ejemplo, la API de MyFitnessPal, Nutritionix).
- Integrar la API con la aplicación para obtener información sobre alimentos comunes.
- Crear interfaz que permita al usuario buscar alimentos comunes y añadirlos a sus comidas.
- Mostrar información detallada de los alimentos (calorías, macronutrientes, etc.) al usuario.

PT3.2. Implementación de histórico de días

- Crear una tabla para almacenar el historial de comidas diarias de los usuarios.
- Implementar una vista que permita al usuario ver un historial de días pasados.
- Añadir funcionalidad para filtrar por fecha.

HERRAMIENTAS UTILIZADAS

Para la realización de este proyecto se emplearon las siguientes tecnologías:

React: Utilizado para el desarrollo del frontend, proporcionando una interfaz de usuario interactiva y dinámica aportando una experiencia fluida y reactiva, ya que permite actualizar la interfaz en tiempo real sin recargar la página.

Firebase: Utilizado para la gestión de la autenticación y gestión de usuarios. Con Firebase, fue posible gestionar el registro, inicio y cierre de sesión de los usuarios.

Firebase Authentication ofrece una solución sencilla y escalable para gestionar el inicio de sesión con diversos métodos, como Google, Facebook, correo electrónico y contraseñas, etc. Además se encarga de la gestión de contraseñas de forma segura y provee características como verificación de correo, restablecimiento de contraseñas. En cuanto a la escalabilidad es automática y se maneja correctamente en situaciones de alto tráfico.

Axios: Librería para la gestión de solicitudes HTTP desde React hacia la API.

Laravel: Este framework de PHP, permitió crear una API RESTful que gestionara los datos de los usuarios permitiendo realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre los datos almacenados de los usuarios. Laravel proporcionó una estructura robusta para gestionar la lógica de negocio y la manipulación de datos en el backend.

MySQL: Se empleó como sistema de gestión de bases de datos, proporcionando una solución relacional para almacenar los perfiles de los usuarios, los alimentos personalizados y el historial de ingestas diarias. MySQL garantizó una base de datos sólida y escalable.

Tailwind CSS: Se utilizó para el diseño de la interfaz de usuario, brindando un enfoque de desarrollo ágil mediante clases utilitarias que permitieron diseñar rápidamente una interfaz moderna, flexible y adaptativa para todos los dispositivos, mejorando la accesibilidad y la experiencia del usuario.

Estas tecnologías trabajaron en conjunto para ofrecer una solución escalable, con un enfoque en la facilidad de uso por parte de los usuarios.

DESCRIPCIÓN DEL PROYECTO

Análisis de funcionalidades claves

La aplicación ofrece una serie de funcionalidades clave que responden a las necesidades del usuario las cuales se describirán a continuación

Autenticación de usuarios

La autenticación de usuarios es una parte fundamental en el control de acceso a aplicaciones web. En este proyecto, se implementó el proceso de autenticación utilizando Firebase Authentication, un servicio proporcionado por Google que facilita la gestión segura de usuarios mediante diferentes métodos de acceso.

Firebase cuenta con varios métodos de autenticación, este proyecto utiliza:

- Autenticación mediante usuario y contraseña
- Autenticación mediante una cuenta de Google

Cuando un usuario desea crear una cuenta, proporciona su correo electrónico y una contraseña segura. Esta información se envía a Firebase utilizando el método:

```
firebase.auth().createUserWithEmailAndPassword(email, password)
```

Firebase gestiona automáticamente el almacenamiento seguro de las credenciales, La verificación del formato del correo y la verificación del formato del correo.

Una vez registrado el usuario, para poder acceder, este insertará su correo y contraseña y la aplicación envía esta información a Firebase a través de:

```
firebase.auth().signInWithEmailAndPassword(email, password)
```

Firebase valida las credenciales y, si son correctas, genera un **token de sesión (JWT)** que representa al usuario autenticado, de igual forma maneja de forma automática la persistencia de la sesión, manteniendo al usuario autenticado entre recargas de página o cierres de aplicación, hasta que este decida cerrar sesión explícitamente.

