



UNIVERSIDAD DE MÁLAGA



Graduado en Ingeniería del Software

Aplicación web para la gestión de menús para personas con intolerancias alimenticias

Web application for the management of menus for people with food intolerances

Realizado por  
Guillermo Eduardo Aguado Moleón

Tutorizado por  
Eduardo Guzmán de los Riscos

Departamento  
Lenguajes y Ciencias de la Computación  
UNIVERSIDAD DE MÁLAGA

MÁLAGA, septiembre de 2022



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADO EN INGENIERÍA DEL SOFTWARE

**Aplicación web para la gestión de menús para personas  
con intolerancias alimenticias**

**Web application for the management of menus for  
people with food intolerances**

Realizado por  
**Guillermo Eduardo Aguado Moleón**

Tutorizado por  
**Eduardo Guzmán De los Riscos**

Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, SEPTIEMBRE DE 2022

Fecha defensa: septiembre de 2022



UNIVERSIDAD  
DE MÁLAGA



# Agradecimientos

A todos los amigos que me han acompañado en estos 4 años de carrera. Tanto a los que llevan desde el principio como los más nuevos, ya que gracias a sus ánimos y apoyo he podido seguir adelante, aunque me encontrara con problemas en el camino.

Agradecer también a mis padres, mi hermana y mi mascota por haber sido mis pilares durante todos estos años, tanto animándome cuando las cosas no pintaban bien como celebrando los logros a lo largo de esta etapa. Además, me gustaría darle un agradecimiento especial a mi hermana porque además de todo esto también me ha enseñado muchas cosas importantes a lo largo de los años.

Y por último, a mi tutor Eduardo Guzmán, tanto por haberme motivado a hacer un trabajo de final de grado orientado al desarrollo web con su entusiasmo y pasión por su labor como docente, como por todo el apoyo que me ha brindado durante este proyecto, sin el cual no me hubiera sido posible llevar esta aplicación a cabo.

Muchas gracias a todos, sin vosotros no podría haber llegado hasta aquí.

# Resumen

Actualmente vivimos en una sociedad con un alto nivel de personas afectadas por intolerancias alimenticias (celiaquía, intolerancia a la fructosa, intolerancia a la lactosa...), pero en la que no contamos con el tiempo necesario para poder preocuparnos por nuestra alimentación y establecer una organización controlada para paliar sus efectos. Con este trabajo se ha buscado proporcionar a la sociedad una herramienta con la que poder gestionar menús del día a día, de una forma rápida y segura, para todas las personas que padecen algún tipo de intolerancia alimenticia.

Para llevar esto a cabo se ha desarrollado una aplicación web de página única con un enfoque especial en el diseño y desarrollo de la interfaz y a la experiencia que tendrán los usuarios al usarla. A nivel tecnológico se ha hecho uso de un *framework* de Javascript, React con Nextjs, para el *frontend*, y de uno de Python, Flask, para el *backend*.

Para lograr conseguir los objetivos la aplicación desarrollada, MyWeeklyMenus, ofrece un catálogo de comidas amplio y diversos con información relativa a alérgenos sobre el que los usuarios pueden hacer búsquedas y filtrados. Además, se pone a disposición de ellos funcionalidades para crear menús semanales con el número de comidas diarias que el usuario necesite y la posibilidad de añadir los alimentos que desee a este.

**Palabras clave:** Alérgenos, Nutrición, Experiencia de usuario, Next.js, Flask.

# Abstract

We currently live in a society with a high level of people affected by food intolerances (celiac disease, fructose intolerance, lactose intolerance...), but in which we don't have the necessary time to worry about our diet and establish a controlled organization to mitigate their effects. This work is intended to provide society with a tool to manage daily menus, in a fast and safe way, for all people who suffer from any kind of food intolerance.

To achieve this, a single page web application has been built with special focus on the design and development of the interface and the experience that users will have when using it. At a technological level React with Nextjs, a Javascript framework, has been used for the frontend, and Flask, a Python framework, has been employed for the backend.

To achieve the objectives, the developed application, MyWeeklyMenus, offers a wide and diverse food catalog with information related to allergens on which users can search and filter. In addition, functionalities are available to create weekly menus with the number of daily meals that the user needs and the possibility of adding the desired foods to it.

**Keywords:** Allergens, Nutrition, User experience, Next.js, Flask.



# Índice

<b>Agradecimientos</b> .....	<b>1</b>
<b>Resumen</b> .....	<b>2</b>
<b>Abstract</b> .....	<b>1</b>
<b>Índice</b> .....	<b>1</b>
<b>Introducción</b> .....	<b>1</b>
1.1 Motivación .....	1
1.2 Objetivos .....	2
1.3 Estructura de la memoria .....	3
<b>Análisis de la situación actual</b> .....	<b>5</b>
2.1 Estado del arte.....	5
2.1.1 Aplicaciones similares.....	5
2.1.2 APIs Similares.....	8
2.1.3 Conclusiones.....	10
<b>Especificación y análisis</b> .....	<b>14</b>
3.1 Requisitos funcionales.....	14
3.2 Requisitos no funcionales.....	14
3.3 Casos de uso .....	15
3.4 Diagramas de secuencia .....	18
<b>Análisis y diseño del almacenamiento</b> .....	<b>21</b>
4.1 Análisis .....	21
4.1.1 Base de datos relacional .....	21
4.1.2 Base de datos no relacional.....	22
4.1.3 Conclusión .....	24
4.2 Diseño .....	24
<b>Tecnologías y Herramientas</b> .....	<b>26</b>
5.1 Introducción .....	26
5.2 Figma.....	26
5.3 Python.....	26
5.4 Flask .....	27
5.5 MongoDB .....	27
5.6 HTML5.....	27
5.7 CSS .....	27
5.8 JavaScript .....	27
5.9 React .....	27
5.10 Next.js .....	28
5.11 GitHub .....	28
5.12 Visual Studio Code .....	28
5.13 MagicDraw .....	28

5.14 Insomnia .....	28
<b>Diseño de la interfaz y experiencia de usuario .....</b>	<b>29</b>
6.1 Introducción .....	29
6.2 Persona .....	29
6.3 Prototipo de baja fidelidad.....	30
6.4 Prototipo de alta fidelidad .....	38
<b>Implementación .....</b>	<b>44</b>
7.1 Introducción .....	44
7.2 Almacenamiento de datos .....	44
7.3 Desarrollo de la API de datos .....	45
7.4 Desarrollo del frontend de la aplicación .....	45
<b>Conclusión y trabajos futuros.....</b>	<b>49</b>
8.1 Introducción .....	49
8.2 Conclusiones y objetivos cumplidos.....	49
8.3 Dificultades encontradas.....	50
8.4 Trabajos futuros .....	50
<b>Referencias.....</b>	<b>52</b>
<b>Manual de Instalación.....</b>	<b>57</b>
Requerimientos:.....	57
Instalación base de datos .....	57
Instalación del backend .....	57
Instalación del frontend .....	59
Despliegue de la aplicación.....	59
<b>Manual de usuario .....</b>	<b>60</b>
Introducción.....	60
Página principal de la aplicación .....	60
Página de Mis menús.....	62
Página información de un menú .....	65
Página de Comidas .....	67
Página de información de una comida.....	69

