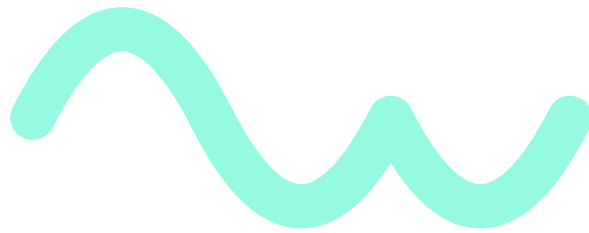


Proyecto Grado Desarrollo de Aplicaciones Web

WellnesTrack



Realizado por:
Daniel Martín Hermoso Hermoso

Tabla de contenido

1. Introducción	4
1.1. Propósito	4
1.2. Objetivos	4
1.3. Antecedentes (estado del arte)	5
1.4. Viabilidad técnica y económica del proyecto	5
Recursos hardware y software necesarios:	5
1.5. Temporalización	6
2. Análisis	7
2.1. Documentación relevante	7
2.2. Definiciones	7
2.3. Requisitos funcionales y no funcionales	7
2.4. Casos de Uso. Diagramas UML de casos de uso	9
2.5. Modelado E/R. Diagramas E/R.	10
2.6. Sketching de la interfaz.....	11
Sketching	11
Wireframing	22
3. Diseño.....	34
3.1 Arquitectura hardware y software de la solución	34
3.2 Modelado funcional de la solución. Diagramas de clase	34
3.3 Modelado de datos. Modelo relacional. Diccionario de Datos	35
3.4 Prototipado de la interfaz	36
4. Implementación	48
4.1 Requisitos de instalación y ejecución	48
4.2 Implementación funcional. Clases.....	48
4.3 Implementación del modelo de datos. Tablas.....	50
5. Pruebas	51
5.1. Pruebas de módulos	51
Prueba numero 1:	51
Prueba numero 2:	51
5.2. Pruebas de integración	52
5.3 Pruebas del sistema.....	53
5.4 Pruebas de instalación.....	54
6. Conclusiones	55
6.1 Grado de consecución de los objetivos inicialmente planteados	55

6.2 Dificultades encontradas	55
6.3 Propuestas de mejora y posibles ampliaciones	55
7. Bibliografía y recursos on-line	56
• Symphony	56
• Angular	56
• Firebase	56
• Base de datos	56
• Docker	56
8. Glosario de términos	57
8.1 Informáticos	57
8.2 Problema	57
9. Anexos	58

1. Introducción

Mi proyecto se centra en desarrollar una web dedicada a poder realizar un seguimiento detallado de la alimentación y actividad física que el usuario lleve en su día a día.

1.1. Propósito

En la sociedad actual, la conciencia sobre la importancia de una alimentación saludable y la actividad física regular está en aumento. Sin embargo, muchos individuos encuentran dificultades para mantener un seguimiento preciso y sistemático de sus hábitos alimenticios y de ejercicio, lo que puede dificultar el logro de sus objetivos de bienestar personal

El proyecto responde a la necesidad de una herramienta integral y accesible que permita a los usuarios llevar un registro detallado y personalizado de su ingesta de alimentos y su actividad física. Al proporcionar una plataforma fácil de usar y altamente funcional, se busca facilitar el proceso de seguimiento de la salud y el bienestar de los usuarios

El proyecto está enfocado en los usuarios que deseen llevar un seguimiento detallado de su ingesta de alimentos, actividad física y progreso de peso.

En una hipotética extensión se podría implementar un apartado en el que los datos se lleven de forma conjunta con una persona experta en el tema ya sea un nutricionista con la posibilidad de usar la web también como una guía con la dieta que seguir o con un entrenador personal para saber que ejercicios realizar.

Fuera del alcance del proyecto quedarían aspectos médicos detallados que cada usuario pueda llegar a necesitar.

1.2. Objetivos

Las metas que se tienen como objetivo no son más que utilizar las herramientas y tecnologías usadas a lo largo de los módulos cursados para demostrar la capacidad de desarrollo y resolución de problemas que tengo como programador, así como el uso e investigación de nuevas tecnologías que se podrían llegar a usar.

Todo esto enfocado en el desarrollo de un proyecto que de verdad tiene un uso real y que soluciona un problema existente en la sociedad actual.

1.3. Antecedentes (estado del arte)

La web más similar a mi proyecto es la web [myfitnesspal](https://myfitnesspal.com). Es una web que cuenta con diferentes partes, desde un blog, comunidad y hasta la que trata sobre lo mismo que mi proyecto.

Es una web a mi parecer antigua, su estilo no es que incite mucho al uso ya que se ve anticuado y no es de lo más sencillo de usar, la búsqueda de los alimentos no es que sea la más sencilla y la creación de estos es tediosa, también cuenta con el apartado para hacer seguimiento de los ejercicios, aunque la búsqueda de ellos no es la mejor.

Es una web que en lo poco que la he usado en la búsqueda de este apartado no es que me haya incitado a volverla a usar, aquí es donde creo que podría hacerlo de una mejor forma en mi proyecto.

1.4. Viabilidad técnica y económica del proyecto

Recursos hardware y software necesarios:

Hardware: Se necesitará un equipo que actúe como servidor, en el cual se ejecutarán contenedores Docker. En estos contenedores, se alojarán el servidor de base de datos MySQL y otro contenedor con el servidor Symfony que proporcionará la API desarrollada. Este equipo también ejecutará el framework Angular, que se utilizará para crear la interfaz de usuario de la página web. Además, los usuarios deberán disponer de dispositivos con acceso a Internet.

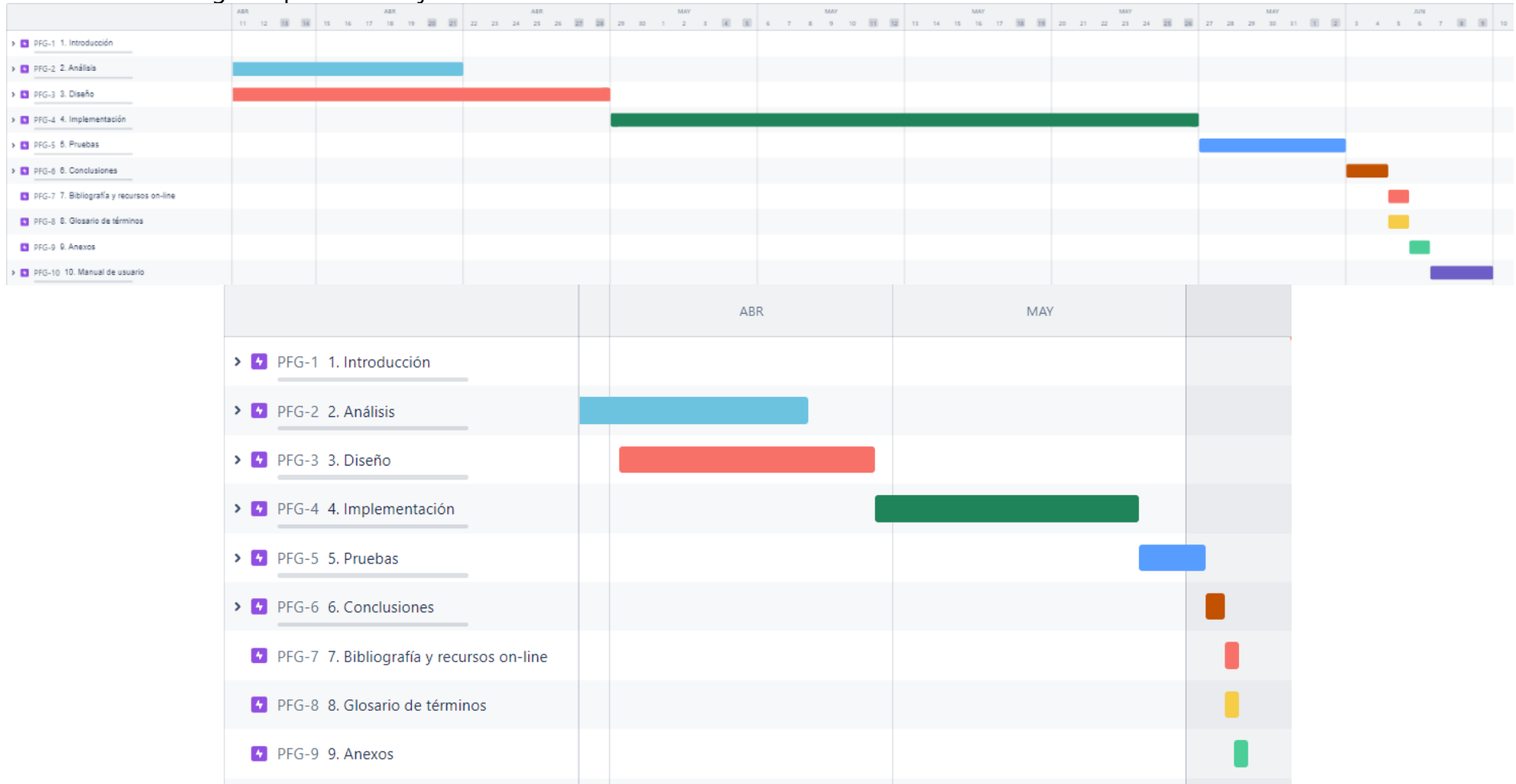
Software: Se utilizarán diversas tecnologías de desarrollo web, como HTML, CSS y JavaScript, para la interfaz de usuario. El framework de desarrollo web Angular se empleará para construir la parte front-end de la aplicación. Para el backend, se utilizará el framework Symfony para proporcionar la API que gestionará la lógica del servidor. La base de datos MySQL se utilizará para almacenar los datos del usuario, así como los datos relacionados con los alimentos, recetas y ejercicios necesarios para el funcionamiento de la aplicación. Estos datos incluirán información detallada sobre los alimentos disponibles, las recetas creadas por los usuarios y/o proporcionadas por los administradores, así como los diferentes tipos de ejercicios disponibles para realizar un seguimiento preciso de la actividad física. Todos estos componentes se ejecutarán en contenedores Docker, lo que garantiza una gestión eficiente y escalable del entorno de desarrollo y producción del proyecto.

La solución seleccionada ha sido pensada por su flexibilidad, escalabilidad y modularidad, haciendo de ella una opción viable para el desarrollo y la implementación de la aplicación web. La arquitectura basada en contenedores Docker y las tecnologías modernas de desarrollo web ofrecen una estructura que permite una gestión más eficiente y simplificada de cada componente del sistema.

1.5. Temporalización

La planificación elegida por mi para el desarrollo del proyecto es la siguiente.

Se añaden dos imágenes para una mejor visibilidad



2. Análisis

2.1. Documentación relevante

Las definiciones han sido obtenidas en su mayoría de [Wikipedia](#)

2.2. Definiciones

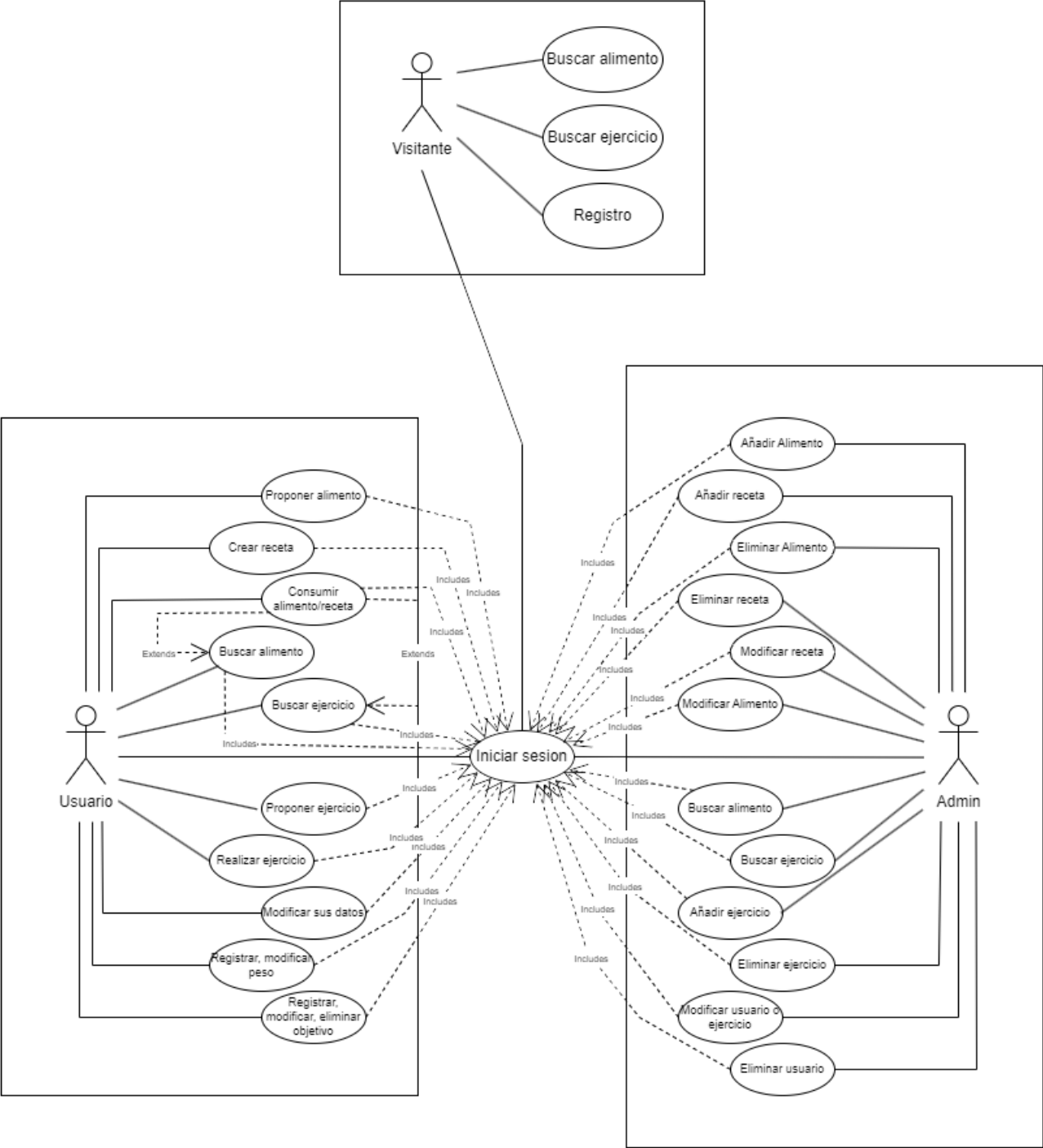
- Alimento: Cualquier sustancia consumida para proporcionar apoyo nutricional a un ser vivo.
- Caloría: Nombre de una unidad de energía que se usa para definir el valor nutricional de los alimentos.
- Índice de Masa Corporal (IMC): Medida utilizada para evaluar el peso corporal en relación con la altura, que puede ayudar a determinar si una persona tiene un peso saludable, infrapeso, sobrepeso u obesidad.
- Proteínas: Son nutrientes esenciales para el cuerpo humano.
- Grasas, Carbohidratos y Azúcares: Son nutrientes que se consumen para dar energía al organismo humano.
- Vitaminas y Minerales: Son compuestos esenciales para el funcionamiento adecuado del organismo humano
- Receta: Conjunto de instrucciones para realizar sobre diferentes alimentos para realizar un plato o bebida.