Con un sistema de autenticación gestionado por Firebase se obtienen las siguientes ventajas:

- Seguridad gestionada y actualizada por Google.
- Soporte para múltiples proveedores de identidad.
- Gestión automática de sesiones y tokens.

Distribución de macronutrientes

Para estimar la cantidad de calorías necesarias por día y posterior distribución en macronutrientes se sigue un proceso que se inicia con el cómputo de la Tasa Metabólica Basal (BMR) y continúa con ajustes en dependencia del nivel de actividad que tenga el usuario y el objetivo que pretende alcanzar.

1. Cálculo de BMR

El BMR (Tasa Metabólica Basal) es la cantidad mínima de energía (en forma de calorías) que el cuerpo necesita para llevar a cabo funciones vitales en reposo, como la respiración, la circulación sanguínea, el control de la temperatura corporal, la digestión y el funcionamiento del sistema nervioso [1]

Existen varias fórmulas reconocidas para calcular este parámetro, entre las cuales se incluyen:

- Ecuación de Harris-Benedict
- Ecuación de Katch-McArdle
- Ecuación de ten Haaf
- Ecuación de Cunningham

De estas cuatro ecuaciones predictivas, según el artículo *"Comparison of Predictive Equations for Resting Metabolic Rate in Healthy Nonobese and Obese Adults: A Systematic Review"*, la ecuación de Mifflin-St Jeor es la que predice con mayor precisión la tasa metabólica en reposo, con un margen de error del 10% respecto a las mediciones realizadas por calorimetría indirecta, tanto en individuos no obesos como obesos. Este valor es superior al de cualquier otra ecuación y también presentó la tasa de error más baja [2]

Por tal motivo, se ha decidido utilizar esta ecuación para el cómputo del BMR en este contexto. No obstante, si se busca aún más precisión, la ecuación de Mifflin-St Jeor resulta especialmente adecuada para individuos con un Índice de masa corporal

(IMC) normal (es decir, con un índice entre 19 y 25), cuando no es necesario conocer el porcentaje de grasa corporal.

$$IMC = \frac{\text{peso}(Kg)}{\text{altura}(m)^2}$$

El BMR se determina teniendo en cuenta el sexo de la persona:

Sexo femenino:

$$BMR = (10 \times \text{peso en kg}) + (6.25 \times \text{altura en cm}) - (5.0 \times \text{edad en años}) - 161$$

Sexo masculino:

$$BMR = (10 \times \text{peso en kg}) + (6.25 \times \text{altura en cm}) - (5.0 \times \text{edad en años}) + 5$$

2. Ajuste según el factor de actividad adecuado. Es lo que se conoce como Gasto Energético Diario Total (TDEE)

Sedentario: Poco o ningún ejercicio, trabajo de oficina

$$TDEE = BMR \times 1.2$$

Ligeramente activo: Ejercicio físico de 1 a 3 días por semana.

$$TDEE = BMR \times 1.375$$

Moderadamente activo: Ejercicio físico de 3 a 5 días por semana.

$$TDEE = BMR \times 1.55$$

Muy activo: Ejercicio físico de 6 a 7 días por semana

$$TDEE = BMR \times 1.725$$

Extremadamente activo: Ejercicio físico dos veces al día.

$$TDEE = BMR \times 1.9$$

Este enfoque es comúnmente aceptado en la literatura de nutrición, aunque los factores exactos pueden variar ligeramente según la fuente.

Es importante tener en cuenta que estas ecuaciones son sólo una estimación y, por lo tanto, deben tenerse en cuenta las necesidades y objetivos nutricionales individuales.

Debe evaluar y reevaluar, si es necesario, los progresos del cliente y cómo se siente y, a partir de allí, hacer ajustes en el plan dietético.

3. Ajuste según objetivo [3]

Ganar masa muscular: Añadir 250-500 calorías a tu TDEE.

Perder peso: Restar 250-500 calorías a tu TDEE.

Mantener el peso: Consumir tu TDEE calculado.

4. Distribución de macronutrientes

A partir de la información consultada en el sitio web [4], se pueden recomendar las siguientes cantidades de macronutrientes:

Proteínas: Se recomienda consumir entre **1.6 y 2.2 gramos por kilogramo de peso corporal** al día para deportes de fuerza, y entre **1.2 y 1.6 gramos por kilogramo de peso corporal** al día para deportes de resistencia.