# 1

## Introducción

### 1.1 Motivación

Actualmente vivimos en una sociedad en la que tanto las intolerancias alimenticias y las alergias son un tema de actualidad sobre el que se realizan muchas investigaciones y descubrimientos cada día. En este sentido, cabe destacar que el número de personas que se ven afectadas por estas es muy elevado y que cada vez va a más como se especificó en (Fiocchi & Fierro, 2017). En 2017, la Organización Mundial de Alergias dictaminó que el número de afectados había aumentado en un 2,5% aproximadamente de la población mundial, aunque es difícil concretar un número concreto debido a que los análisis para identificarlas se hacen de forma diferente en cada país. Algunos estudios (Tang & Mullins, 2017) indican que la prevalencia puede ser de un total del 10% de la población. Otro problema adicional es que de momento no existen curas por lo que son limitaciones con las que estas personas deben a priori acarrear y convivir durante todas sus vidas.

¿Y qué se considera una alergia alimenticia? (Sicherer, 2011) Cuando después de ingerir una comida aparecen efectos nocivos hablamos de una alergia alimenticia. Hay muchos tipos de respuestas del cuerpo, siendo la más común de ella la del anticuerpo IgE que produce reacciones inmediatas instantáneas. También existen otros tipos de respuestas con retardo como problemas digestivos o problemas de dermatitis. Además, hay diferentes grados de reacciones alérgicas siendo la anafilaxis la más peligrosa de todas debido a su carácter letal.

¿Y en qué se diferencian estas de las intolerancias alimenticias? (Sicherer, 2011) Aunque ambas se parecen en que se deben a la ingestión de comida, afectan al individuo solo en una pequeña manera concreta, donde las alergias son causadas por respuestas inmunes, y las intolerancias por cambios en el metabolismo o la digestión, debido a déficit enzimático o a una susceptibilidad de los sujetos a componentes básicos de las comidas. Un buen ejemplo para representar esto podría ser la diferencia entre la alergia

y la intolerancia a la "leche", mientras que la intolerancia se debe a una falta de encima de la lactosa, la alergia a la leche es causada porque los individuos han desarrollado anticuerpos IgE contra algunas de las proteínas que contiene la leche de forma natural.

Sin embargo, a pesar de la importancia que hemos visto que tienen estos problemas en las vidas de las personas que los padecen, debido al ritmo de vida actual, la tarea de organizarse un menú adecuado se complica demasiado. Las personas que tienen estos problemas no suelen tener el tiempo necesario para dedicarse a hacer un menú correcto que cumpla sus necesidades, llevando a veces a que este tipo de colectivo lleve una dieta monótona, lo que puede traducirse en una dieta insana.

Por otro lado, la popularidad de las aplicaciones web está incrementando debido a su versatilidad y accesibilidad en diversas plataformas permitiendo que cualquier persona con acceso a Internet pueda hacer uso de ellas fácilmente. Es por ello que su uso se está popularizando tanto para aplicaciones de ocio como para aplicaciones de carácter laboral. Debido a este incremento en el uso de este tipo de aplicaciones, se ha desarrollado el concepto de "Single-Page Application" (Arranz Santamaria, 2015) en el cual todo el contenido se encuentra en una única página otorgándole al usuario una experiencia más fluida y similar a una aplicación de escritorio.

Por lo que con este trabajo de fin de grado, se busca realizar una aplicación web del tipo *Single-Page Application* (SPA) con la cual poder gestionar sus menús de una forma rápida y segura, en caso de padecer algún tipo de intolerancia alimenticia.

## 1.2 Objetivos

El desarrollo de este trabajo de fin de grado tiene como objetivo principal la elaboración de una aplicación SPA para que los usuarios pueden elaborar sus propios menús de acuerdo a sus necesidades entre un catálogo de comidas ofrecido. Para ello se establecen los siguientes subobjetivos necesarios para alcanzar este:

- Primero se estudiará el estado del arte de aplicaciones de carácter similar o de fin similar, para hacer un análisis de cuáles son las cualidades más importantes que debería tener la aplicación.
- Se investigará y analizará cuál será el público al que se desea dirigir la aplicación para desarrollar una experiencia de usuario de acuerdo a este.
- Estudio y análisis de cuál es el método de almacenamiento más adecuado para el caso de uso en concreto con el que se va a trabajar, además de realización del diagrama de la base de datos.
- Diseño de dos tipos de mockup que consistirán en:
  - Prototipo de baja fidelidad: Desarrollo de un mockup simple que no se centrará en los estilos de la aplicación web sino en la navegabilidad y la funcionalidad que deberá tener esta.
  - Prototipo de alta fidelidad: En esta sección se realizará un prototipo cuya función es reproducir cómo será la apariencia final de la aplicación, además de hacer un análisis de la usabilidad y la experiencia de esta.

- Desarrollo de una API REST de datos de comida mediante la elaboración de los diferentes métodos de petición HTTP habituales en estas para realizar CRUD de las tablas o colecciones que se estimen necesarios, junto con los diferentes filtrados o búsquedas específicas del catálogo de alimentos. Además se elaborará archivo (por ejemplo, CSV o JSON) o un script que permita la carga masiva de información a la base de datos. La idea es facilitar el proceso de introducir y actualizar toda la información referente a comidas que contendrá la aplicación, a partir de fuentes externas.
- Y, por último, el desarrollo funcional de la aplicación para lo que se establecerán una serie de funcionalidades clave a desarrollar, para adecuarse de esta manera a la metodología ágil que se busca seguir. Estas funcionalidades consisten en:
  - Opción de creación de menús con diferentes comidas al día: Permitir que un usuario pueda crear un menú con el número de comidas diarias que necesite, seleccionando todos los platos que contendrá este.
  - Diferentes búsquedas y filtrados de catálogo de comidas: Creación de diferentes filtrados y búsquedas para que el usuario pueda realizar una búsqueda más concreta del catálogo de alimentos por factores como el nombre o alérgenos.
  - Implementación de un registro de usuarios: Implementar un sistema de inicio sesión en la aplicación de manera que la información que este desee almacenar se quede relacionado con, por ejemplo, una dirección de correo o unas credenciales.

### 1.3 Estructura de la memoria

La redacción de esta memoria se ha dividido en diferentes capítulos para una lectura más cómoda y una mejor estructuración. A continuación, se enunciarán los diferentes capítulos de forma ordenada, junto a una breve descripción de lo que se abordará en cada uno de ellos:

#### 1. Introducción:

En este capítulo se explicará la motivación por la que sea realiza este trabajo de final de grado junto con la redacción de los objetivos que se pretenden abarcar en este proyecto y, por último, se explica la estructura que seguirá la memoria.