2.3. Requisitos funcionales y no funcionales

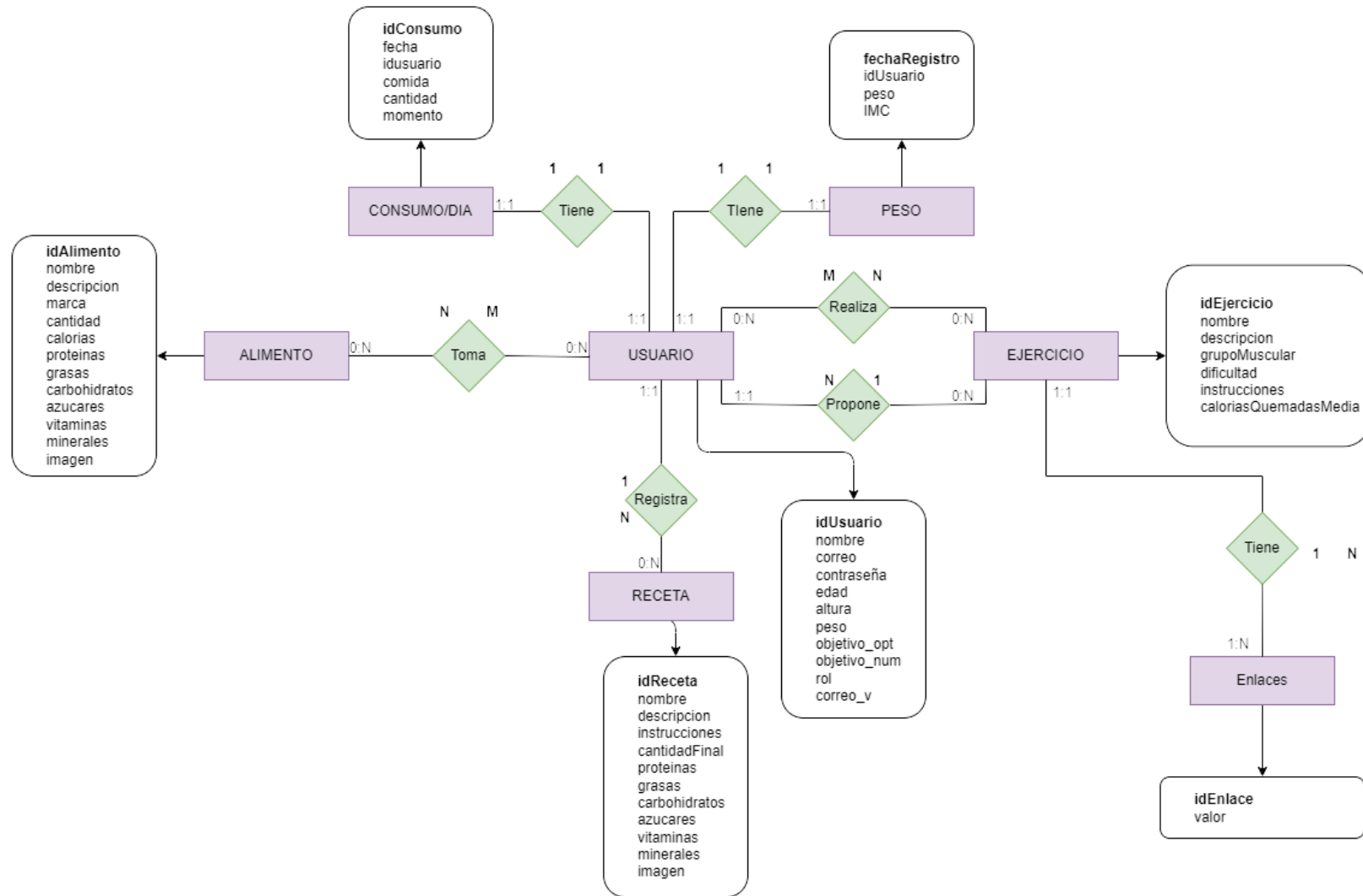
- Requisitos funcionales de datos
 - Sobre los usuarios se guardará: Nombre, edad, correo, contraseña, peso, altura, objetivo, peso objetivo.
 - De los alimentos se guardará: Nombre, descripción, marca y sus valores nutricionales
 - De las recetas se guardará: Nombre, descripción, ingredientes y sus valores nutricionales
 - De los ejercicios se guardará: Nombre, descripción, grupo muscular, instrucciones, dificultad, calorías quemadas
- Requisitos funcionales de contenido
 - Los ejercicios y recetas podrían tener enlaces a videos
 - Los alimentos y recetas podrían tener una imagen asociada
- Requisitos funcionales de transaccionales
 - Los usuarios puede iniciar sesión introduciendo correo y contraseña.
 - Los usuarios podrán añadir alimentos o recetas a su lista de consumiciones diaria
 - Los usuarios podrán añadir ejercicios a su lista de ejercicios realizados diaria
 - Los usuarios podrán proponer alimentos y ejercicios
 - Los usuarios podrán crear sus propias recetas
 - Los usuarios podrán introducir su peso a su registro de forma diaria

- Los usuarios podrán modificar, terminar y crear su objetivo
 - Los administradores podrán revisar, aceptar y eliminar propuestas de alimentos y ejercicios
 - Los administradores podrán crear y eliminar alimentos, recetas y ejercicios
 - Los administradores podrán inhabilitar y eliminar usuarios
- Requisitos funcionales de interfaz
 - La web brindará un mecanismo a los usuarios para modificar sus datos personales
- Requisitos no funcionales
 - Los usuarios solo podrán consumir las recetas creadas por el mismo o por los administradores
 - Los usuarios podrán proponer alimentos y ejercicios para un uso global
 - Los alimentos y ejercicios serán creados únicamente por los administradores

2.4. Casos de Uso. Diagramas UML de casos de uso



2.5. Modelado E/R. Diagramas E/R.



2.6. Sketching de la interfaz

Sketching

Estas son las imágenes escaneadas del sketching realizado sobre la web. Todas ellas se encuentran en la carpeta de imágenes.