Carbohidratos: Para deportistas de resistencia, se sugieren entre **6 y 10 gramos por kilogramo de peso corporal** al día, mientras que para deportes de fuerza, la recomendación es entre **4 y 7 gramos por kilogramo de peso corporal** al día.

Grasas: Se recomienda consumir entre **1 y 1.5 gramos por kilogramo de peso corporal** al día o, en su defecto, entre el **20% y 35%** de las calorías totales diarias.

En KiloCalControl para estimar la distribución de las cantidades de macronutrientes se ha elegido el valor medio dentro de los rangos recomendados para cada macronutriente, con el objetivo de proporcionar una guía equilibrada y accesible para la mayoría de los usuarios. Luego los valores establecidos serían:

Proteínas: 1.9 gramos por kilogramo de peso corporal

Carbohidratos: 7 gramos por kilogramo de peso corporal

Grasas: 1.25 gramos por kilogramo de peso corporal

Gestión de alimentos

Esta sección de la plataforma permite a los usuarios agregar, editar y eliminar alimentos dentro de una base de datos personalizada, adaptada a sus necesidades específicas.

- Cada alimento registrado contiene atributos relevantes como:
- Descripción del alimento.
- Cantidad de calorías basada en la unidad o porción especificada.
- Valores nutricionales como grasas, proteínas y carbohidratos.

Adicionalmente, por cada alimento creado se almacena el ID único del usuario (UID), lo que permite mantener un control individualizado de los alimentos asociados a cada cuenta.

Esta funcionalidad garantiza que cada usuario pueda construir y gestionar su propio catálogo de alimentos, sin interferir en los registros de otros usuarios.

Control de calorías

El sistema calcula automáticamente el total de calorías ingeridas por el usuario a lo largo del día en función de los alimentos registrados. Al seleccionar un alimento y especificar la cantidad consumida, el programa multiplica esta cantidad por el valor calórico correspondiente y actualiza el total diario. Se incluyen indicadores para mostrar si el usuario ha alcanzado el objetivo de calorías establecido previamente, contribuyendo así a un mejor control nutricional y a la promoción de hábitos alimenticios saludables.

Diseño

Diseño de la arquitectura

La arquitectura de la aplicación web sigue un modelo cliente-servidor clásico, donde el frontend desarrollado en React se comunica con una API REST construida en Laravel, la cual a su vez interactúa con una base de datos relacional MySQL.

La aplicación se divide en dos partes principales:

❖ Frontend (React)

Se encarga de la presentación de la información y de la interacción con el usuario. Gestiona el enrutamiento, los formularios, la visualización de alimentos y gestión de los mismos por ingesta.

components/: Componentes de la interfaz como tarjetas de alimentos, formularios de ingreso, y botones de acción.

hooks/: Custom hooks que encapsulan la lógica de acceso a los contextos y simplifican su uso en los componentes.

pages/: Páginas principales como Inicio, Registro de alimentos, Historial de ingestas, etc.

context/: Gestión de los estados globales como la sesión del usuario.

❖ Backend (Laravel API)

Expone endpoints REST que permiten realizar operaciones de gestión de información de los usuarios, gestión de alimentos y registro de ingestas diarias.