#### 2. Análisis de la situación actual:

En este capítulo se estudiará el estado del arte de aplicaciones y APIs de carácter o finalidad similar, junto con análisis del público objetivo al que se pretende dirigir la aplicación.

#### 3. Especificación y análisis:

En esta parte de la memoria se enumerarán y explicarán todos los requisitos funcionales y no funcionales que se buscan solventar en este trabajo, además de desarrollar diagramas UML para definir y especificar los casos de uso que deberá tener el sistema.

#### 4. Análisis y diseño del almacenamiento:

Aquí se elaborará un análisis y estudio de cuál es el método de almacenamiento más apropiado para este caso de estudio, y se realizará un diseño de la base de datos que utilizará en este.

#### 5. Tecnologías:

En esta parte se explicarán y mostrarán las tecnologías que se utilizarán para la elaboración de esta aplicación web.

**6. Diseño de la interfaz y experiencia de usuario:**

En este capítulo se hablará sobre los dos prototipados que se deberán realizar para el diseño de la interfaz de usuario de la aplicación. Primero se empezará por el prototipo de bajo fidelidad y luego se pasará al de alta fidelidad basado en este anterior, junto con la explicación de las diferentes medidas que se han considerado necesarias para que la experiencia de usuario sea lo más adecuada posible.

**7. Implementación:**

En este apartado se llevará a cabo todo el desarrollo software de la aplicación web, desde la elaboración de la API REST hasta la construcción del front-end de acuerdo con los prototipos del capítulo anterior, además de la explicación y argumentación de los cambios que se hayan visto necesarios realizar.

**8. Conclusión y Trabajos Futuros:**

En este último capítulo se hablará de los objetivos cumplidos juntos las dificultades que se han encontrado para llevar a cabo la realización de estos. Por último, también se hará una reflexión de las posibles líneas futuras y cambios que se podrían aplicar al proyecto.

# 2

## Análisis de la situación actual

### 2.1 Estado del arte

A continuación se hará un estudio de aplicaciones y APIs que se encuentren actualmente disponibles para el público que cumplen o pueden llegar a cumplir una función similar a lo que se busca lograr con este trabajo de fin de grado.

#### 2.1.1 Aplicaciones similares

- **Notion**  
Notion (Notion, 2022) es una aplicación de organización a nivel general no especializada en comidas como el resto de la lista la cual destaca por ser altamente personalizable, permitiendo al usuario tener diferentes páginas para organizarse con etiquetas y temas personalizables. A pesar de no estar pensada específicamente para gestionar menús debido a su alto nivel de personalización podría utilizarse para ello, aunque no contaría con ningún tipo de base de datos por lo que todos los alimentos deberían introducirse de forma manual.
- **Nutrition Addition**  
*Nutrition Addition* (Nutrition Addition, 2022) es una aplicación de la cual no se ha podido obtener mucha información debido a que no cuenta con una interfaz muy intuitiva y todo su contenido es de pago sin disponibilidad de *preview*. Solo cuenta con soporte en inglés pero parece ofrecer muchas funcionalidades para un menú como son los filtros y

análisis nutricionales. Parece estar enfocada únicamente a restaurantes y a dispositivos móviles.

- **Nestlé Menú Planner**

*Nestlé Menú Planner* (Nestlé, 2022) es una iniciativa de la empresa Nestlé cuyo fin es ayudar a las personas a llevar menús más saludables en su día a día de una forma fácil y cómoda, entre sus funcionalidades destaca:

- Posibilidad de descargar los menús en formato PDF, donde se dispondrá el menú con iconos simplificados por categoría de alimentos.
- Todas las semanas se ofrece un menú (con adaptación vegetariana) sano para los usuarios lo puedan seguir sin tener que preocuparse de crear ellos su propio menú.
- Disponen de una sección para visualizar el menú del día actual de una forma más detallada, permitiéndote ver por cada alimento la siguiente serie de características:
  - Nombre.
  - Foto.
  - Dificultad de la elaboración.
  - Número de personas para las que están pensadas las cantidades de la receta.
  - Ingredientes.
  - Preparación.
  - Consejo nutricional.
  - Pequeño análisis nutricional simplificado.
  - Cantidad de comida que poner según la edad.
- Herramienta para recomendar cenas con la cual dados uno o dos ingredientes principales te sugiera comidas que cumplan esas especificaciones.
- Buscador de recetas por nombres, con filtros por:
  - Primer y segundo plato, plato único y postre.
  - Grupos de alimentos.
- Guía para hacerte un menú equilibrado de una forma sencilla.

A pesar de contar con tantas funcionalidades no te ofrece la posibilidad de crearte tu propio menú customizado dentro de la herramienta, ni tampoco contiene soporte para más idiomas además de castellano.

- **Easy Menu Planner**

*Easy Menu Planner* (Easy Menu Planner, 2022) es una aplicación de dispositivos móviles para la gestión de menús de la que se pueden destacar las siguientes características:

- Permite la sincronización entre varios dispositivos.
- La aplicación genera automáticamente una lista de la compra de los ingredientes que necesites para la comida de esa semana.
- Ofrece un organizador simple de comidas a nivel semanal con tres comidas al día (desayuno, almuerzo y cena).
- Ofrece la posibilidad de hacer una planificación del mes actual.

- La aplicación dispone de una pequeña base de datos de alimentos, pero ofrece la posibilidad de añadir manualmente platos que no se encuentren dentro de esta base de datos.
  - Dentro de cada comida se encuentra la siguiente información personalizable:
    - Nombre.
    - Ingredientes.
    - Etiquetas.
    - Notas
  - También dispone de la posibilidad de añadir un widget a la pantalla principal de tu teléfono móvil tanto con el menú semanal como con la lista de la compra.
- **Ekilu**  
*Ekilu* (Enkilu, 2022) es una aplicación sobre la comida y la vida saludable de la que podemos destacar las siguientes cualidades:
    - Cuenta con soporte tanto para castellano como para inglés.
    - Se encuentra disponible tanto en formato de aplicación móvil como en formato de aplicación web, siendo esta un poco más limitada que la otra función.
    - Contiene función de lista de la compra.
    - Ofrece una sección con ideas de platos para cocina de diferentes categorías.
    - Buscador de comidas por ingredientes.
    - Dispone de una pestaña con consejos sobre nutrición, ejercicio, sostenibilidad...
    - Dentro de cada alimento se ofrece la siguiente información:
      - Imagen.
      - Nombre.
      - Puntuación de usuarios.
      - Pasos (Elaboración).
      - Trucos y consejos.
      - Etiquetas.
      - Dificultad.
      - Ingredientes (Este atributo se puede ajustar al número de comensales para los que se desee realizar la comida).
      - Tiempo de elaboración.