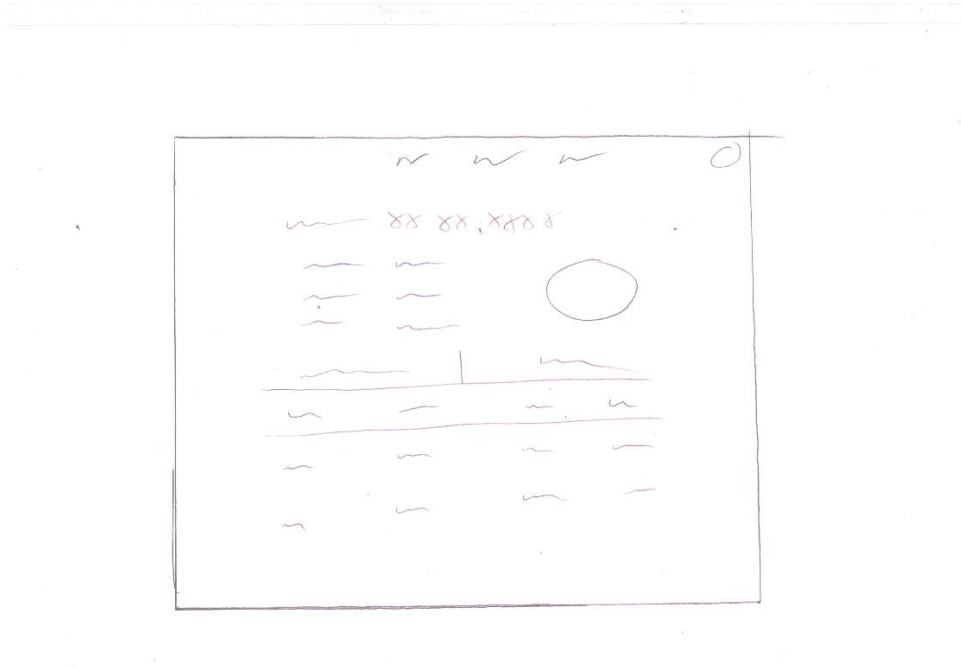


Imagen 1: Página de inicio

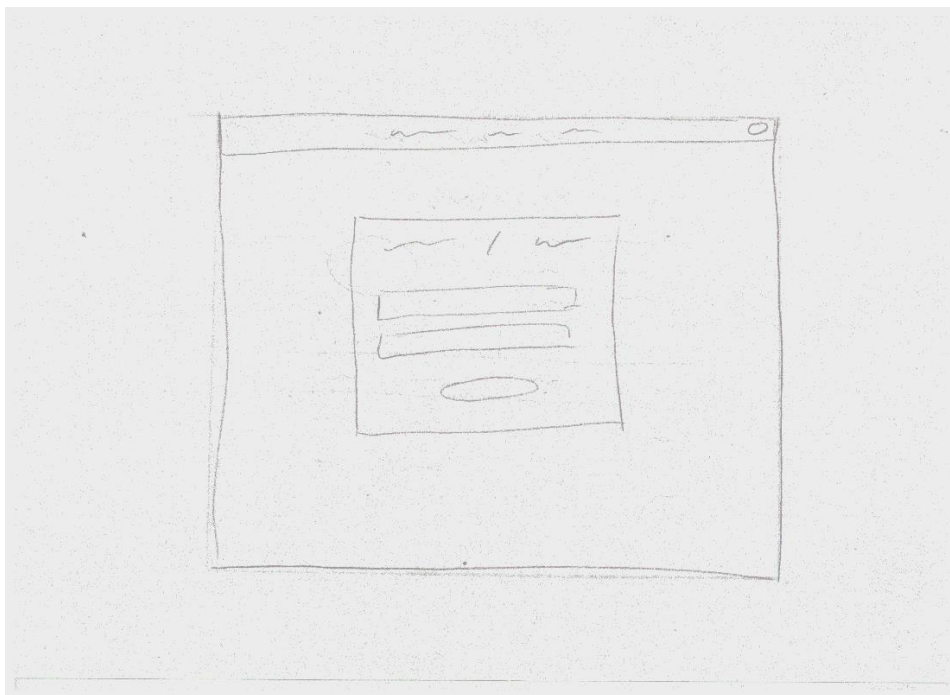


Imagen 2: Página de login

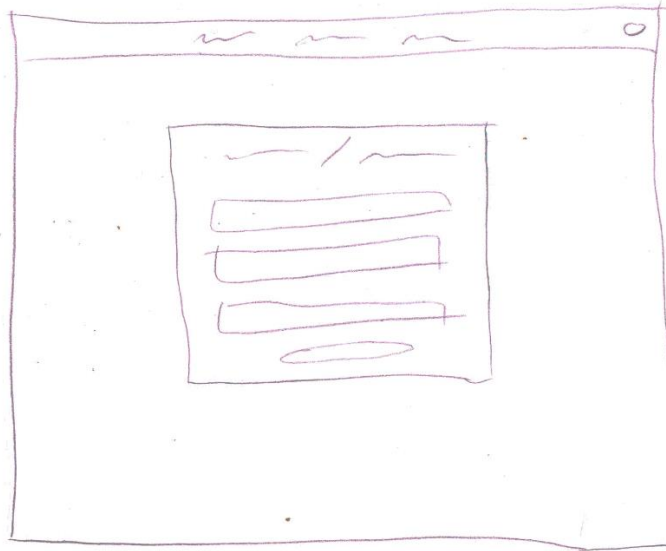


Imagen 3: Página de registro

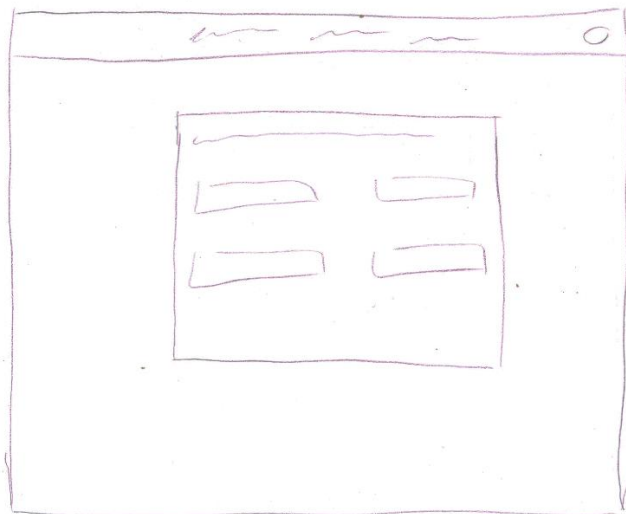


Imagen 4: Página de registro, datos personales

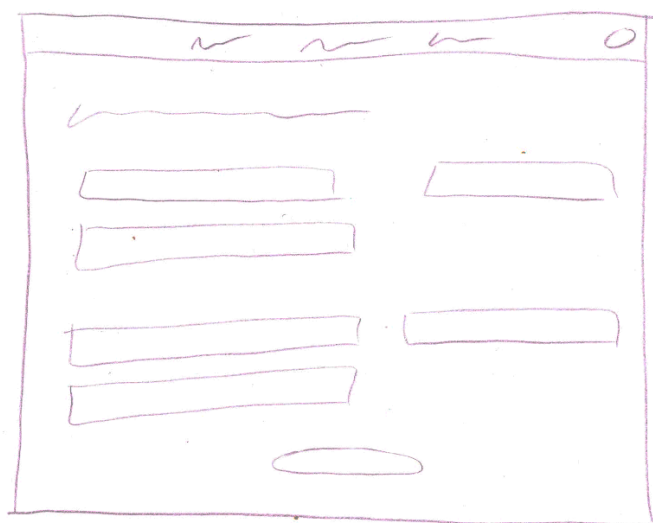


Imagen 5: Página de perfil

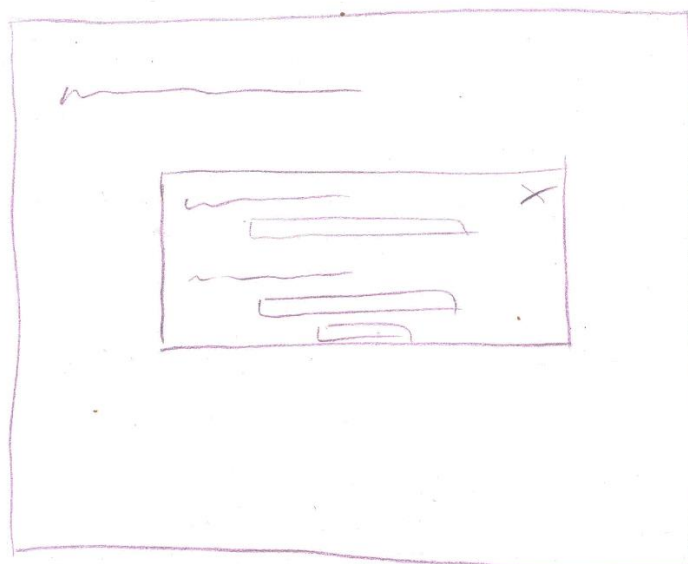


Imagen 6: Página de perfil, modificar dato

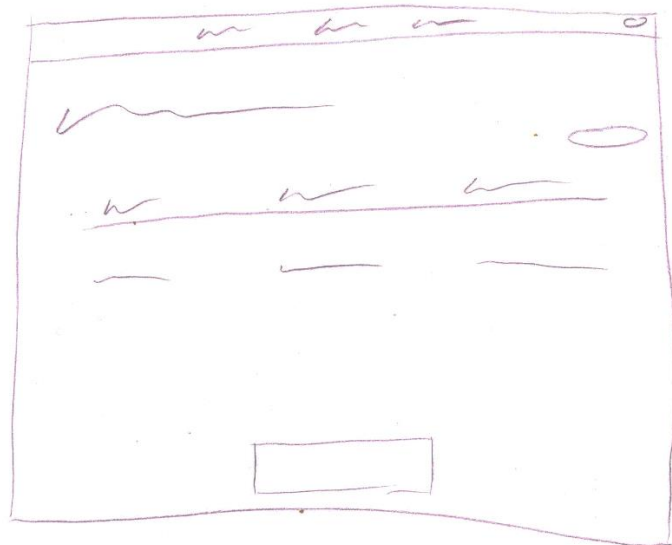


Imagen 7: Página de comidas home

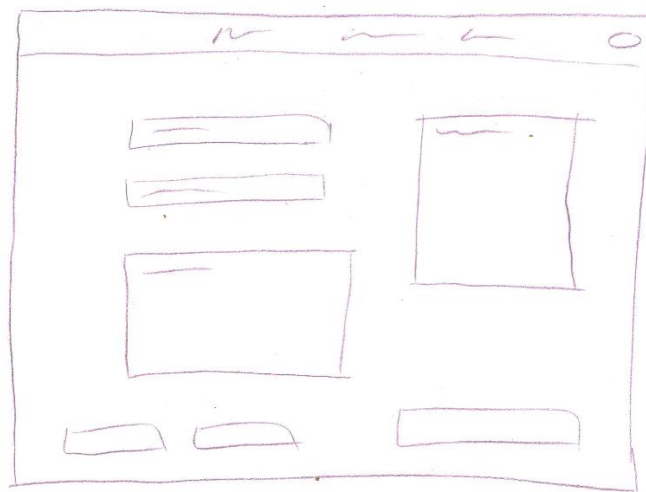


Imagen 8: Popup consumir alimento

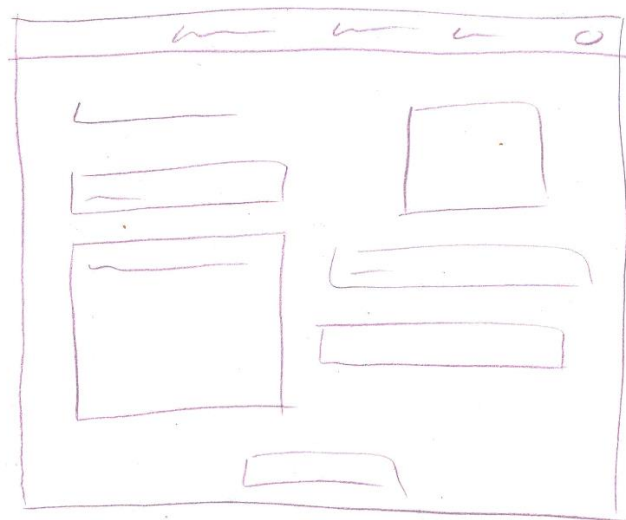


Imagen 9: Formulario proponer alimento

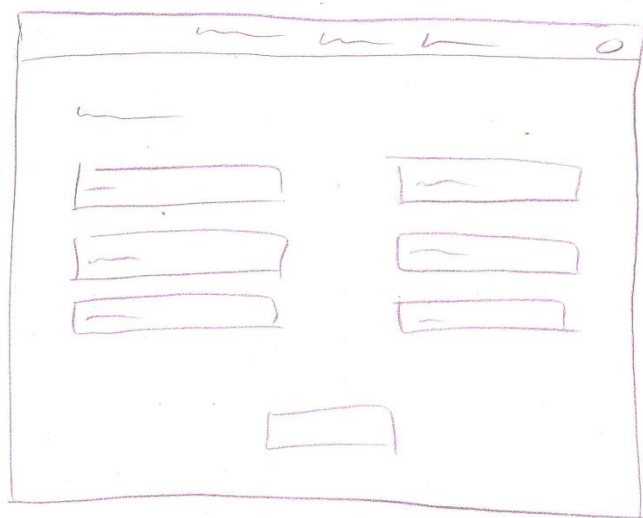


Imagen 10: Formulario proponer alimento, valores nutricionales

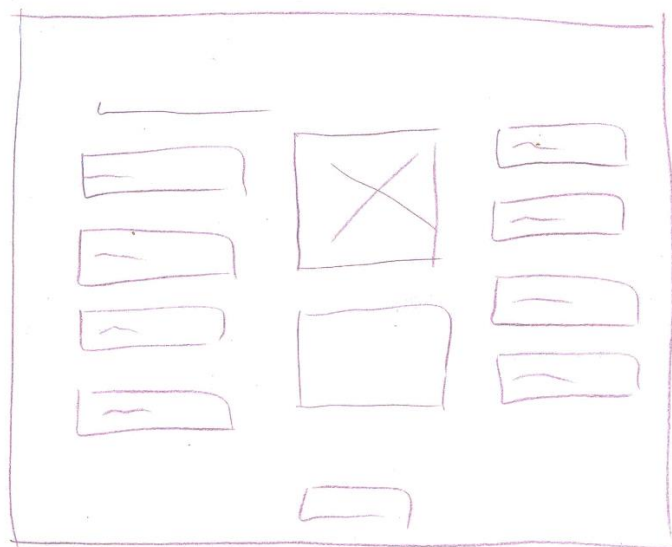


Imagen 11: Página información de alimento

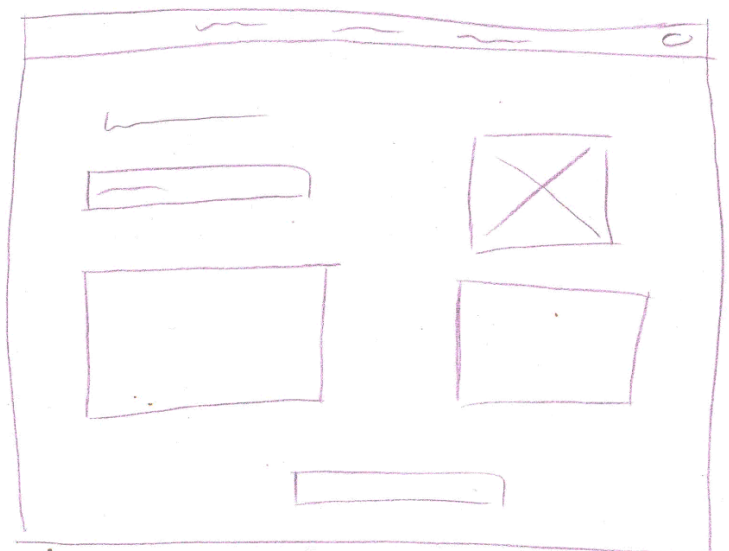


Imagen 12: Formulario agregar receta

A hand-drawn sketch of a form for adding a recipe and nutritional values. The form is enclosed in a rectangular border. At the top, there is a title bar with three wavy lines and a small circle on the right. The main area contains two columns of three rectangular input fields each, and a single rectangular input field at the bottom center.

Imagen 13: Formulario agregar receta, valores nutricionales

A hand-drawn sketch of a recipe information page. The form is enclosed in a rectangular border. At the top, there is a title bar with three wavy lines and a small circle on the right. The main area contains two columns of four rectangular input fields each, and a single rectangular input field at the bottom center. A large square with an 'X' is positioned in the center of the two columns.

Imagen 14: Página información de receta



Imagen 15: Página de peso home

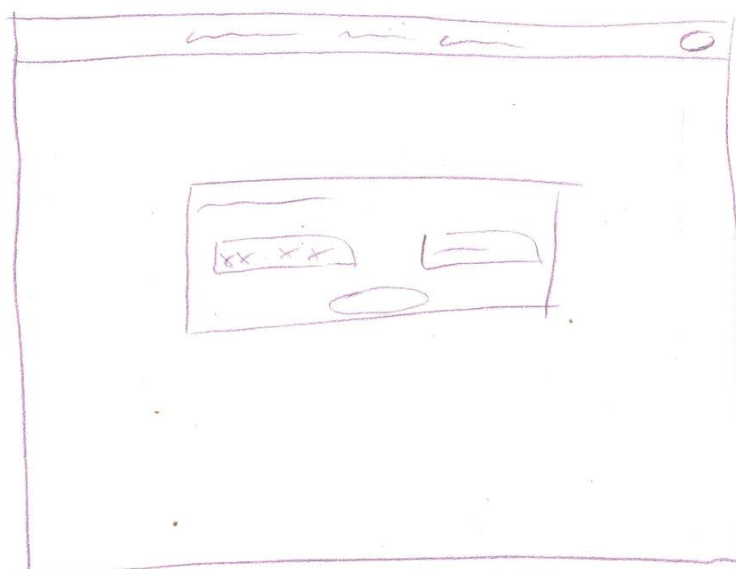


Imagen 16: Popup registrar peso

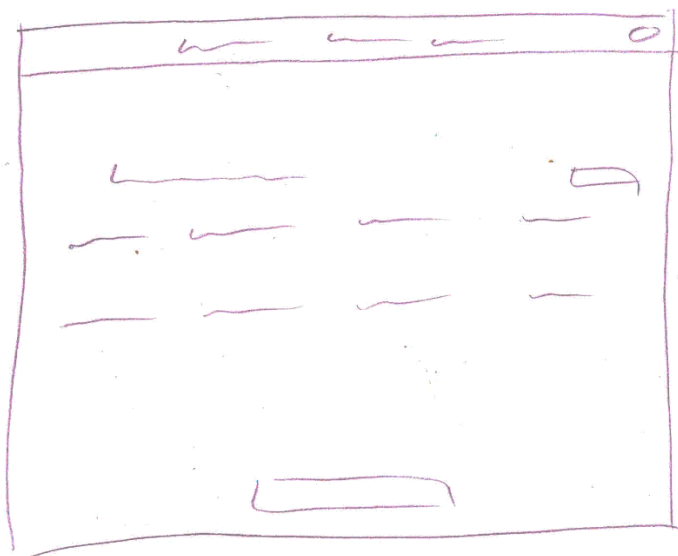


Imagen 17: Página ejercicios home

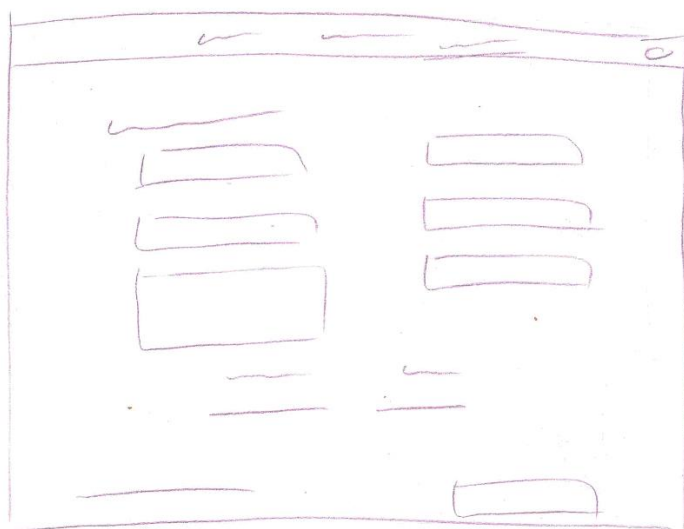


Imagen 18: Formulario agregar ejercicio realizado

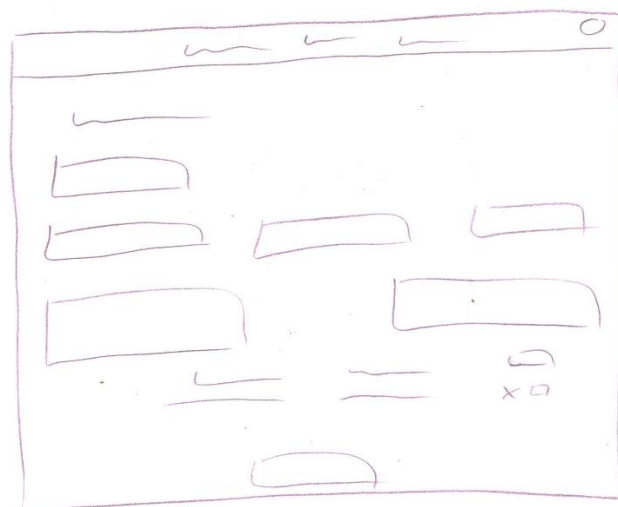


Imagen 19: Formulario proponer ejercicio

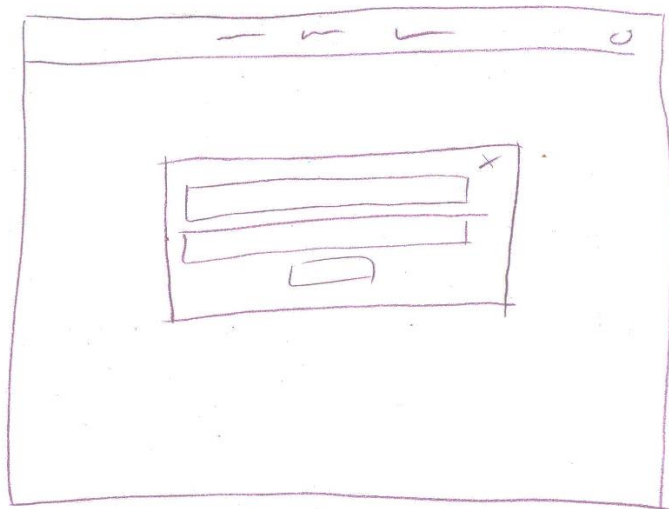


Imagen 20: Popup agregar enlace a ejercicio

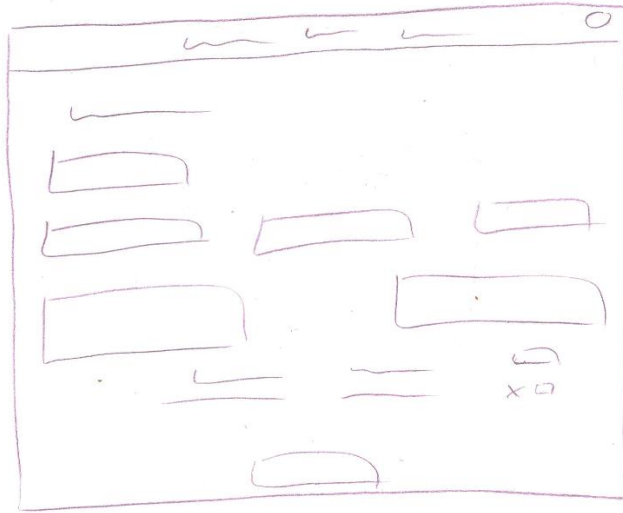


Imagen 21: Página ver información de ejercicio

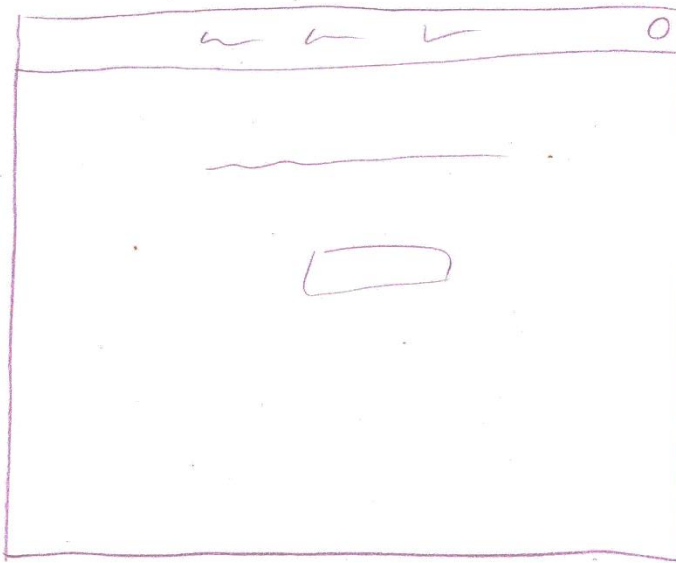


Imagen 22: Página mensaje de error

Wireframing

Estas son las imágenes del Wireframing obtenidas desde Figma.