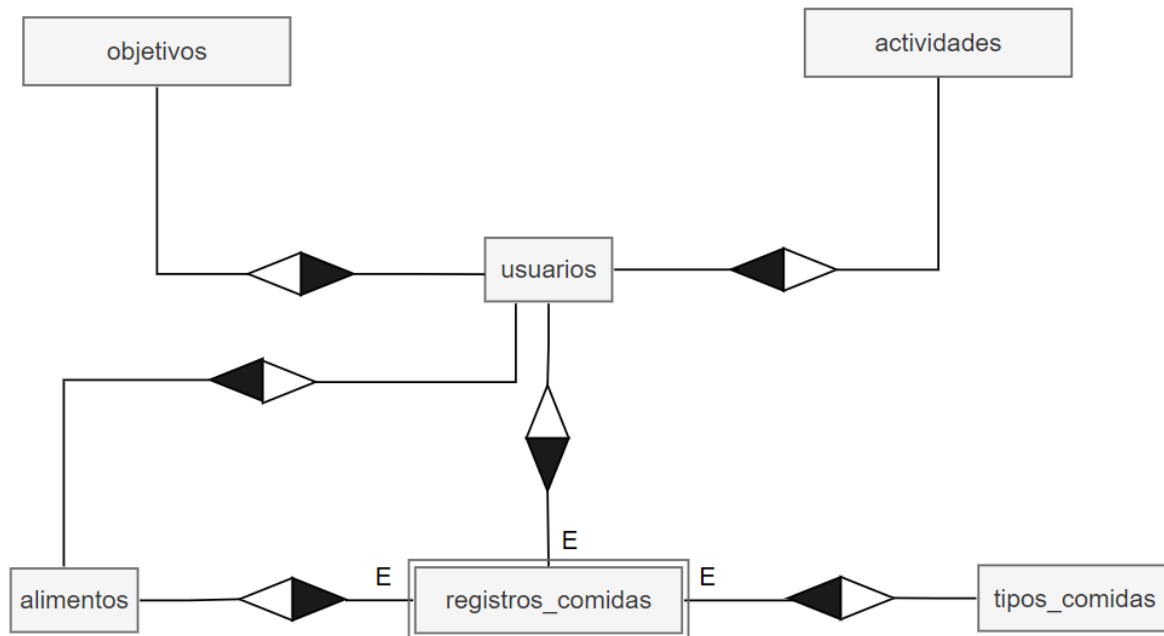
app/Http/Controllers/: Controladores que gestionan las solicitudes del frontend.

routes/api.php: Archivo donde se definen las rutas de la API.

database/migrations/: Definición de las tablas de la base de datos.

app/Models/: Modelos Eloquent para interactuar con MySQL.

Diagrama entidad-relación de la base de datos



Diseño lógico

❖ Tabla: Usuarios

Campo	Tipo de dato	Descripción
uid	VARCHAR	Identificador único del usuario (clave primaria)
nombre	VARCHAR	Nombre del usuario
apellidos	VARCHAR	Apellidos del usuario
edad	INT	Edad del usuario
sexo	CHAR(1)	Sexo del usuario (por ejemplo, 'M' o 'F')
objetivo	INT	Tipo de objetivo
actividad	INT	Nivel de actividad (representado como un valor categórico)
obj_calorias	DECIMAL	Calorías diarias objetivo
obj_proteinas	DECIMAL	Gramos de proteínas diarios objetivo
obj_grasas	DECIMAL	Gramos de grasas diarios objetivo
obj_carbohidratos	DECIMAL	Gramos de carbohidratos diarios objetivo

Clave primaria: { uid }

Clave foránea : { objetivo } hace referencia a **objetivos**

Clave foránea : { actividad } hace referencia a **actividades**

❖ **Tabla: tipos_comidas**

Campo	Tipo de dato	Descripción
id	INT	Identificador único del tipo de comida (clave primaria)
descripción	VARCHAR	Nombre de la comida

Clave primaria: { id }

❖ **Tabla: Objetivos**

Campo	Tipo de dato	Descripción
id	INT	Identificador único del tipo de objetivo (clave primaria)
descripción	VARCHAR	Descripción del objetivo

Clave primaria: { id }

❖ **Tabla: Actividad**

Campo	Tipo de dato	Descripción
id	INT	Identificador único del tipo de actividad (clave primaria)
descripción	VARCHAR	Descripción de la actividad

Clave primaria: { id }

❖ **Tabla: Alimentos**

Campo	Tipo de dato	Descripción
id	VARCHAR	Identificador único del alimento (clave primaria)
uid	VARCHAR	Identificador del usuario
base	DECIMAL	Base en gramos para la cual se indican las cantidades macros
calorias	INT	Calorías contenidas en la base
proteinas	DECIMAL	Proteínas contenidas en la base
grasas	DECIMAL	Grasas contenidas en la base
carbohidratos		Carbohidratos contenidos en la base