Sin embargo a pesar de contar con un abanico tan amplio de funcionalidades, muchas otras se esconden detrás de la versión premium de la aplicación, estando entre estas las siguientes:

    - Menú semanal elaborado por un nutricionista profesional.
    - Información nutricional de cada comida.
    - Lista de la compra elaborada de forma automática.
    - Categorías premium de recetas.
    - Otras funcionalidades no relacionadas con la comida.  - Dentro de las categorías únicas de la versión móvil se encuentra la posibilidad de asignar comidas a cada día del mes, pero sin la

opción de indicar de que tipo de comida es esta: desayuno, almuerzo, cena...

- Dispone también de un perfil por cada usuario.

- **Otras aplicaciones similares**

A continuación se hablará de un conjunto de aplicaciones, Planifood (2022); Recipe Calendar (2022); Reciper Keeper (2022) que se han encontrado que por separado no tienen mucha relevancia porque no tienen muchas características que las diferencien del resto, pero que de una forma conjunta sí son interesantes de analizar. Las funcionalidades más importantes a destacar de este conjunto de aplicaciones son:

- Posibilidad de poder compartir recetas al igual que te permite añadir recetas a la aplicación.
- Disponibilidad de las aplicaciones en más de una plataforma.
- Recomendaciones de alimentos basados en tus preferencias.
- Permitir al usuario importar recetas de Internet en la aplicación.
- Funcionalidades para escanear texto de recetarios.

### 2.1.2 APIs Similares

- **TheMealDB**

TheMealDB (TheDataDB, 2022) es una API de comidas y alimentos desarrollada por el equipo de TheDataDB. Cuenta con muchas funcionalidades a las que se puede acceder de forma gratuita, entre estas se encuentran:

- Dentro de cada entrada de comida cuenta con las siguientes características:
  - Nombre en inglés.
  - Categoría.
  - Gastronomía de la que procede.
  - Instrucciones para elaborar esta.
  - Imagen.
  - Ingredientes necesarios para realizarla.
  - Tutorial de YouTube donde se explica cómo hacer la receta.
  - Etiquetas.
  - Información sobre de dónde se ha obtenido la receta.
- Dentro de la API se permiten hacer los siguientes filtrados:
  - Búsqueda por nombre.
  - Filtrado por inicial.
  - Búsqueda por id.
  - Listar todas las categorías de alimentos existentes.
  - Listar todas las categorías, ingredientes y gastronomías.
  - Filtrar por ingrediente principal.
  - Filtrar por categoría.
  - Filtrar por gastronomía.

- Dentro de los filtrados hay algunos que se encuentran limitados solo a la versión premium estos son:
  - Mostrar aleatoriamente 10 alimentos.
  - Mostrar los alimentos añadidos recientemente.
  - Filtrar por más de un ingrediente.
- Funciones para obtener solo la imagen de una comida o un ingrediente.
- Cuenta con una base de datos muy grande y con número de usos limitados a través de API Keys, que se obtienen por medio de pago.
- Proporciona una interfaz gráfica sencilla con la que poder probar todas estas funcionalidades.

A pesar de todo el abanico de características con las que cuenta esta API, no tiene soporte para otros idiomas encontrándose en su totalidad en inglés.

- **Edamam**

Edamam (Ignatiev & Penev, 2022) es una API compleja centrada en alimentación, dentro de esta podemos encontrar tres APIs internas:

- Análisis nutricional a partir de una serie de alimentos.
- Base de datos de comidas basada en el lenguaje natural.
- Buscador de recetas a partir de los siguientes criterios:
  - Palabras claves.
  - Calorías.
  - Ingredientes e intolerancias alimenticias.

Este análisis se centrará principalmente en la última mencionada. Dentro de sus funcionalidades y características se destaca lo siguiente:

- Es necesario registrarse para hacer uso de la API.
- No cuenta con soporte para otros idiomas; esta solo se encuentra disponible en inglés.
- Probar el funcionamiento de la API es complejo, aunque dispone de una interfaz gráfica que simplifica la tarea y proporciona una forma de ver varias de estas funcionalidades.
- Cada receta contiene las siguientes características:
  - Nombre.
  - Imagen.
  - Origen de la receta.
  - Ingredientes calculados para un número de comensales, aparentemente este se puede variar dependiendo del número de comensales pero en las recetas probadas esta funcionalidad no parecía estar disponible.
  - Preparación de la receta, o en su defecto, enlace a la página de donde se ha obtenido esta.
  - Valores nutricionales de una forma muy elaborada.
  - Etiquetas entre las que se encuentran los alérgenos, aunque en la previsualización de la interfaz gráfica si se cambiaba el número de comensales estas parecían dejar de estar visibles.

- (Dentro de la interfaz gráfica) Pequeña gráfica poca intuitiva que aparentemente indica el porcentaje de grasas, carbohidratos y otros valores como proteínas, vitaminas, agua...

### **2.1.3 Conclusiones**

Tras a ver realizado la búsqueda de las aplicaciones similares que hay en el mercado, se puede desarrollar un pequeño análisis de las cualidades comunes que comparten, cuáles no y cuáles podrían considerarse las funcionalidades más importantes a tener en cuenta a la hora de realizar el listado de requisitos funcionales y no funcionales.

Como se puede observar en la Figura 1, entre las funcionalidades de aplicaciones las que más destacan son la disponibilidad en diferentes dispositivos, la sincronización en estos y también la posibilidad de contar con recomendaciones dentro de las comidas. También se puede ver que hay varias funcionalidades que son muy poco comunes en las aplicaciones a pesar de que pueden aportar mucha utilidad como ofrecer versiones o información de que comidas son vegetarianas, contar con una pestaña de perfil dentro de la aplicación, permitirte compartir recetas, etc.

Por otro lado, en la Figura 2 se puede encontrar que es muy importante que las comidas contengan imágenes y formas de obtener estas en las APIs. Sin embargo, respecto a filtros no existe mucha diferencia dentro estos aunque sí se puede apreciar que no es común en ellas ni el control de alérgeno ni el soporte en más de un idioma.

Funcionalidades																			
Aplicaciones	Personalización de la aplicación	Soporte en más de un idioma	Permite descargar menús	Ofrece menús hechos sanos	Version vegetariana	Información detallada de las comidas	Recomienda comidas	Buscador de recetas con filtros	Consejos para hacer menú equilibrado	Múltiples dispositivos	Permite sincronización en varios dispositivos	Lista de la compra	Organizador semanal	Organizador mensual	Base de datos de comidas	Permite añadir comidas	Cuenta con perfil	Permite compartir recetas	Escanner textos recetas/Im portar de Internet
Notion																			
Nutrition Addition																			
Nuestra Menú Planner																			
Easy Menu Planner																			
Ekliv																			
Otras Aplicaciones																			
	Nativo de la aplicación																		
	No le ofrece nativamente la aplicación, pero puede contar con ello manualmente o no está claro																		

Figura 1 Tabla de resumen de funcionalidades de las aplicaciones similares





# 3

## Especificación y análisis

### 3.1 Requisitos funcionales

Tras los análisis previos del dominio sobre el que se centrará el desarrollo de la aplicación que se va a abordar y de la situación actual de esta en el ámbito de las aplicaciones y las APIs, a continuación se va a redactar las listas de requisitos funcionales que se pretenden abordar en esta trabajo de fin de grado.