Imagen 23: Página de inicio, tabs Ejercicios

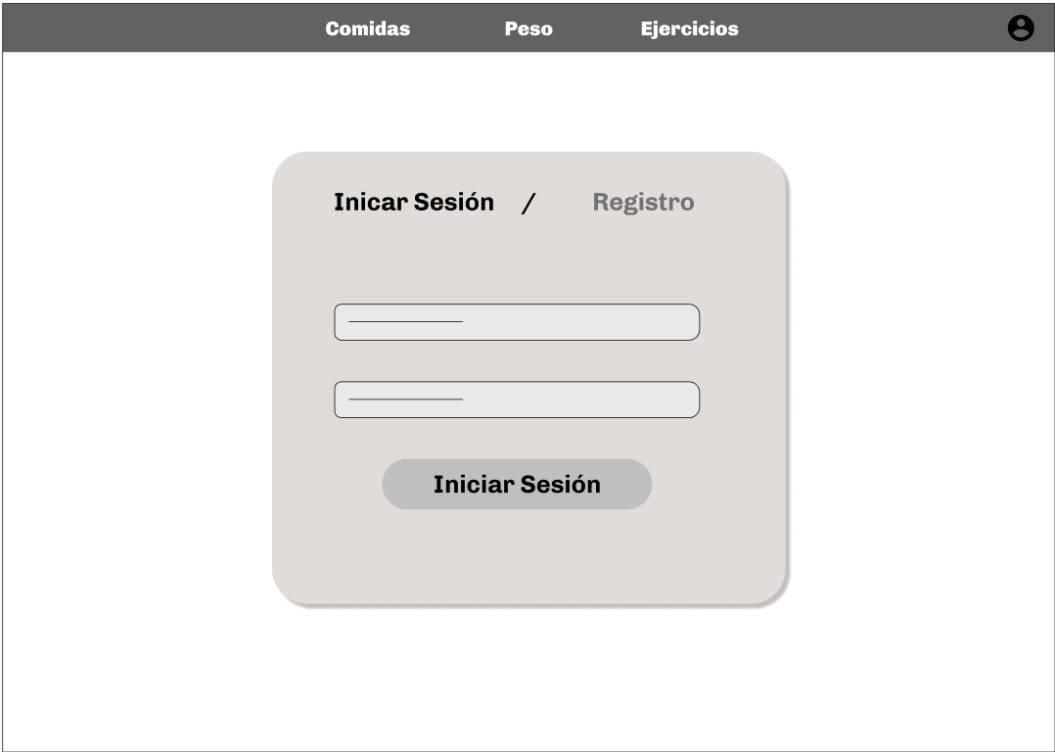
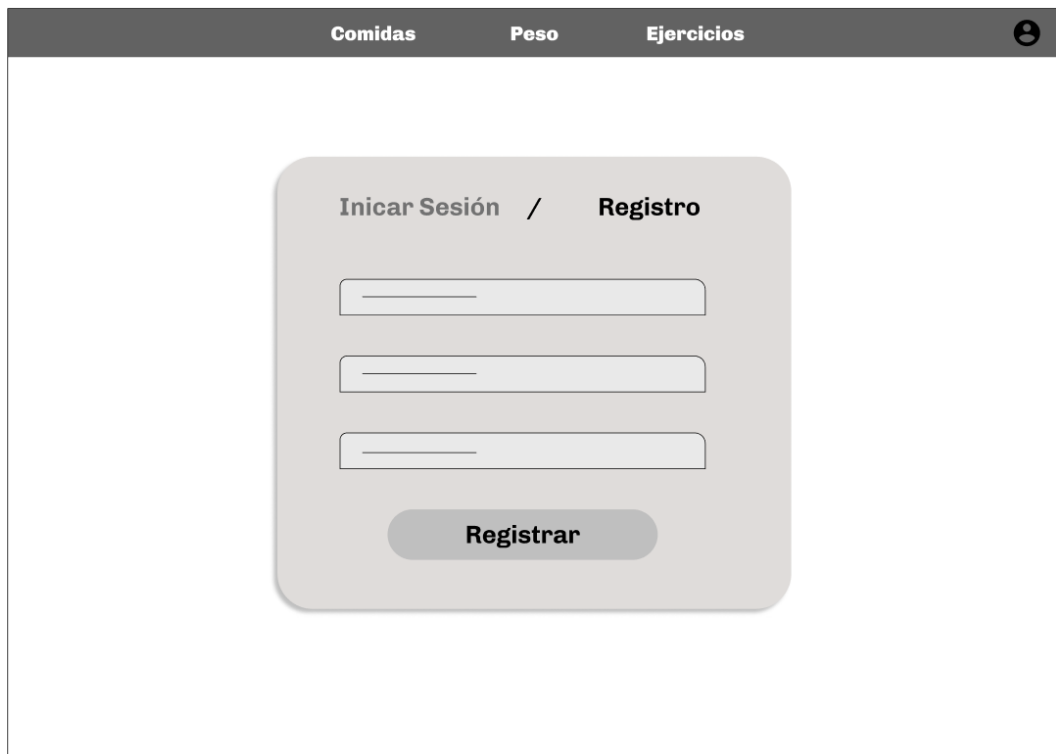



Imagen 24:Fomulario inicio de sesión



Comidas Peso Ejercicios 

Inicar Sesión / Registro

Registrar

Imagen 25: Formulario de registro:



Comidas Peso Ejercicios 

Rellena los siguientes datos

Registrar

Imagen 26: Formulario de registro, datos personales

Comidas

Peso

Ejercicios

Estos son tus datos personales

Cerrar Sesión

Imagen 27: Página de perfil

Comidas

Peso

Ejercicios

Formulario cambio dato personal

Valor actual

Introduce el nuevo valor

Cancelar

Guardar

Imagen 28: Página de perfil, modificar datos

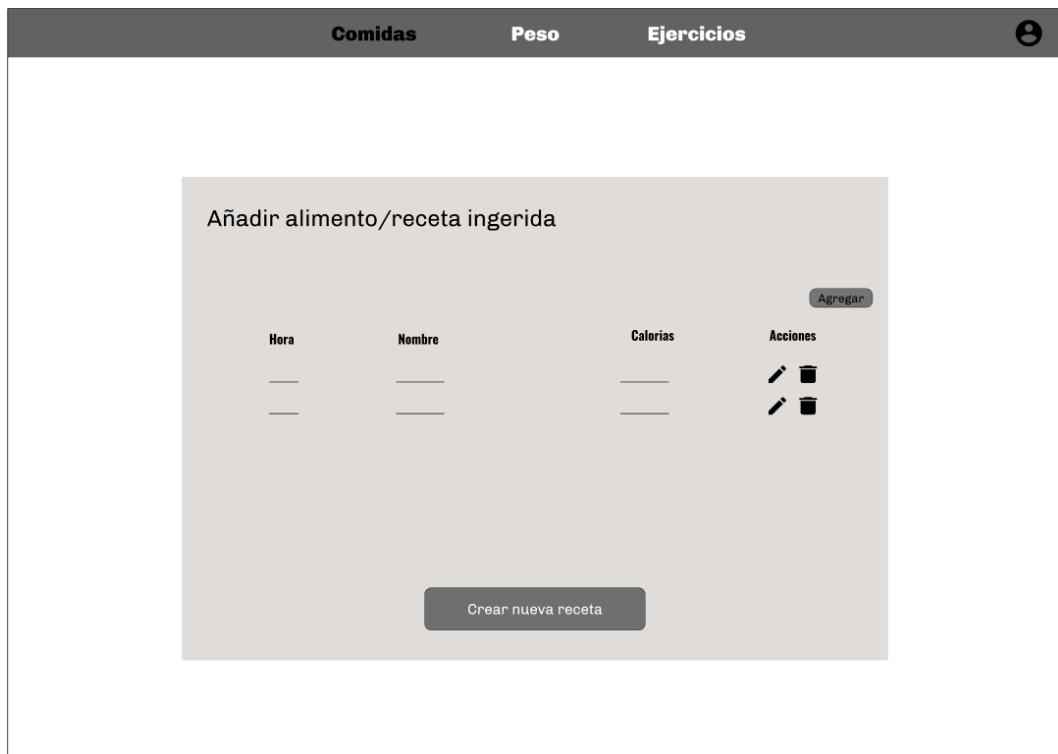


Imagen 29: Página de comidas home

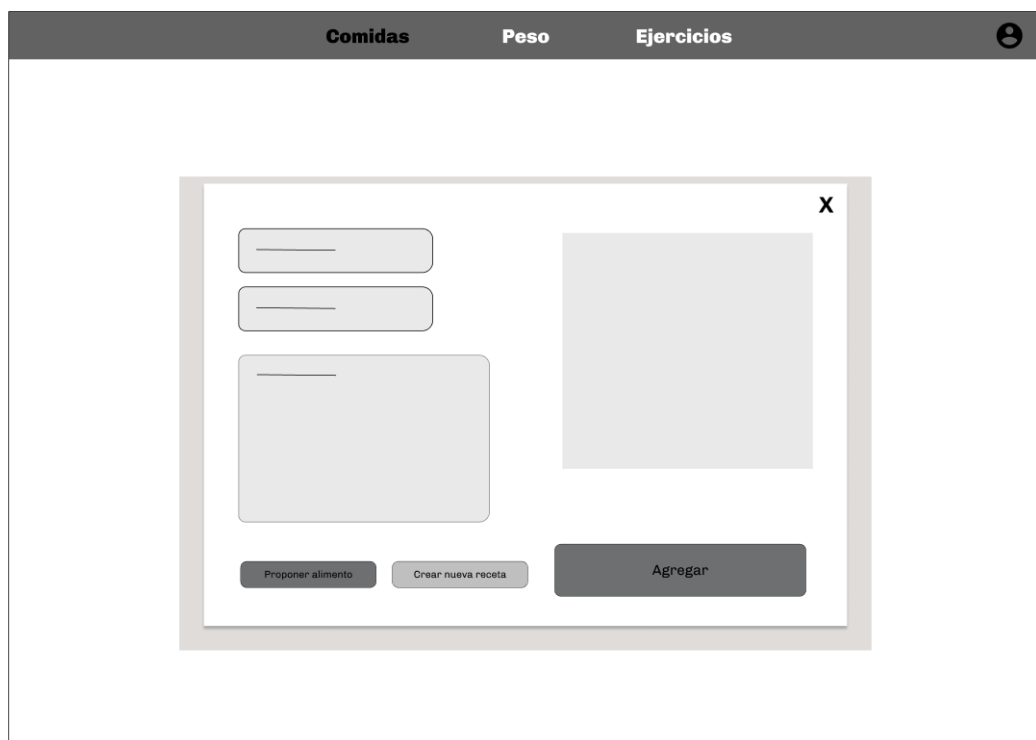


Imagen 30: Popup consumir alimento

Comidas

Peso

Ejercicios

Proponer Alimento

Siguiente

Comidas

Peso

Ejercicios

Proponer Alimento

Proponer

Comidas
Peso
Ejercicios

Informacion de Alimento

Salir

Imagen 33: Página información de alimento

Comidas
Peso
Ejercicios

Agregar Receta

Siguiete

Imagen 34: Formulario agregar receta

Comidas

Peso

Ejercicios

Proponer Alimento

Proponer

Imagen 35: Formulario agregar receta, valores nutricionales

Comidas

Peso

Ejercicios

Informacion de Receta

Salir

Imagen 36: Página información de receta



Imagen 37: Pagina de peso home

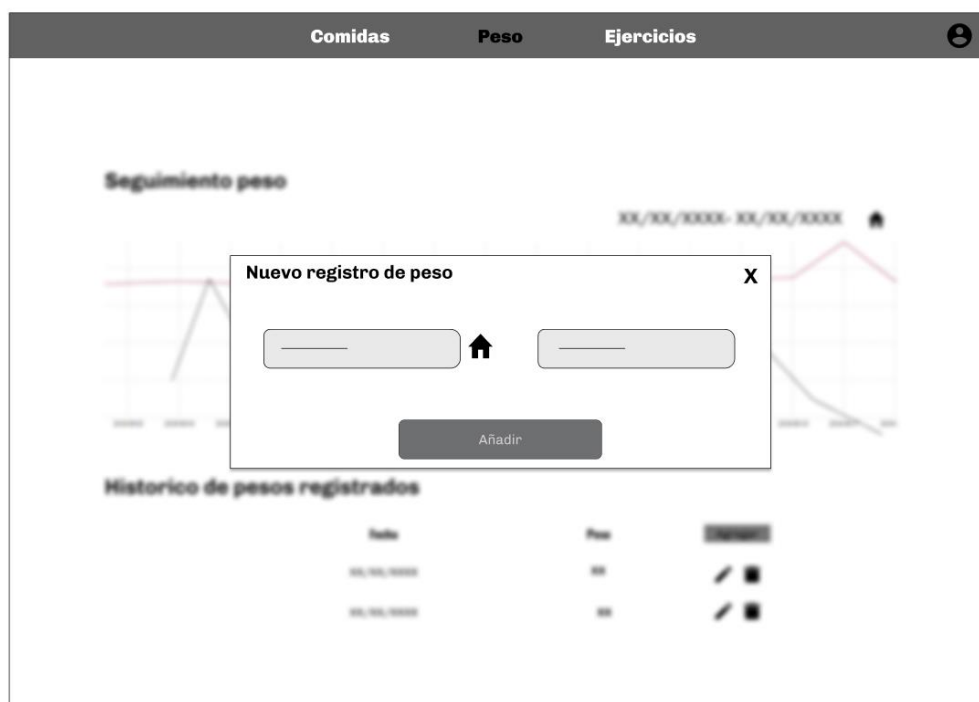


Imagen 38: Popup peso agregar peso



Imagen 39: Página peso home

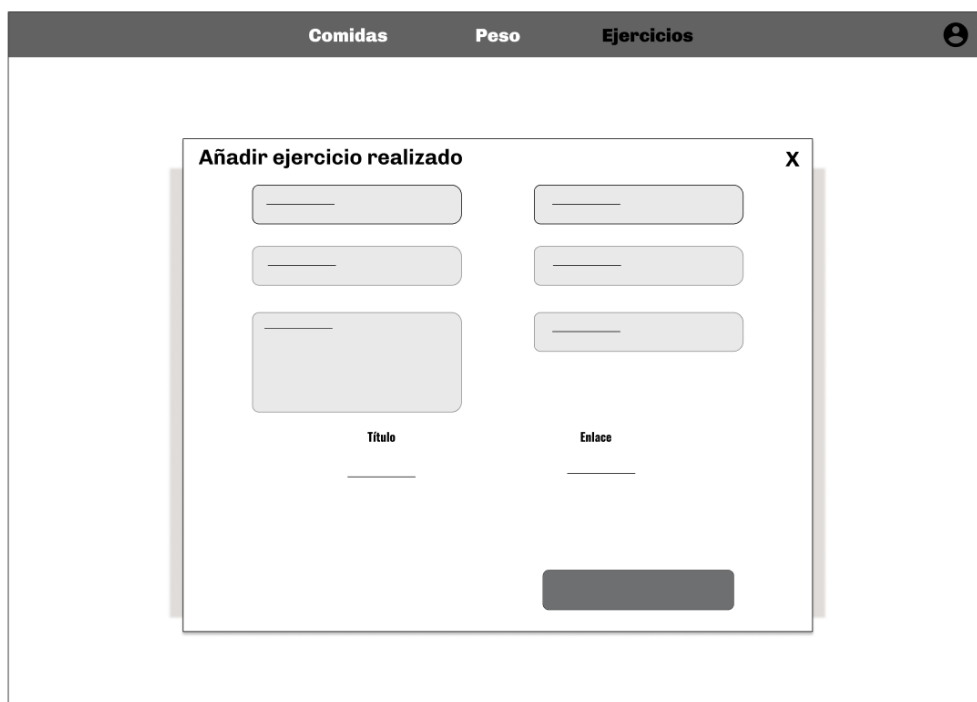


Imagen 40: Popup registrar peso

The screenshot shows a web application with a dark header containing the navigation links 'Comidas', 'Peso', and 'Ejercicios', along with a user profile icon. The main content area is titled 'Proponer Ejercicio' and contains a form with the following elements:

- A single-line text input field.
- Three single-line text input fields arranged horizontally.
- Two larger multi-line text input fields.
- Labels 'Titulo' and 'Enlace' positioned above two single-line text input fields.
- An 'Agregar' button with a plus icon.
- Two sets of edit (pencil) and delete (trash) icons.
- An 'Enviar' button at the bottom.