Clave primaria: { id }

Clave foránea : { uid } hace referencia a **usuarios**

❖ **Tabla: Registros**

Campo	Tipo de dato	Descripción
id	VARCHAR	Identificador único del registro (clave primaria)
Uid	VARCHAR	Identificador del usuario
fecha	date	Fecha de registro
tipo_comida_id	INT	Tipo de comida
alimento_id	INT	Identificador del alimento
cantidad	DECIMAL	Cantidad de alimento en gramos

Clave primaria: { id }

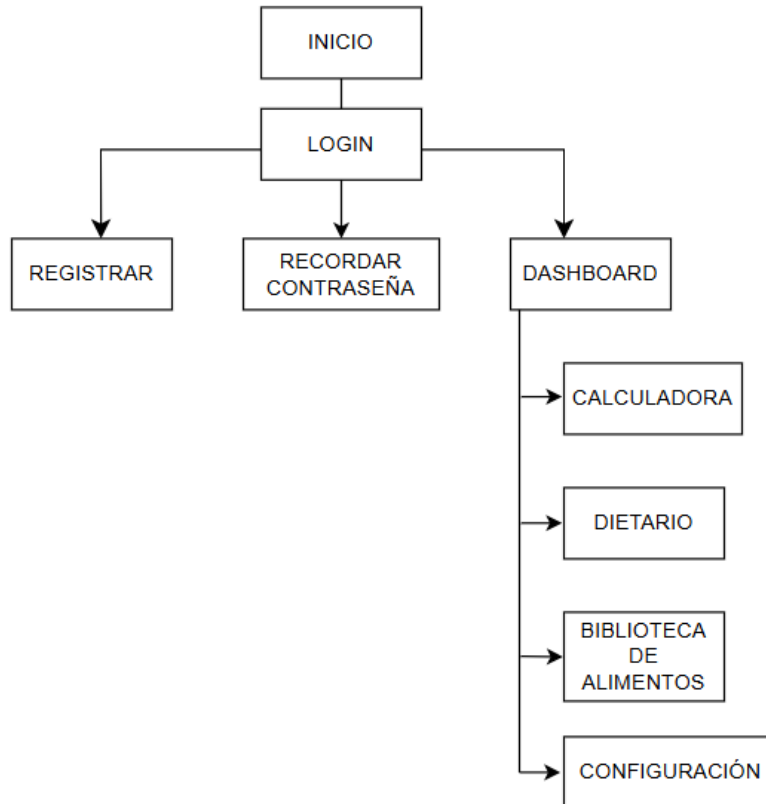
Clave foránea : { uid } hace referencia a **usuarios**

Clave foránea : { tipo_comida_id } hace referencia a **tipos_comidas**

Clave foránea : { alimento_id } hace referencia a **alimentos**

Valor único : { uid, fecha , tipo_comida_id, alimento_id }

Diagrama de navegación de la web



Flujo de trabajo dentro de la web

1. El usuario inicia sesión o se registra si aún no tiene una cuenta. El inicio de sesión puede realizarse mediante usuario y contraseña, o a través de su cuenta de Google.
2. Una vez dentro, para comenzar a utilizar la aplicación, debe introducir los datos necesarios para calcular las calorías que debe consumir diariamente: edad, peso, altura, sexo, nivel de actividad y objetivo deseado.
3. Con estos datos, el sistema permitirá al usuario registrar las comidas diarias, organizadas por grupos: desayuno, comida, cena y merienda.
4. El usuario interactúa con la interfaz desarrollada en React, realizando acciones como agregar alimentos o registrar ingestas. Este proceso puede hacerse de forma manual, introduciendo las macronutrientes del alimento, o automáticamente, indicando el código de barras del producto.
5. React utiliza Axios para enviar solicitudes HTTP (POST, GET, PUT o DELETE) a la API desarrollada en Laravel.
6. La API recibe la solicitud, valida los datos y realiza las operaciones correspondientes sobre la base de datos MySQL.

7. Laravel responde al frontend con los datos actualizados o con el estado de la operación solicitada.
8. React actualiza la interfaz en función de las respuestas del backend.
9. El estado de la aplicación se gestiona de forma global mediante Context y se consume en los componentes a través de custom hooks como useAuthContext, useAlimentosContext o useComputoContext, lo que favorece un flujo de datos claro, reutilizable y mantenible.