- RF1. El usuario podrá iniciar sesión
- RF2. El usuario podrá cerrar sesión
- RF3. El usuario podrá crear un menú
  - RF3.1. El usuario podrá elegir el número de comidas diarias que contendrá el menú.
  - RF3.2. El usuario podrá agregar comidas al menú.
  - RF3.3. El usuario podrá remplazar las comidas del menú.
- RF4. El usuario podrá acceder a sus menús.
- RF5. El usuario podrá buscar comidas por el nombre.
- RF6. El usuario podrá filtrar comidas por alérgenos.
- RF7. El usuario podrá filtrar comidas por opción vegetariana o vegana.
- RF8. El usuario podrá filtrar comidas por gastronomía.

### 3.2 Requisitos no funcionales

A continuación, se redactan los requisitos no funcionales:

- RNF1. El sistema deberá utilizar OAuth para las sesiones de los usuarios.

- RNF2. El sistema deberá contar con un diseño responsive para pantallas de ordenador estándar y para pantallas de móvil.

### 3.3 Casos de uso

Nombre	Descripción
Caso de uso	Inicio de sesión
Actores	Usuario estándar
Descripción	El usuario podrá iniciar sesión en la aplicación con su cuenta de Google
Precondiciones	El usuario no ha iniciado sesión y se encuentra en la página de entrada de la aplicación.
Escenario Principal	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el botón de iniciar sesión de la página de entrada.</li> <li>2. El usuario elige la cuenta de Google con la que desea conectarse a la aplicación.</li> <li>3. El usuario accede a la página principal de la aplicación con el estado actualizado.</li> </ol>
Escenario Alternativo	2.b. El usuario cierra la ventana emergente de Google y no inicia sesión.

Nombre	Descripción
Caso de uso	Cerrar sesión
Actores	Usuario estándar
Descripción	El usuario que haya iniciado sesión podrá cerrar esta.
Precondiciones	El usuario se encuentra con sesión iniciada.
Escenario Principal	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el logo de la aplicación en la barra de navegación.</li> <li>2. El usuario accede a la página principal.</li> <li>3. El usuario pulsa sobre el botón de cerrar sesión.</li> <li>4. El usuario accede a la página de entrada de la aplicación.</li> </ol>
Escenario Alternativo	No hay escenario alternativo

Nombre	Descripción
Caso de uso	Crear un menú
Actores	Usuario estándar
Descripción	El usuario que haya iniciado sesión podrá crear un menú.
Precondiciones	El usuario se encuentra con sesión iniciada.
Escenario Principal	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el icono de desplegar de la barra de navegación y a continuación sobre la opción "Mis Menús"</li> </ol>

	<p>2. El usuario accede a la página de "Mis Menús".</p> <p>3. El usuario pulsa sobre el botón de crear menú.</p> <p>4. El usuario accede al formulario para introducir los datos del menú que desea crear.</p> <p>5. El usuario introduce el nombre del menú y selecciona el número de comidas diarias.</p> <p>6. El usuario visualiza que en la lista de mis menús se encuentra el nuevo menú creado.</p>
<b>Escenario Alternativo</b>	<p>5b. El usuario deja el campo de nombre vacío y/o selecciona un valor no válido para el número de comidas diarias.</p> <p>6b. El sistema informa al usuario de que los valores introducidos no son válidos.</p> <p>5c. El usuario pulsa sobre el botón de cancelar y cierra el formulario.</p>

<b>Nombre</b>	<b>Descripción</b>
<b>Caso de uso</b>	Agregar comida a un menú
<b>Actores</b>	Usuario estándar
<b>Descripción</b>	El usuario que haya iniciado sesión podrá añadir comida a un menú creado.
<b>Precondiciones</b>	El usuario se encuentra en sesión iniciada y tiene al menos un menú creado.
<b>Escenario Principal</b>	<p>1. El usuario pulsa sobre el icono de desplegar de la barra de navegación y a continuación sobre la opción "Comidas".</p> <p>2. El usuario accede a la página de "Comidas".</p> <p>3. El usuario pulsa sobre el botón de añadir del alimento que desea agregar.</p> <p>4. El usuario accede al cuadro de diálogo de agregar un alimento.</p> <p>5. El usuario selecciona el menú, el día y el número de la comida donde sea introducir el alimento.</p> <p>6. El usuario pulsa sobre el botón de añadir y sale de la ventana modal.</p>
<b>Escenario Alternativo</b>	<p>6b. El usuario pulsa sobre el botón de cancelar, no añade el alimento y sale de la ventana modal.</p>

<b>Nombre</b>	<b>Descripción</b>
---------------	--------------------

<b>Caso de uso</b>	Acceder a un menú creado
<b>Actores</b>	Usuario estándar
<b>Descripción</b>	El usuario que haya iniciado sesión podrá acceder a la información de un menú creado.
<b>Precondiciones</b>	El usuario se encuentra en sesión iniciada y tiene al menos un menú creado.
<b>Escenario Principal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el icono de desplegar de la barra de navegación y a continuación sobre la opción "Mis Menús"</li> <li>2. El usuario accede a la página de "Mis Menús".</li> <li>3. El usuario pulsa sobre el menú que desea visualizar.</li> <li>4. El usuario accede a la página de información de ese menú.</li> </ol>
<b>Escenario Alternativo</b>	No hay escenario alternativo

<b>Nombre</b>	<b>Descripción</b>
<b>Caso de uso</b>	Buscar comidas
<b>Actores</b>	Usuario estándar
<b>Descripción</b>	El usuario podrá buscar comidas que contengan una cadena de texto introducida.
<b>Precondiciones</b>	No hay precondiciones.
<b>Escenario Principal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el icono de desplegar de la barra de navegación y a continuación sobre la opción "Comidas"</li> <li>2. El usuario accede a la página de "Comidas".</li> <li>3. El usuario pulsa sobre la barra de búsqueda, introduce la cadena de texto que desea que contenga el nombre del alimento y pulsa sobre el botón de buscar.</li> <li>4. El usuario visualiza los alimentos que coinciden con la especificación de la búsqueda.</li> </ol>
<b>Escenario Alternativo</b>	No hay escenarios alternativos.