Imagen 41: Pagina ejercicio home

This screenshot shows the same 'Proponer Ejercicio' form as in the previous image, but with a modal popup open in the center. The modal has a close button (X) in the top right corner and contains:

- Two single-line text input fields.
- An 'Agregar' button.

The background form is visible but slightly faded behind the modal.

Pagina 42: Popup añadir enlace ejercicio

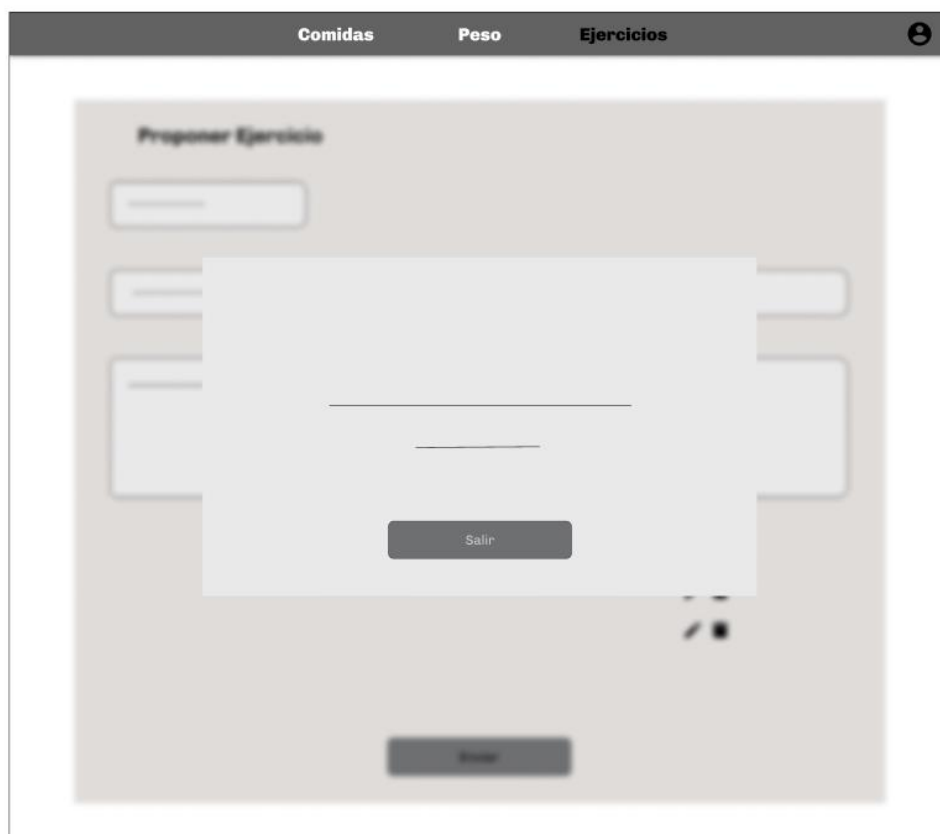


Imagen 43: Mensaje agradecimiento

The screenshot shows a web application interface with a dark header bar containing the navigation links 'Comidas', 'Peso', and 'Ejercicios', along with a user profile icon. The main content area is titled 'Información Ejercicio'. It features a form with several input fields. The form is organized into two columns. The left column has three input fields, and the right column has two input fields. Below the input fields, there are labels 'Titulo' and 'Enlace' with corresponding input fields. At the bottom of the form, there is a dark button.

Imagen 44: Página información ejercicio

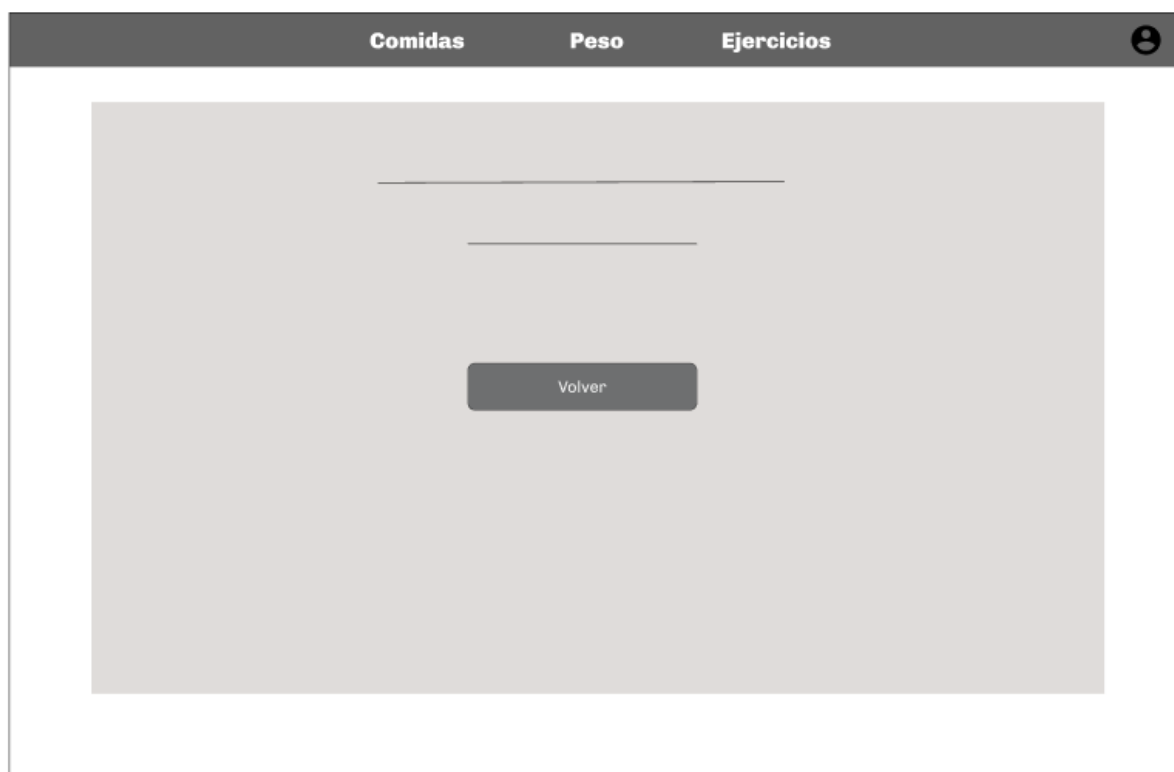


Imagen 45: Página mensaje de error

3. Diseño

3.1 Arquitectura hardware y software de la solución

Hardware:

Equipo local o servidor

Software:

Docker: para manejar el contenedor con la base de datos

Symfony con Doctrine: para ejecutar el servidor de la Api

Angular: para lanzar la web

3.2 Modelado funcional de la solución. Diagramas de clase

En la siguiente imagen muestro como ha quedado el diagrama de clase final.

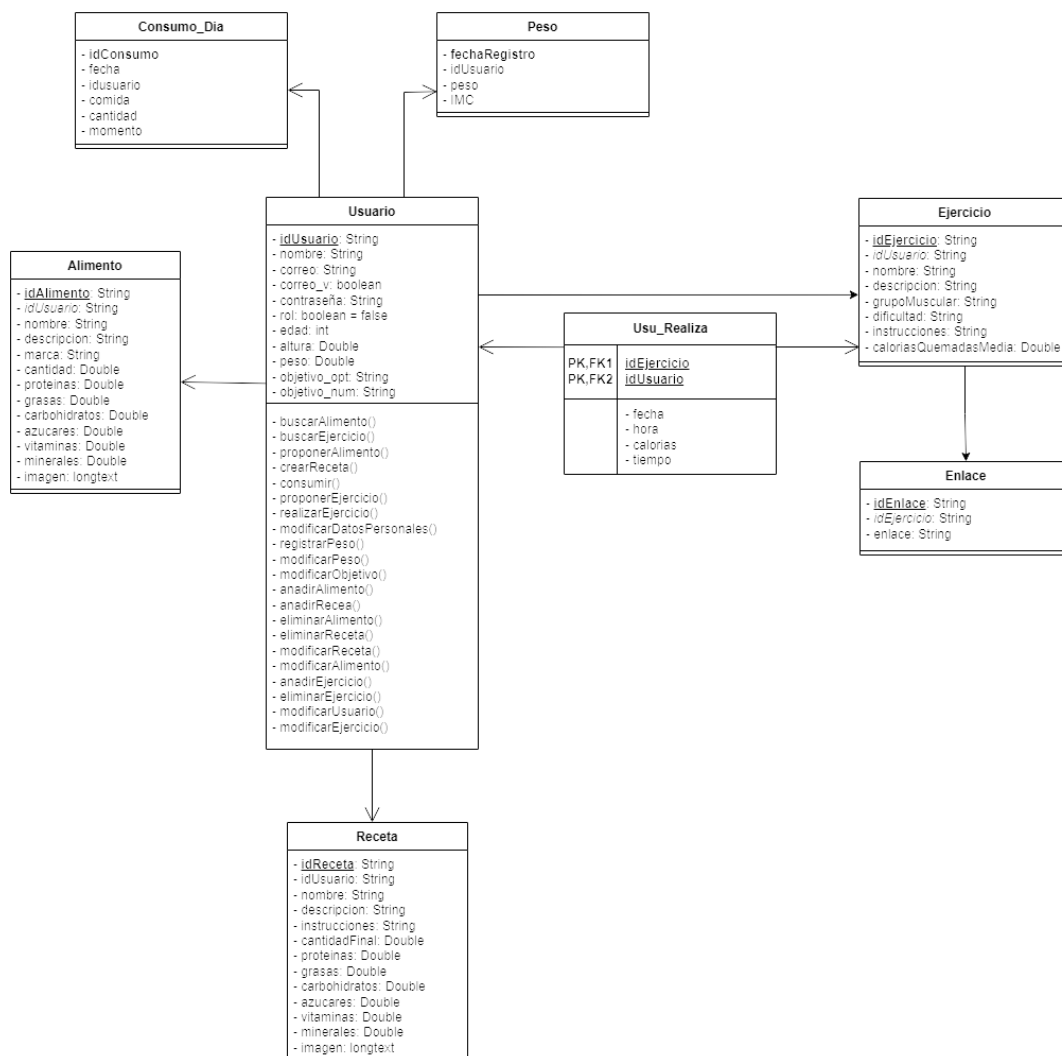


Imagen 46: Diagrama de clase

3.3 Modelado de datos. Modelo relacional. Diccionario de Datos

En la siguiente imagen muestro como ha quedado el modelo relacional de mi base de datos.

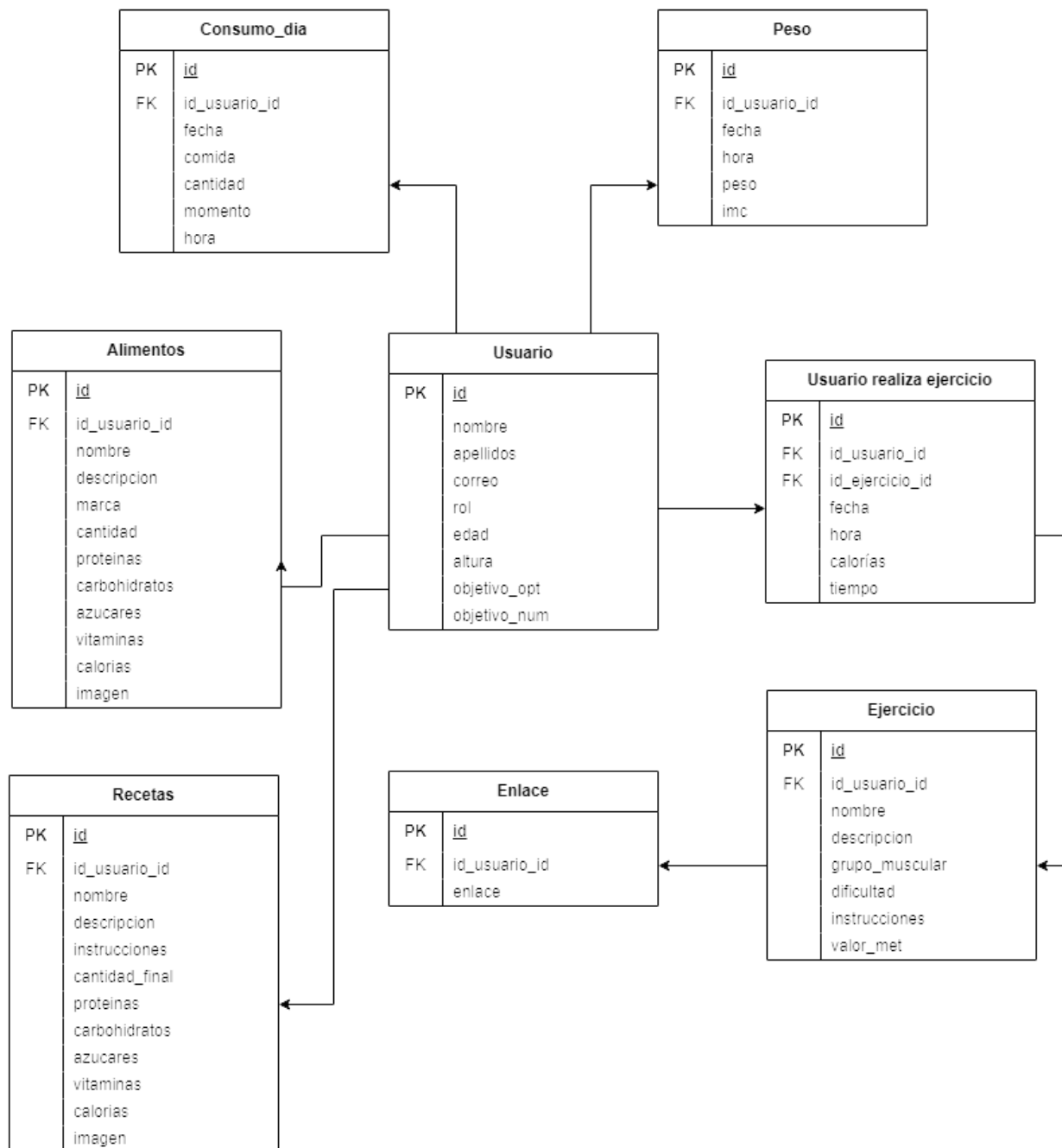


Imagen 47: Modelo relacional

3.4 Prototipado de la interfaz

Estas son las imágenes del prototipado de la web realizado en Figma.