Implementación

Pruebas

Documentación

AMPLIACIONES FUTURAS

Mejoras propuestas para la aplicación:

1. **Sistema de notificaciones:** Alertar sobre el progreso de sus objetivos. Esto mejoraría la adherencia a los objetivos y la motivación del usuario, manteniéndolos enfocados en su proceso de transformación.
2. **Menús completos preestablecidos:** Ofrecer menús prediseñados según los objetivos nutricionales (pérdida de peso, ganancia muscular, mantenimiento) para facilitar la planificación de las comidas. Los usuarios podrán agregar menús completos a las comidas principales del día.
3. **Selección de ecuación para el cómputo de macronutrientes:** Permitir a los usuarios elegir entre diferentes ecuaciones científicas (como Mifflin-St Jeor, Harris-Benedict, Katch-McArdle) para calcular sus necesidades de macronutrientes, adaptándose a sus preferencias y datos personales. Esto ofrece mayor flexibilidad y precisión en los cálculos, ayudando a los usuarios a obtener resultados más adecuados a su perfil y objetivos.
4. **Posibilidad de añadir más comidas (meriendas, snack, etc.):** Ampliar la funcionalidad para permitir a los usuarios agregar más comidas a lo largo del día, como meriendas o snacks, además de las tres comidas principales (desayuno, comida y cena). Esto brindará a los usuarios una mayor flexibilidad para organizar su ingesta calórica según sus necesidades diarias.

CONCLUSIONES

La aplicación desarrollada cumple con el objetivo de proporcionar una herramienta accesible y fácil de usar para el control de la nutrición personal. Facilita el acceso y gestión de cuentas de usuario, permitiendo un registro y acceso sencillo para que los usuarios puedan crear y administrar sus perfiles de manera segura. Además, personaliza la experiencia de cada usuario, almacenando y actualizando información relevante sobre sus datos y objetivos, lo que fomenta un enfoque individualizado en su proceso de transformación.

BIBLIOGRAFÍA Y WEBGRAFÍA

- [1] M. Abreu, «Ecuación Mifflin-St. Jeor para profesionales de la nutrición,» 3 mayo 2023. [En línea]. Available: https://nutrium.com/blog/es/ecuacion-mifflin-st-jeor-para-profesionales-de-la-nutricion/?utm_source=chatgpt.com.
- [2] D. Frankenfield, L. Roth-Yousey y C. Compher, «Comparison of predictive equations for resting metabolic rate in healthy nonobese and obese adults: a systematic review,» *Journal of the Academic of Nutrition and Dietetics*, pp. 775-789, 2005.
- [3] M. Hospitals, «Optimice su dieta: Calculadora de ingesta calórica,» [En línea]. Available: https://www.medicoverhospitals.in/es/fitness-health-calculators/calorie-intake-calculator?utm_source=chatgpt.com.
- [4] Besfor, «Cómo calcular las cantidades ideales de macronutrientes según tu nivel de actividad física?,» Noviembre 2024. [En línea]. Available: <https://besfor.com/calcular-macronutrientes-segun-actividad-fisica/>.
- [5] M. D. Mifflin, S. T. S. Jeor, L. A. Hill, S. A. Daugherty y Y. O. Koh, «A new predictive equation for resting energy expenditure in healthy individuals,» *ScienceDirect*, pp. 241-247, 1990.
- [6] onlinetoolkit, «Calculadora Mifflin-St Jeor: Tasa Metabólica Basal,» 19 Septiembre 2024. [En línea]. Available: https://onlinetoolkit.co/es/calculadora-mifflin-st-jeor-tasa-metabolica-basal/?utm_source=chatgpt.com.
- [7] «Laravel Documentation,» [En línea]. Available: <https://laravel.com/docs/11.x/readme>.
- [8] «Learn React,» [En línea]. Available: <https://react.dev/learn>.
- [9] «Get started with Tailwind CSS,» [En línea]. Available: <https://tailwindcss.com/docs/installation/using-vite>.
- [10] «Developer documentation for Firebase,» [En línea]. Available: <https://firebase.google.com/docs>.

ANEXOS