<b>Nombre</b>	<b>Descripción</b>
<b>Caso de uso</b>	Filtrar comidas
<b>Actores</b>	Usuario estándar
<b>Descripción</b>	El usuario podrá filtrar una comida por su gastronomía o por sus ingredientes (alérgenos y apto para vegetarianos)
<b>Precondiciones</b>	No hay precondiciones.

### Escenario Principal

1. El usuario pulsa sobre el icono de desplegar de la barra de navegación y a continuación sobre la opción "Comidas"
2. El usuario accede a la página de "Comidas".
3. El usuario pulsa sobre el botón de filtro.
4. El usuario accede al desplegable con las opciones de filtrado.
5. El usuario selecciona las gastronomías y los componentes por los que desea filtrar.
6. El usuario pulsa sobre el botón de aplicar filtros y visualiza todos los elementos que coincidan con los criterios del filtro y de la barra de búsqueda.

### Escenario Alternativo

- 6b. El usuario pulsa sobre el botón de cancelar y cierra el desplegable.

## 3.4 Diagramas de secuencia

A continuación, para una mayor comprensión de la interrelación que existe entre las acciones de la aplicación y los objetos dentro del sistema se han desarrollado diagramas de frecuencia de los requisitos con mayor interés a nivel de interacción con el sistema, como se ve en las Figura 3, Figura 4, Figura 5, Figura 6 y Figura 7, para una mayor simplicidad y mejor visibilidad se han supuesto unos prerrequisitos más estrictos en los que el usuario ya se encuentre situado en la página donde se realiza la acción. También con la elaboración de estos diagramas se puede observar que hay algunos casos de uso, como el de búsqueda y filtrado de comida, que son muy parecidos a nivel de acciones lo que puede ayudar luego a la hora de elaborar el desarrollo.