Imagen 48: Página de inicio

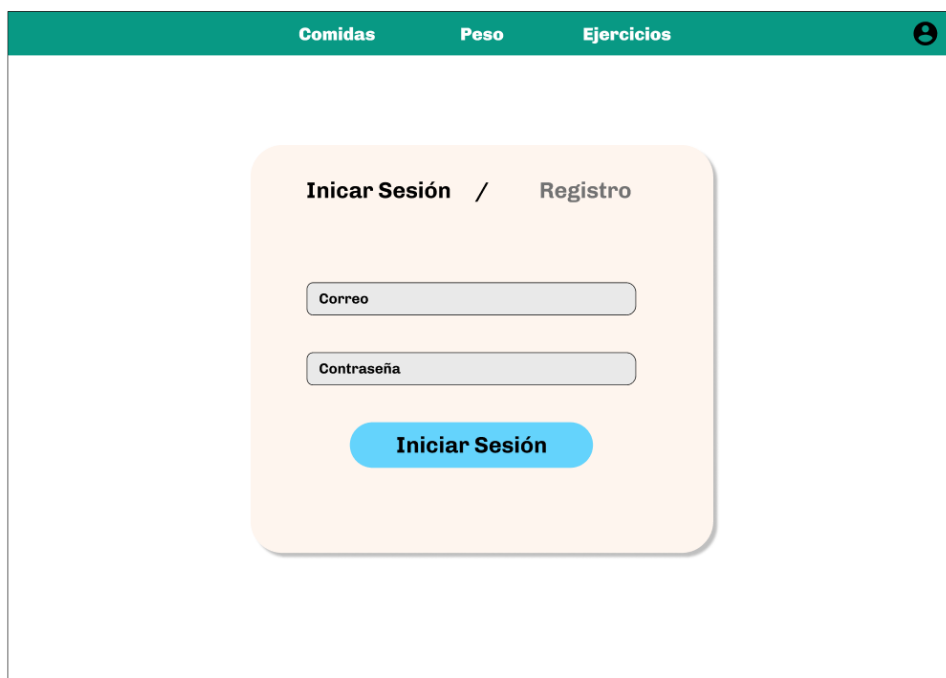



Imagen 49: Pagina de login

Comidas Peso Ejercicios 

Iniciar Sesión / Registro


Correo

Contraseña

Repetir contraseña

Registrar

Imagen 50: Pagina de registro

Comidas Peso Ejercicios 

Rellena los siguientes datos

Nombre Apellidos

Altura Peso

Registrar

Imagen 51: Pagina de registro, datos personales

The screenshot shows a web application interface with a teal header bar containing the navigation links "Comidas", "Peso", and "Ejercicios", along with a user profile icon on the right. The main content area is titled "Estos son tus datos personales". It contains several input fields for personal information: "Nombre", "Apellidos", "Correo", "Altura", "Peso", and "Objetivo". Each of these fields has a small pencil icon to its right, indicating they are editable. At the bottom of the form, there is a blue button labeled "Cerrar Sesión".

Imagen 52: Pagina de perfil

The screenshot displays a web application interface with a teal header bar containing the navigation links "Comidas", "Peso", and "Ejercicios", and a user profile icon on the right. The main content area is titled "Formulario cambio dato personal". It features a form with two input fields: "Valor actual" (containing the text "PropiedadOld") and "Introduce el nuevo valor" (containing the text "PropiedadNuevo"). Below these fields are two buttons: a teal "Cancelar" button and a blue "Guardar" button.

Imagen 53: Pagina de perfil, modificar dato

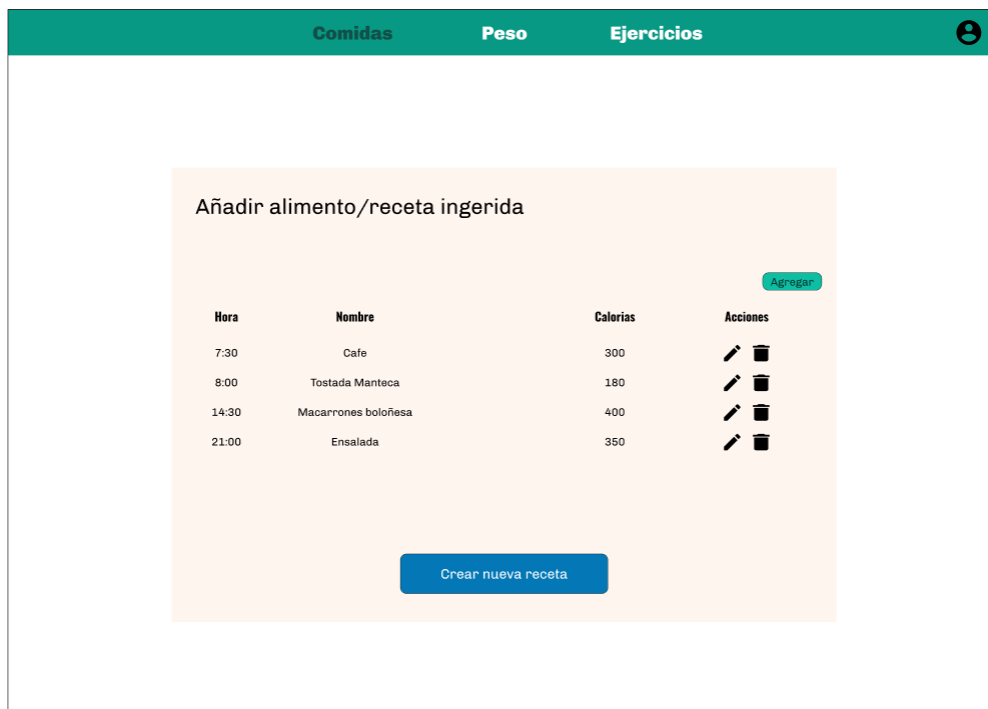


Imagen 54: Pagina de comidas home

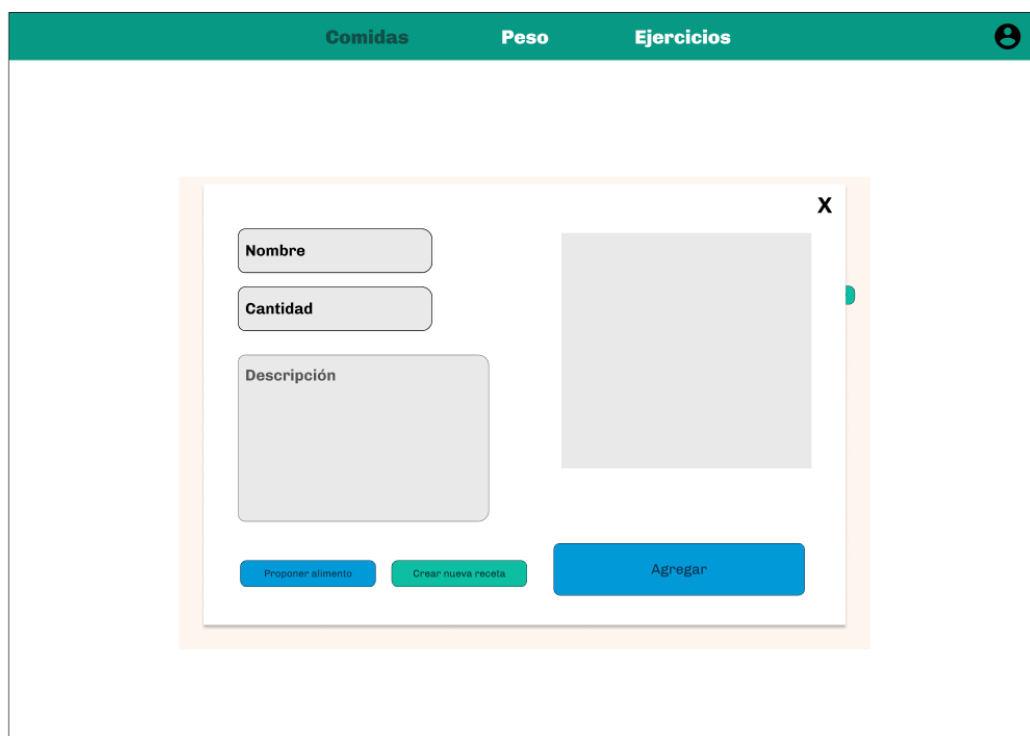


Imagen 55: Popup agregar alimento consumido

Comidas

Peso

Ejercicios

Proponer Alimento

Nombre

Descripción

Cantidad

Marca

Siguiente

Imagen 56: Formulario proponer alimento

Comidas

Peso

Ejercicios

Proponer Alimento

Calorías

Grasas

Proteínas

Vitaminas

Azúcares

Carbohidratos

Proponer

Imagen 57: Formulario proponer alimento, datos personales

Comidas

Peso

Ejercicios

Informacion de Alimento

Nombre

Marca

Cantidad

Calorias

Grasas

Descripción

Vitaminas

Proteinas

Azucares

Carbohidratos

Salir

Imagen 58: Pagina información de alimento

Comidas

Peso

Ejercicios

Agregar Receta

Nombre

Instrucciones

Descripción

Siguiente

Imagen 59: Formulario agregar receta

Comidas Peso Ejercicios

Proponer Alimento

Calorías

Grasas

Proteínas

Vitaminas

Azúcares

Minerales

Carbohidratos

Proponer

Imagen 60: Formulario agregar receta, datos nutricionales

Comidas Peso Ejercicios

Información de Receta

Nombre

Cantidad final

Instrucciones

Descripción

Calorías

Minerales

Grasas

Vitaminas

Proteínas

Azúcares

Carbohidratos

Salir

Imagen 61: Página información de receta

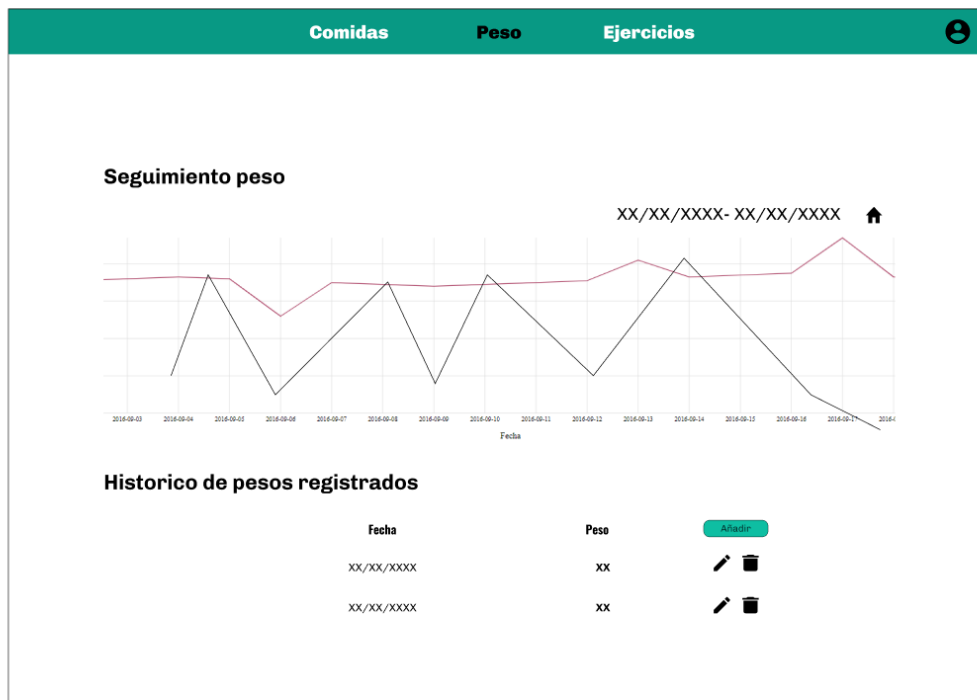


Imagen 62: Pagina de peso home

The screenshot shows the 'Peso' page with a 'Nuevo registro de peso' (New weight record) popup form overlaid. The form has a title bar with 'Nuevo registro de peso' and a close button 'X'. It contains two input fields: 'Fecha' (Date) and 'Peso (kg)' (Weight in kg), separated by a house icon. Below the inputs is a blue 'Añadir' (Add) button. The background shows the same line graph and table as in the previous image, but they are blurred.

Imagen 63: Popup registrar peso

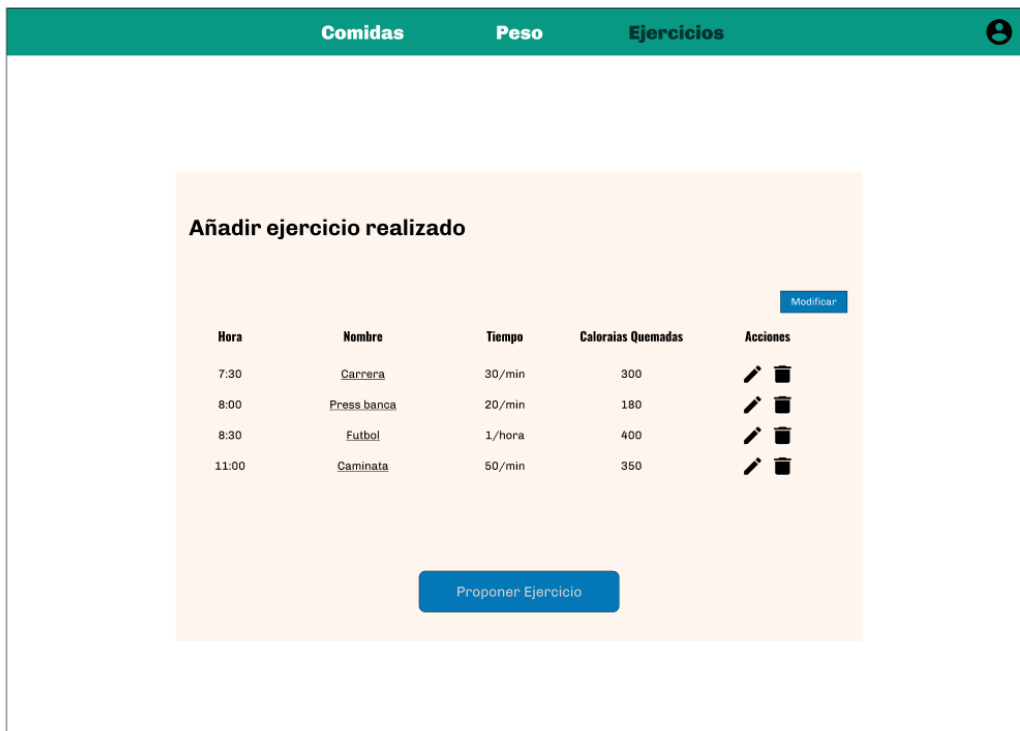


Imagen 63: Pagina de ejercicio home

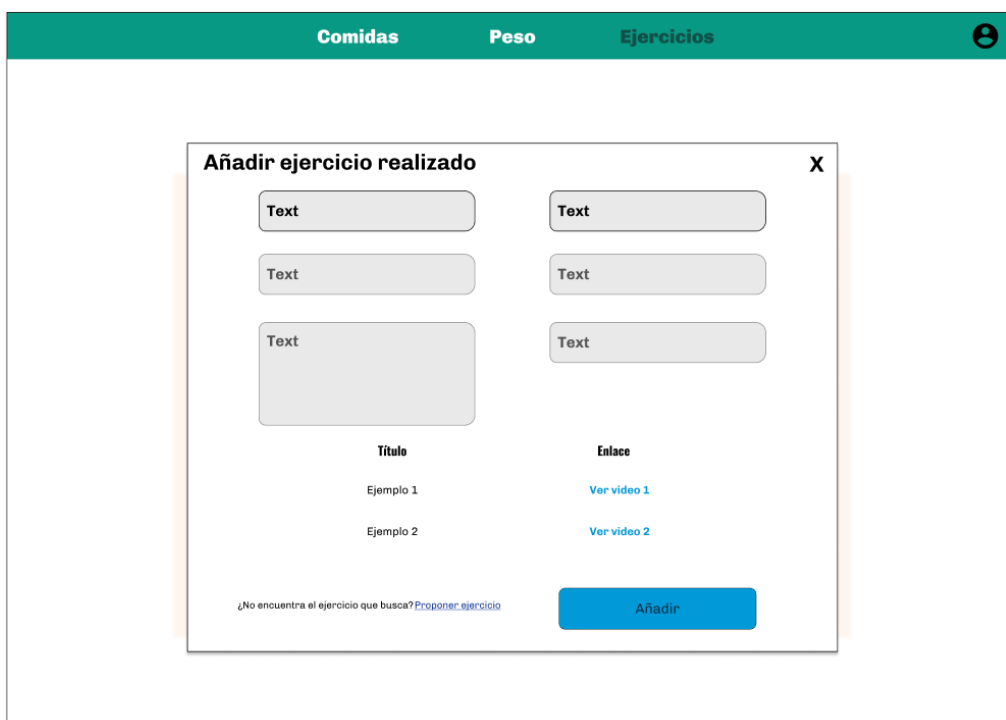


Imagen 64: Popup añadir ejercicio realizado

Comidas

Peso

Ejercicios

Proponer Ejercicio

Nombre

Difficultad

Grupo Muscular

Valor MET

Descripción

Instrucciones

Título

Ejemplo 1

Ejemplo 2

Enlace

Ver video 1

Ver video 2

Agregar

Enviar

Imagen 65: Formulario proponer ejercicio

Comidas

Peso

Ejercicios

Proponer Ejercicio

Nombre

Difficultad

Grupo Muscular

Valor MET

Descripción

Instrucciones

Título

Ejemplo 1

Ejemplo 2

Enlace

Ver video 1

Ver video 2

Agregar

Enviar

X

Título

URL

Agregar

Imagen 66: Popup añadir enlace

45



Imagen 67: Mensaje de agradecimiento

The screenshot shows the 'Información Ejercicio' form within the same web application. The header and navigation tabs are identical. The form contains several input fields: 'Nombre', 'Dificultad', 'Grupo Muscular', 'Valor MET', 'Descripción', and 'Instrucciones'. Below the 'Descripción' and 'Instrucciones' fields, there are two columns of example data. The 'Titulo' column lists 'Ejemplo 1' and 'Ejemplo 2'. The 'Enlace' column lists 'Ver video 1' and 'Ver video 2', both in blue text. A blue button labeled 'Enviar' is located at the bottom center of the form.

Imagen 68: Pagina información de ejercicio

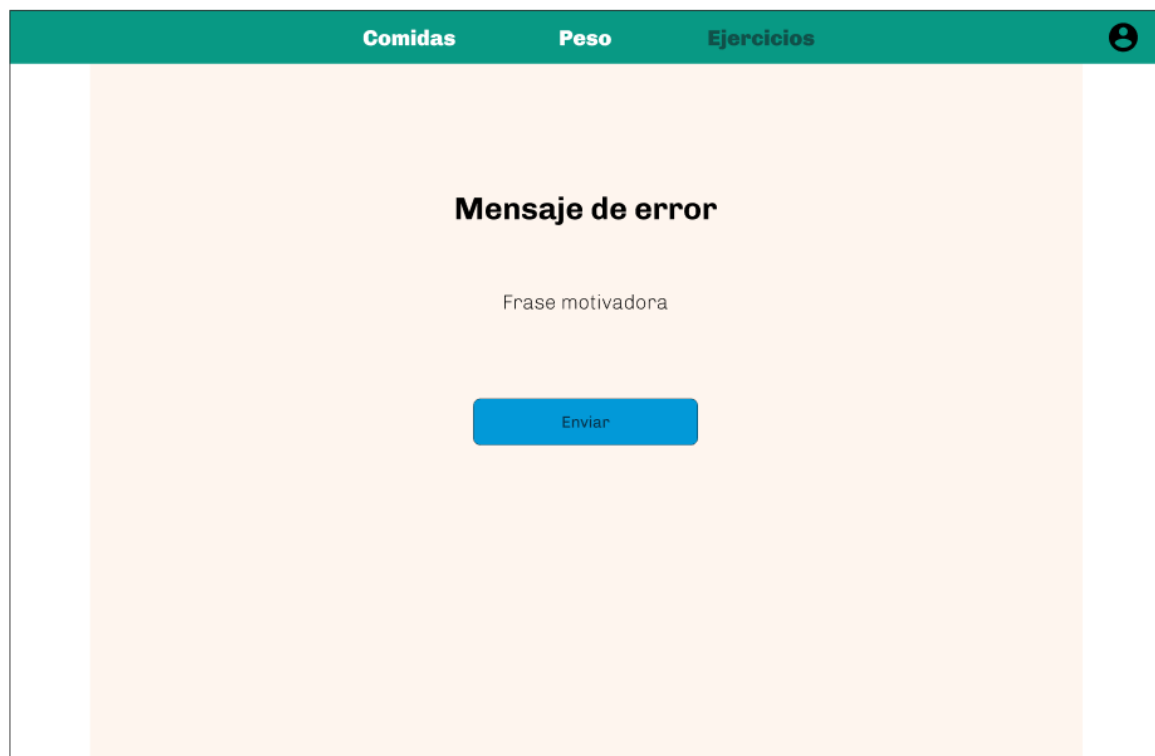


Imagen 69: 'Página mensaje de error

4. Implementación

4.1 Requisitos de instalación y ejecución

Para la ejecución del proyecto al completo, tanto para Linux o Windows, ya que ha sido probado en ambos, se necesitan los siguientes softwares:

Para la base de datos es necesario tener algún gestor de base de datos, en mi caso ha sido el proporcionado por [XAMPP](#) con la versión más reciente v8.2.12, este siendo ejecutado en un contenedor Docker debido a requisitos del proyecto, usando la versión v4.29, en un contenedor Ubuntu 20.04 el cual he subido a Docker Hub y se puede encontrar con el nombre [hehedaniel/backcompleto:backfuncional](#).

Para la Api es necesario contar con Symfony, en mi caso he usado la versión v5.8.19 junto con Doctrine, por ende, también se necesita PHP, en mi caso la versión v8.2.10. Esta Api también se puede ejecutar en el propio Docker en el que ejecutamos la base de datos.

Por último, para el front-end creado con Angular necesitamos, la propia instalación de Angular en la que usamos Angular v17.3.10, Angular Cli v17.3.8, Node v20.14.0 y Npm v10.7.0 .

A la hora de ejecución es indispensable realizar los comandos necesarios para la instalación de dependencias en ambas partes del proyecto, para la Api "composer update" y para el Front-end "Npm install".

Tras esto habrá que lanzar ambas partes, siendo "Symfony server:start" para lanzar la Api, "ng serve -o" (-o hace que se habrá la ventana directamente en el navegador) para angular y por último para lanzar la base de datos en mi caso ha sido "/opt/lampp/lampp start" para ejecutar el gestor de base de datos en el contenedor.

4.2 Implementación funcional. Clases

Symfony

- Controller
 - Alimento
 - ConsumoDia
 - Ejercicio
 - Enlace
 - Peso
 - Recetas
 - Usuario
 - UsuarioRealizaEjercicio
- Entity

- Alimento
- ConsumoDia
- Ejercicio
- Enlace
- Peso
- Recetas
- Usuario
- UsuarioRealizaEjercicio
- Repository
 - Alimento
 - ConsumoDia
 - Ejercicio
 - Enlace
 - Peso
 - Recetas
 - Usuario
 - UsuarioRealizaEjercicio
 -
- Util
 - Cbbdd
 - Respuesta

Angular

- Components
 - comidas-home
 - ejercicios-home
 - form-actualizar-datos-usuario
 - form-anadir-peso
 - form-consumir
 - form-delete-peso
 - form-editar-peso
 - form-eliminar-consumo-diario
 - form-proponer-alimento-nutrientes
 - form-proponer-alimento
 - formulario-tabs
 - home
 - login
 - peso-grafica
 - pesos-home
 - registro
 - registrodatos
 - resumen-alimento
 - tabla-comidas
 - tabla-ejercicios
 - tabla-pasos

- tabs-comida-ejercicio
 - toolbar
 - usuario-perfil
 - vista-alimento
- Models
 - alimento
 - peso
 - usuario
- Services
 - alimentos
 - auth
 - ejercicios
 - peso
 - usuario

4.3 Implementación del modelo de datos. Tablas

Tras la realización de la base de datos he obtenido las tablas:

- Usuario
- Alimentos
- Recetas
- ConsumoDia
- Ejercicios
- EnlacesEjercicios
- UsuariorealizaEjercicios
- Peso

5. Pruebas

5.1. Pruebas de módulos

Ya que no he podido realizar al completo la página web de Angular y se han quedado partes de la API sin uso he decidido en este primer punto mostrar algo de lo que no he llegado a darle uso.

Prueba numero 1:

Esta prueba ha consistido en la creación de una nueva receta mediante una petición POST al servidor Symfony.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8000/recetas/crear
- Status:** 200 OK
- Size:** 426 Bytes
- Time:** 25 ms
- Body (Request):**

```
{
  "nombre": "Macarrones con carne",
  "descripcion": "Los macarrones con carne que me ha hecho mi abuela siempre que tanto me gustan",
  "instrucciones": "Hervir los macarrones al punto y hacer la carne picada con tomate, echarle queso y gratinar",
  "cantidadFinal": 600,
  "grasas": 300,
  "carbohidratos": 500,
  "proteinas": 200,
  "azucares": 0,
  "vitaminas": 30,
  "calorias": 1000,
  "imagen": "imagen en base 64 no disponible",
  "idUser": "5"
}
```
- Body (Response):**

```
{
  "code": 200,
  "respuesta": {
    "nombre": "Macarrones con carne",
    "descripcion": "Los macarrones con carne que me ha hecho mi abuela siempre que tanto me gustan",
    "instrucciones": "Hervir los macarrones al punto y hacer la carne picada con tomate, echarle queso y gratinar",
    "cantidadFinal": 600,
    "proteinas": 200,
    "grasas": 300,
    "carbohidratos": 500,
    "azucares": 0,
    "vitaminas": 30,
    "calorias": 1000,
    "imagen": "imagen en base 64 no disponible",
    "id": 6
  }
}
```

Prueba numero 2:

Esta prueba ha consistido en realizar una modificación a la receta creada anteriormente mediante una petición PUT.

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:8000/recetas/editar
- Status:** 200 OK
- Size:** 444 Bytes
- Time:** 34 ms
- Body (Request):**

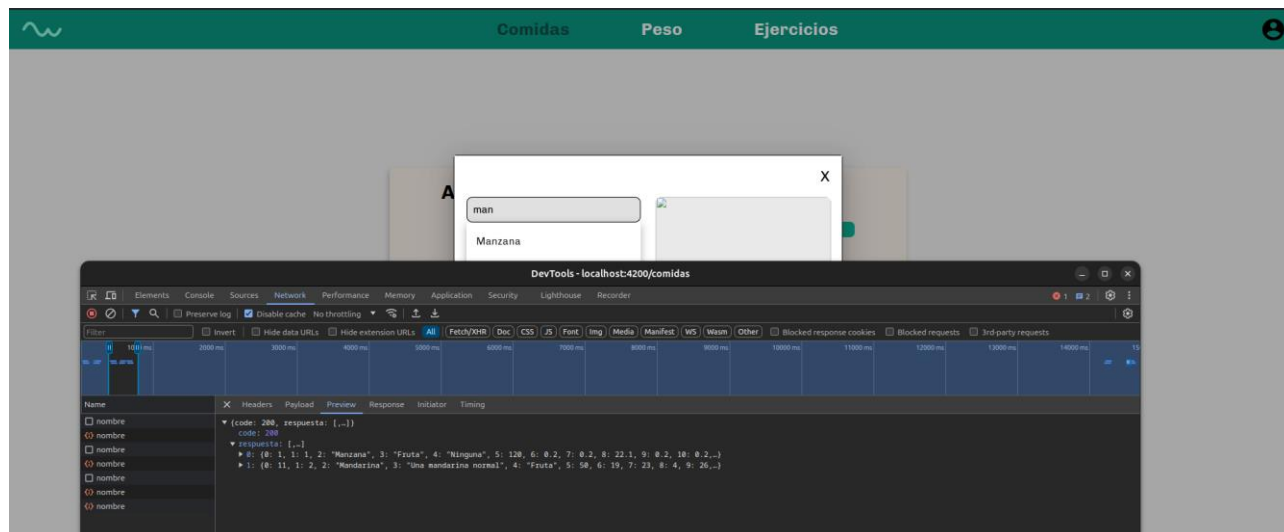
```
{
  "id": "6",
  "nombre": "Macarrones con carne y queso gratinado",
  "descripcion": "Los macarrones con carne que me ha hecho mi abuela siempre que tanto me gustan",
  "instrucciones": "Hervir los macarrones al punto y hacer la carne picada con tomate, echarle queso y gratinar",
  "cantidadFinal": 600,
  "grasas": 300,
  "carbohidratos": 500,
  "proteinas": 200,
  "azucares": 0,
  "vitaminas": 30,
  "calorias": 1000,
  "imagen": "imagen en base 64 no disponible",
  "idUser": "5"
}
```
- Body (Response):**

```
{
  "code": 200,
  "respuesta": {
    "nombre": "Macarrones con carne y queso gratinado",
    "descripcion": "Los macarrones con carne que me ha hecho mi abuela siempre que tanto me gustan",
    "instrucciones": "Hervir los macarrones al punto y hacer la carne picada con tomate, echarle queso y gratinar",
    "cantidadFinal": 600,
    "proteinas": 200,
    "grasas": 300,
    "carbohidratos": 500,
    "azucares": 0,
    "vitaminas": 30,
    "calorias": 1000,
    "imagen": "imagen en base 64 no disponible",
    "id": 6
  }
}
```

5.2. Pruebas de integración

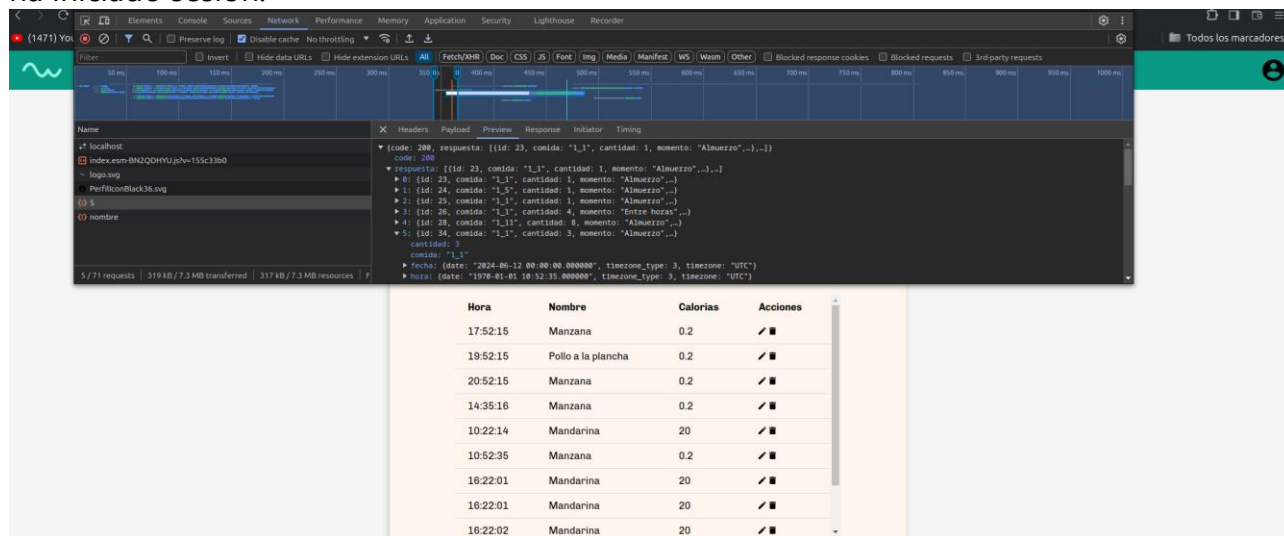
Prueba numero 1:

En esta prueba muestro como se integra la api con angular mediante la realización de peticiones a la api conforme el usuario escribe en la barra de búsqueda. Se ve como devuelve todos los alimentos que encuentra que contienen la cadena escrita por el usuario.