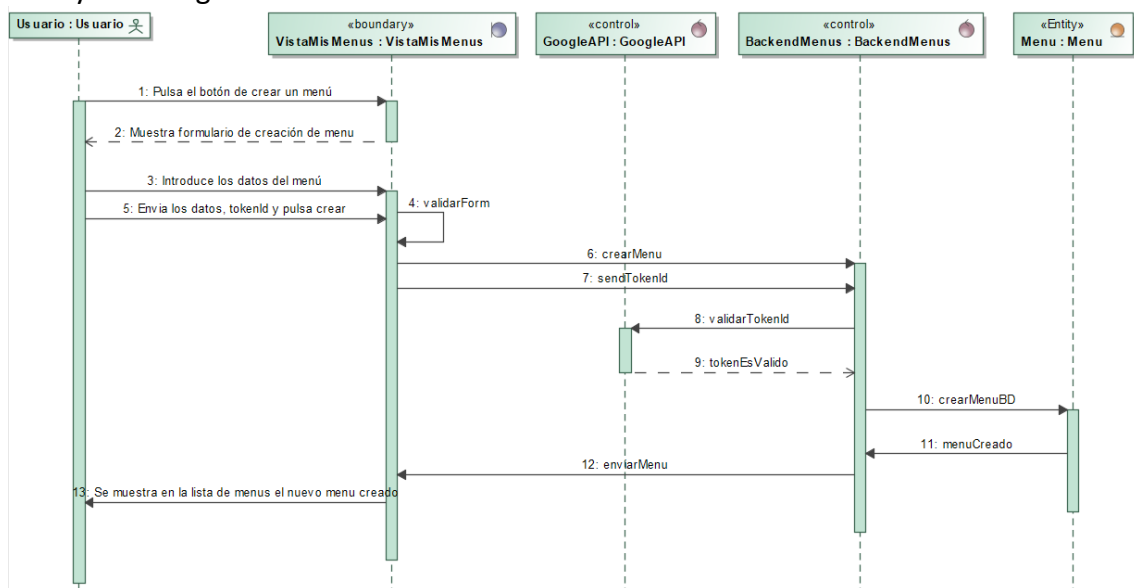


Figura 3 Diagrama de secuencia de Crear un menú

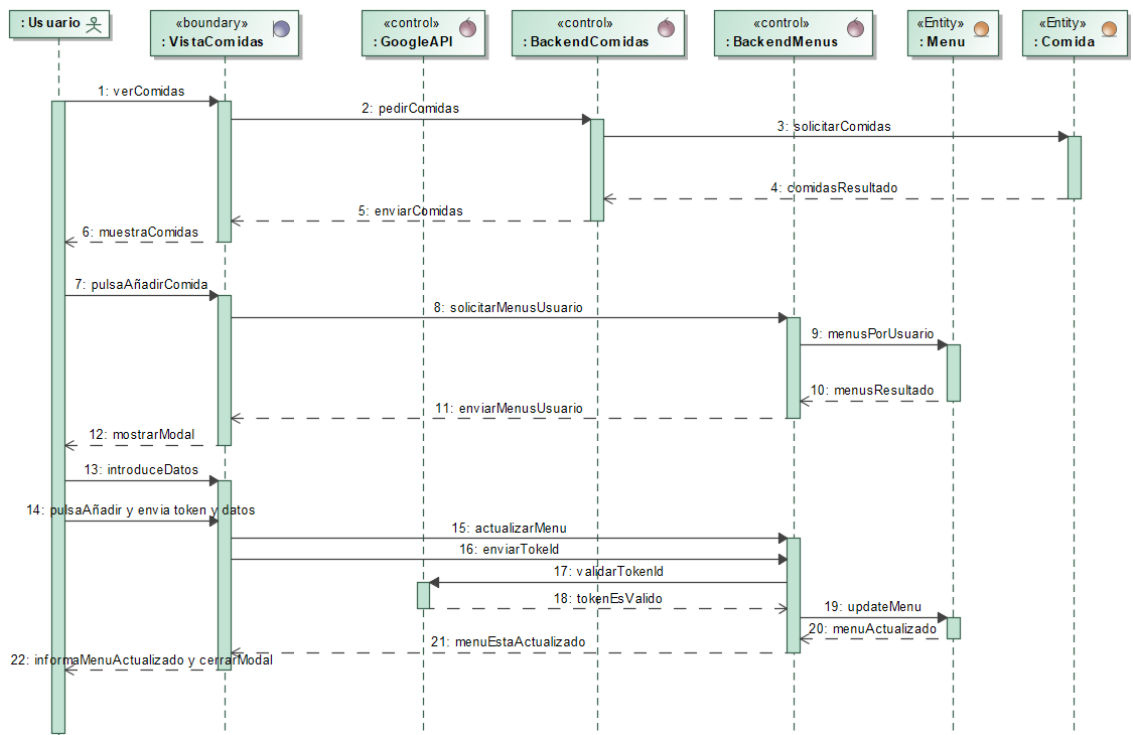


Figura 4 Diagrama de secuencia de Agregar comida a un menú

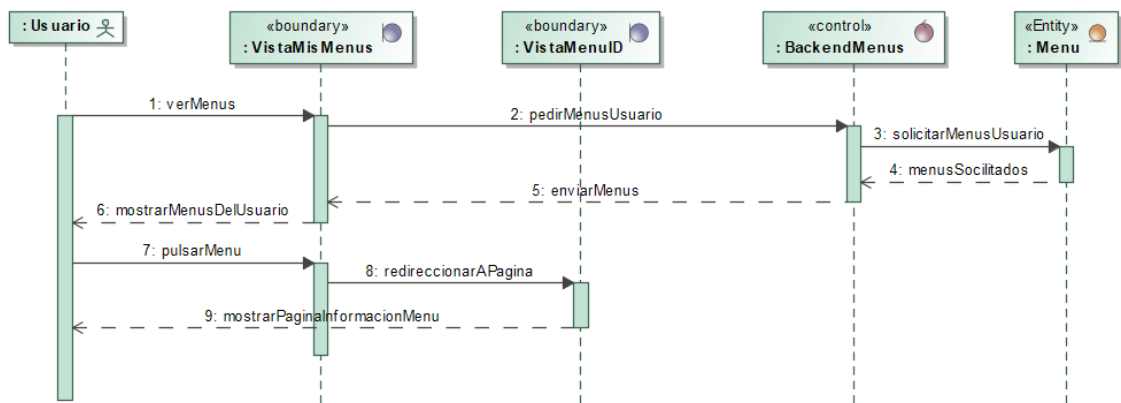


Figura 5 Diagrama de secuencia de Acceder a un menú creado

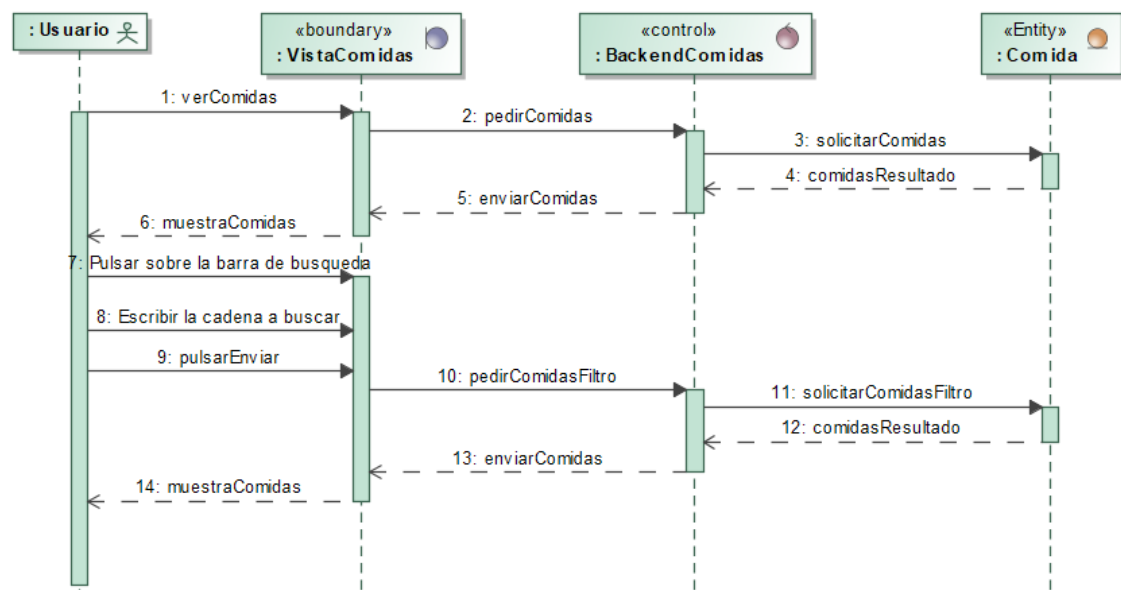
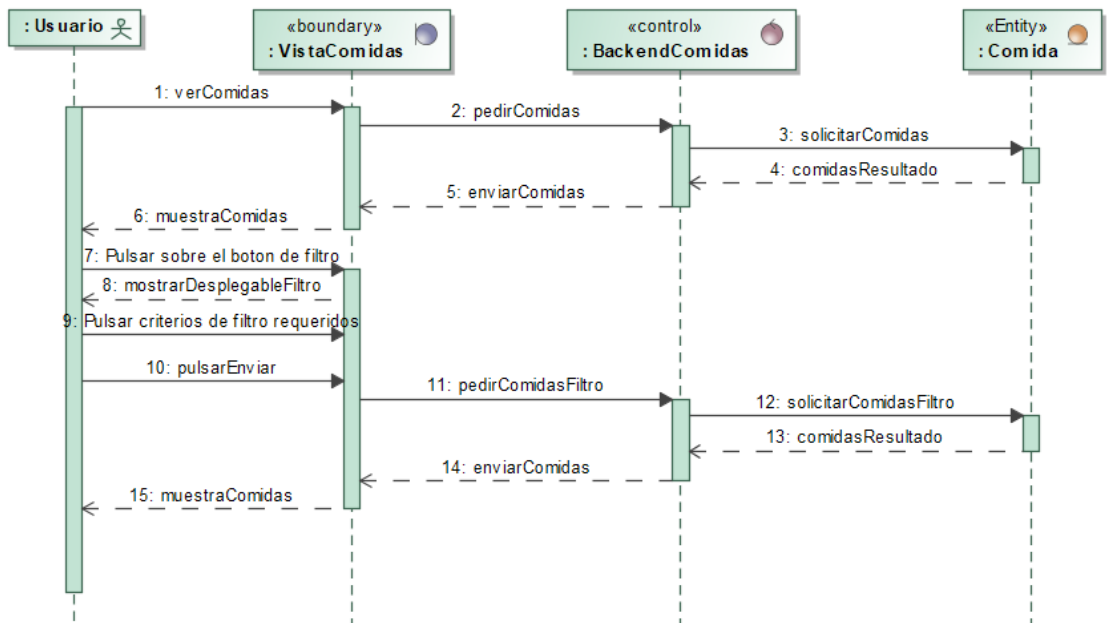


Figura 6 Diagrama de secuencia de Buscar comidas



**Figura 7** Diagrama de secuencia Filtrar comidas

# 4

## Análisis y diseño del almacenamiento

### 4.1 Análisis

En este capítulo se analizará qué tipo de información se desea almacenar en la base de datos de la aplicación y qué tipo de esquema será más adecuado usar, es decir, uno de tipo de relacional o uno de tipo de no relacional. Para ello se dispondrán las diferentes ventajas y desventajas que aporta cada uno, y al final se optará por uno de los dos esquemas teniendo en cuenta lo desarrollado.