Prueba numero 2:

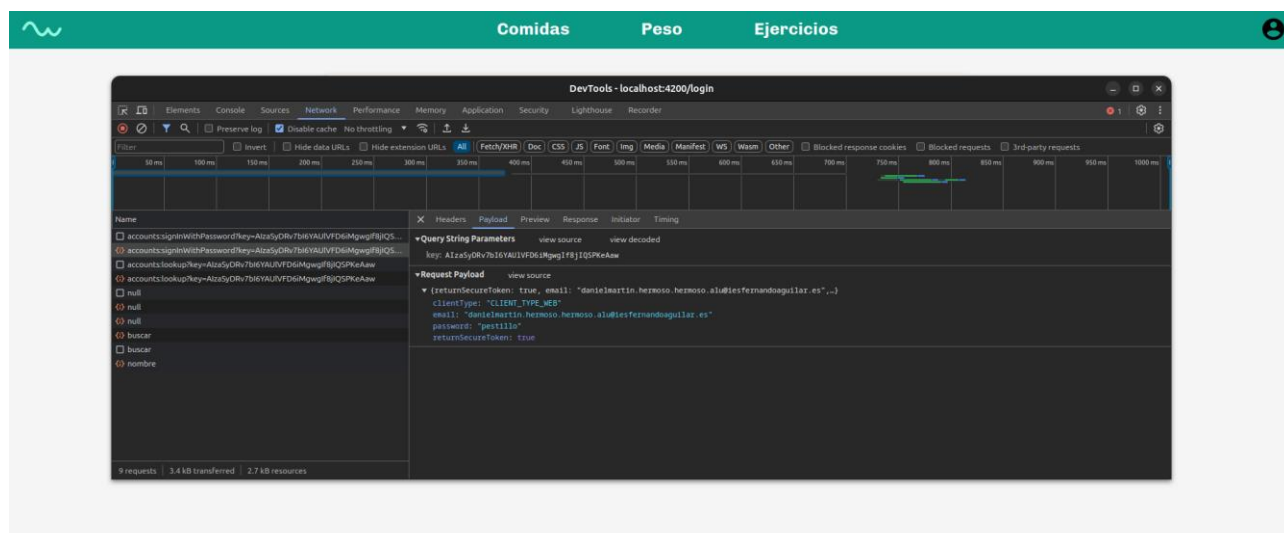
En esta prueba muestro como la api devuelve los alimentos consumidos por el usuario que ha iniciado sesión.



5.3 Pruebas del sistema

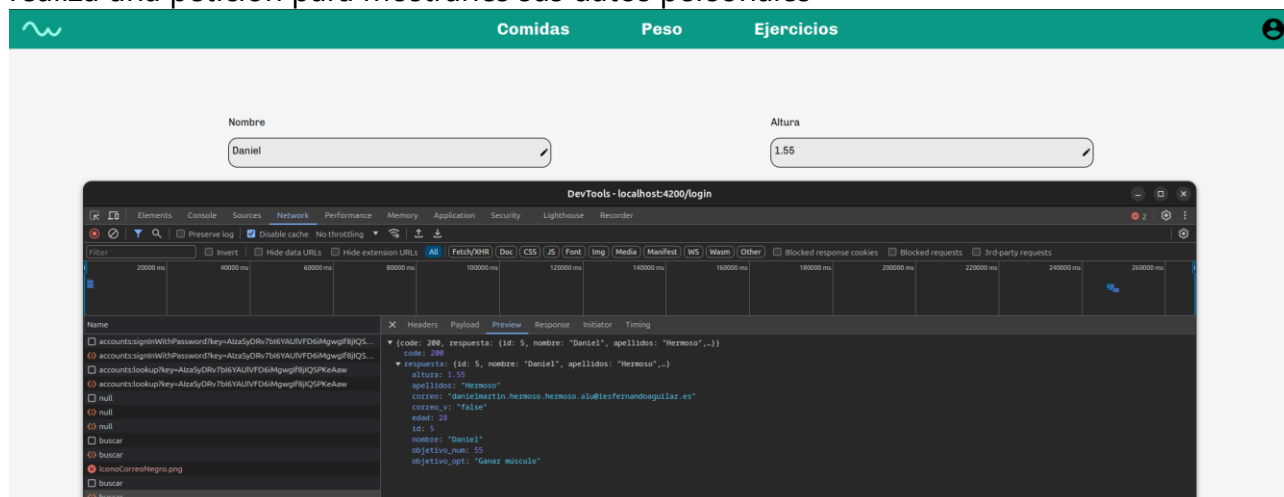
Prueba numero 1:

En la siguiente imagen muestro como a la hora de realizar la acción de iniciar sesión el sistema se comunica con Firebase para comprobar los datos con los que el usuario intenta iniciar sesión.



Prueba numero 2:

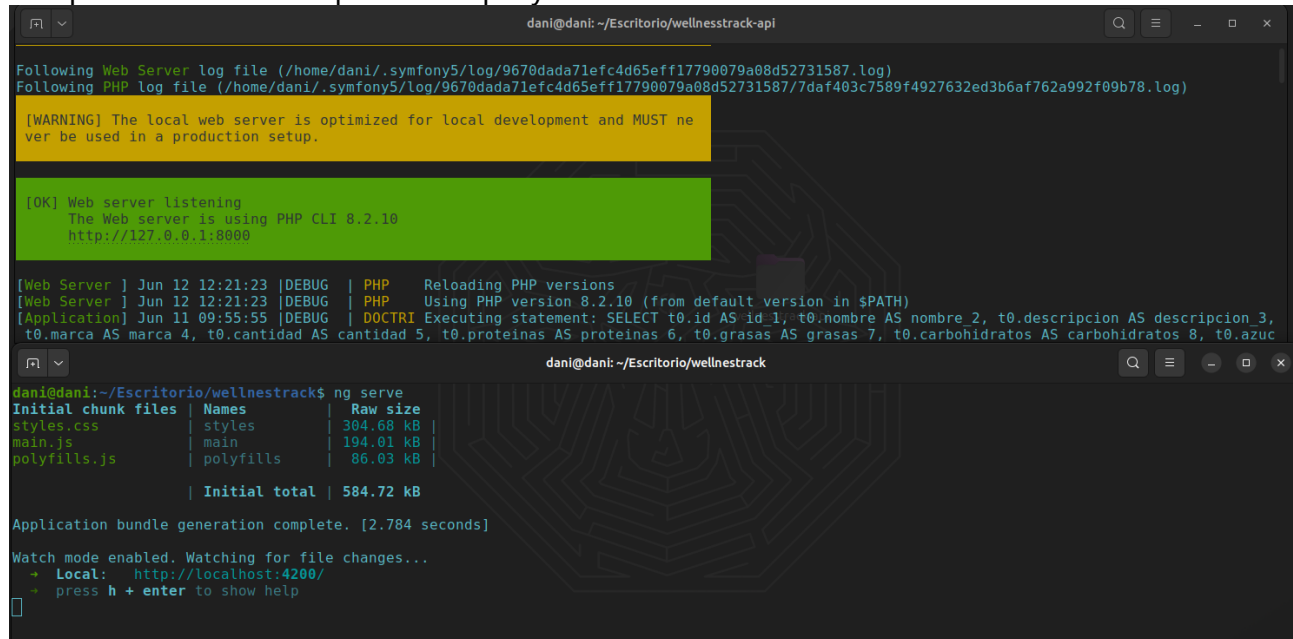
Aquí se muestra como una vez un usuario que ya ha iniciado sesión accede a su perfil se realiza una petición para mostrarles sus datos personales



5.4 Pruebas de instalación

Prueba número 1 y numero 2:

En la siguiente imagen se pueden ver las dos consolas al ejecutarse los diferentes comandos para lanzar ambas partes del proyecto.



The image shows two terminal windows. The top window is titled 'dani@dani: ~/Escritorio/wellnestrack-api' and displays the following output:

```
Following Web Server log file (/home/dani/.symfony5/log/9670dada71efc4d65eff17790079a08d52731587.log)
Following PHP log file (/home/dani/.symfony5/log/9670dada71efc4d65eff17790079a08d52731587/7daf403c7589f4927632ed3b6af762a992f09b78.log)

[WARNING] The local web server is optimized for local development and MUST never be used in a production setup.

[OK] Web server listening
The Web server is using PHP CLI 8.2.10
http://127.0.0.1:8000

[Web Server ] Jun 12 12:21:23 |DEBUG | PHP      Reloading PHP versions
[Web Server ] Jun 12 12:21:23 |DEBUG | PHP      Using PHP version 8.2.10 (from default version in $PATH)
[Application] Jun 11 09:55:55 |DEBUG | DOCTRINE Executing statement: SELECT t0.id AS id_1, t0.nombre AS nombre_2, t0.descripcion AS descripcion_3, t0.marca AS marca_4, t0.cantidad AS cantidad_5, t0.proteinas AS proteinas_6, t0.grasas AS grasas_7, t0.carbohidratos AS carbohidratos_8, t0.azuc...
```

The bottom window is titled 'dani@dani: ~/Escritorio/wellnestrack' and displays the following output:

```
dani@dani:~/Escritorio/wellnestrack$ ng serve
Initial chunk files | Names | Raw size
styles.css | styles | 304.68 kB |
main.js | main | 194.01 kB |
polyfills.js | polyfills | 86.03 kB |
| Initial total | 584.72 kB

Application bundle generation complete. [2.784 seconds]

Watch mode enabled. Watching for file changes...
+ Local: http://localhost:4200/
+ press h + enter to show help
```

6. Conclusiones

6.1 Grado de consecución de los objetivos inicialmente planteados

Los objetivos requeridos en el conjunto del proyecto han sido mayormente conseguidos, en cambio, los objetivos planteados por mí a la hora del planteamiento y diseño del proyecto han estado lejos de poder conseguirse.

Esto debido a falta de tiempo y diversos problemas que he tenido a lo largo de la creación del proyecto.

La no consecución de la totalidad de los objetivos planteados por mí no quita que a lo largo del proyecto no se haya realizado los diversos puntos más importantes del proyecto como pueden ser la conexión y uso de la base de datos desde la web mediante la API, el uso de librerías para la muestra de información de una mejor forma, etc.

6.2 Dificultades encontradas

A lo largo de la realización del proyecto me he encontrado con diversos problemas, algunos de ellos no he sido capaz de resolverlos por mí en su momento, en cambio a base de prueba y error ha sido solventados, otras dificultades sí que han sido resueltas conforme se iban encontrado de forma sencilla.

6.3 Propuestas de mejora y posibles ampliaciones

Desde el momento en el que planteé el proyecto tuve como meta una propuesta de mejora la cual he estado muy cerca de cumplir:

Esta consistía en el uso de una api de terceros que me proporcionaba los datos registrados en relojes y pulseras inteligentes sobre los ejercicios que los usuarios realizaban diferentes entramientos y ejercicios.

Esta propuesta de mejora la vine gestionando desde el día en el que planteé el proyecto, poco tiempo después encontré un tercero que podría proporcionarme los datos que necesitaba con el cual contacte, explique y planteé el uso que quería darle a la api que él podría proporcionarme, con el resultado de llegar a tener la api a mi disposición para su uso durante el tiempo que establecimos.

Esta propuesta de mejora no la he llegado a cumplir debido a la falta de tiempo para el desarrollo de la parte en la que estaría.

Otra mejora que he querido introducir fue la de que el usuario tuviera la posibilidad de elegir entre un modo claro y modo oscuro a la hora de visualizar la página web.

Esta mejora no he podido llevarla a cabo debido a la falta de tiempo y mi dificultad para el diseño de interfaces teniendo que hacer uso de una de mis peores habilidades como es la elección de colores.

7. Bibliografía y recursos on-line

- **Symfony**

- Crear api con Symfony. [Video](#)
- Crear documentación en PHP. [Video](#)
- Crear web con Symfony. [Video](#)
- Creación de Api en Symfony. Canal [Latte And Code](#)
 - Curso de Symfony 5, Capítulos:
 - Configuración del proyecto. [Video](#)
 - Controllers y rutas. [Video](#)
 - Servicios y container. [Video](#)
 - Base de datos e integración con Doctrine. [Video](#)
 - Formularios. [Video](#)
 - Peticiones HttpClient. [Video](#)
- Creación de login con Doctrine. [Video](#)
- Introducción a Symfony. [Video](#)
- Introducción a Doctrine y Entidades. [Video](#)
- Curso sobre Symfony – Repository . [Video](#)
- Curso sobre Symfony – Cud. [Video](#)
- Curso sobre Symfony – Registro de usuarios. [Video](#)
- Curso sobre Symfony – Controlador. [Video](#)
- Curso sobre Symfony – Métodos Mágicos. [Video](#)
- Curso sobre Symfony – Relaciones. [Video](#)
- Introducción e Instalación Symfony. [Video](#)
- Relaciones Symfony y Doctrine. [Video](#)
- Relaciones Doctrine. [Video](#)

- **Angular**

- Uso del componente *Tabs* de Angular Material. [Video](#)
- Uso del componente *Tabs* de Angular Material. [Video](#)

- **Firebase**

- Como usar Firebase. [Video](#)
- Login Angular, Firebase y Bootstrap. [Video](#)
- Email de verificación Firebase. [Video](#)

- **Base de datos**

- Relaciones MySQL. [Video](#)

- **Docker**

- Docker con Apache, PHP, MySQL y PHPMyAdmin. [Video](#)

8. Glosario de términos

8.1 Informáticos

- **Docker:** Plataforma de software que permite a los desarrolladores empacar, enviar y ejecutar aplicaciones en contenedores.
- **MySQL:** Sistema de gestión de bases de datos relacional, utilizado para almacenar y gestionar datos.
- **API:** Interfaz de programación de aplicaciones, que permite la comunicación entre diferentes componentes de software.
- **Symfony:** Framework de desarrollo web PHP utilizado para crear aplicaciones web y API.
- **Symfony CLI:** Herramienta de línea de comandos para trabajar con proyectos Symfony.
- **Doctrine:** ORM (Mapeo Objeto-Relacional) para PHP, utilizado en proyectos Symfony para mapear objetos a tablas de base de datos.
- **Composer:** Gestor de dependencias para PHP, utilizado para administrar las dependencias del proyecto Symfony.
- **Angular:** Framework de desarrollo de aplicaciones web desarrollado por Google, utilizado para crear aplicaciones de una sola página.
- **Angular CLI:** Herramienta de línea de comandos utilizada para iniciar, desarrollar y ejecutar proyectos en Angular.
- **Npm:** Gestor de paquetes para Node.js, utilizado para instalar y administrar paquetes de JavaScript.

8. 2 Problema

- **Alimento:** Cualquier sustancia consumida para proporcionar apoyo nutricional a un ser vivo.
- **Caloría:** Nombre de una unidad de energía que se usa para definir el valor nutricional de los alimentos.
- **Índice de Masa Corporal (IMC):** Medida utilizada para evaluar el peso corporal en relación con la altura, que puede ayudar a determinar si una persona tiene un peso saludable, infrapeso, sobrepeso u obesidad.
- **Proteínas:** Son nutrientes esenciales para el cuerpo humano.
- **Grasas, Carbohidratos y Azúcares:** Son nutrientes que se consumen para dar energía al organismo humano.
- **Vitaminas y Minerales:** Son compuestos esenciales para el funcionamiento adecuado del organismo humano
- **Receta:** Conjunto de instrucciones para realizar sobre diferentes alimentos para realizar un plato o bebida.

9. Anexos

Como anexo es entrado el archivo "Guía de Estilos WellnesTrack – Daniel Martin Hermoso Hermoso.pdf"