#### 4.1.1 Base de datos relacional

Las bases de datos con estructura relacional (MongoDB, 2022b) tienen un sistema de almacenamiento basado en tablas las cuales suelen tener información compartidas entre ellas. Para interactuar con estas lo más común es hacer uso de SQL, lenguaje de consultas estructurados, el cual está estandarizado en el desarrollo del software permitiendo un fácil acceso a tutoriales y documentación sobre este. A continuación se listan las posibles ventajas y desventajas que ofrece un sistema de este tipo:

- **Cumplimiento de ACID:**  
Se asegura el cumplimiento del estándar ACID (Atomicidad, coherencia, aislamiento y durabilidad) el cual es muy importante para asegurar la fiabilidad de las transacciones dentro de las bases datos. El principio general es que, si un cambio falla, toda la transacción fallará por lo que el estado de la base de datos no será modificado y se mantendrá igual que antes de intentar la transacción.
- **Precisión de los datos:**

El uso de claves primarias y foráneas permite asegurar que no hay información duplicada. Ayudando a asegurar la exactitud de los datos porque nunca habrá información repetida.

- **Normalización:**

El proceso de normalización consiste en garantizar que los datos se organicen de tal manera que se reduzcan o eliminen las anomalías de los datos. Esto, a su vez, reduce los costes de almacenamiento.

- **Simplicidad:**

Este tipo de sistemas existen desde hace tiempo que se ha desarrollado una gran variedad de herramientas y recursos para empezar a usar bases de datos relaciones e interactuar con ellas. La similitud de la sintaxis de SQL con el inglés ayuda que gente que no sea desarrolladoras pueda realizar informes y consultas a partir de los datos.

- **Escalabilidad:**

Los sistemas de gestión de bases de datos relacionales están pensados históricamente para ser ejecutados en una sola máquina, por lo que, si resulta que los requisitos de la máquina no son suficientes, por problemas de espacio o por un aumento de la frecuencia con la que se accede a esta, se deberá hacer mejoras en el hardware de la máquina lo que se suele conocer como escalado vertical.

Esto puede resultar increíblemente caro y además hay un límite de cuanto se puede mejorar el hardware, ya que entre otras razones los costes terminarán superando a los beneficios. Además de que es muy probable que llegue un punto en el que no se pueda obtener un hardware capaz de alojar la base de datos. La única solución sería comprar una máquina que soporte un mejor hardware, pero nada de eso es barato.

- **Flexibilidad:**

En las bases de datos relacionales los esquemas son rígidos. Se definen las columnas y los tipos de datos que almacenan esas columnas, incluyendo cualquier restricción como por ejemplo la longitud o el formato. Ejemplos comunes de restricciones podrían ser la longitud de un número de teléfono o el mínimo y máximo para una columna de nombre.

Aunque esto significa que se pueden interpretar los datos y las relaciones entre tabla más fácilmente, implica que realizar cambios sobre las estructuras de los datos es una tarea muy compleja. Esto obliga al desarrollador a elegir como serán los datos desde el principio, lo cual no es siempre posible. Si se deciden hacer cambios más tarde, te ves obligado a cambiar todos los datos, lo que conlleva que la base de datos esté desconectada de forma temporal.

- **Rendimiento:**

El rendimiento de la base de datos está estrechamente ligado a la complejidad de sus tablas: el número de ellas al igual que la cantidad de datos que cada una de ellas alberga.

#### 4.1.2 Base de datos no relacional

Las bases de datos con estructura no relacional (MongoDB, 2022b, 2022c) o también conocidas como bases de datos NoSQL, son aquellas que no hacen uso de los conceptos de tabla, campos y columnas de las bases de datos relacionales.

Los sistemas no relacionales han sido diseñados teniendo en cuenta la nube por lo que son perfectos para ser escalados de forma horizontal. Debido a lo amplio que es concepto podemos encontrar diferentes tipos dentro de las bases de datos no relacionales:

- **Base de datos basada en documentos:**

Este tipo de bases de datos almacenan la información en documentos con un formato de estructuras similar a los archivos json que admiten diversos tipos de datos. Estos tipos incluyen cadenas, números como enteros, reales, *long*, fechas, objetos, matrices e incluso documentos anidados. Los datos se almacenan en pares, similares a los pares clave/valor.

Debido a que los documentos son parecidos al formato json, son mucho más fáciles de leer y entender para el usuario. Los datos se encuentran organizados y son fáciles de observar. Muchas veces no existe la necesidad de referenciar múltiples colecciones o documentos para ver los datos. Los documentos se corresponden perfectamente con los objetos en la metodología de programación orientada a objetos, facilitando de esta manera mucho el trabajo.

- **Base de datos de clave-valor:**

Este tipo de bases de datos son de los más simples que existen ya que solo consta de dos partes: clave y valor.

La clave se usa para recuperar la información de la base de datos. La simplicidad de este tipo de base de datos también es una ventaja debido a que todo está almacenado en una clave única y en un valor que son o los datos o la ubicación de estos, la lectura y escritura es siempre mucho más rápida.

Sin embargo, esta simplicidad restringe los tipos de casos de usos para los que se puede usar. Requisitos más complejos de datos no pueden ser soportados por estos tipos.

- **Base de datos de grafos:**

Las bases de datos de grafos son las más especializadas de las no relacionales. Utilizan una estructura de elementos llamados nodos que almacenan los datos, y las aristas entre ellos contienen atributos sobre las relaciones.

Las relaciones están definidas en las aristas, lo que hace que las búsquedas relacionadas con estas relaciones sean naturalmente rápidas. Además, son flexibles porque los nuevos nodos y aristas pueden ser añadidos fácilmente. Tampoco tienen que tener un esquema definido como una base de datos relacional tradicional.

Sin embargo, no son todas buenas para consultar toda la base de datos, donde las relaciones no están del todo o para nada definidas. Tampoco constan de un lenguaje estándar para la realización de consultas, lo que implica que cambiar de una base de datos basada en grafos a otra conlleve una curva de aprendizaje adicional.

- **Base de datos de columnas anchas:**

Por último, pero no menos importante, se encuentran las bases de datos de columnas anchas, estas al igual que las bases de datos relacionales

almacenan los datos en tablas, columnas y filas. Sin embargo, los nombres y el formato de las columnas no tienen por qué coincidir en cada fila. Las columnas pueden incluso almacenarse en varios servidores. Se consideran almacenamiento bidimensional de clave-valor porque utilizan mapeo multidimensional para referenciar los datos por fila y columna.

Al igual que las bases de datos clave-valor de dos columnas, las de columnas anchas tienen la ventaja de ser flexibles, por lo que las consultas son rápidas. Gracias a esta flexibilidad, son buenas para manejar "big data" y datos no estructurados.

Sin embargo, en comparación con las bases de datos relacionales, las de columnas anchas son mucho más lentas a la hora de gestionar las transacciones. Las columnas agrupan juntos atributos similares en vez de usar diferentes filas y separarlos en archivos separados, por lo que las transacciones tienen que ser llevadas a través de varios archivos.

#### **4.1.3 Conclusión**

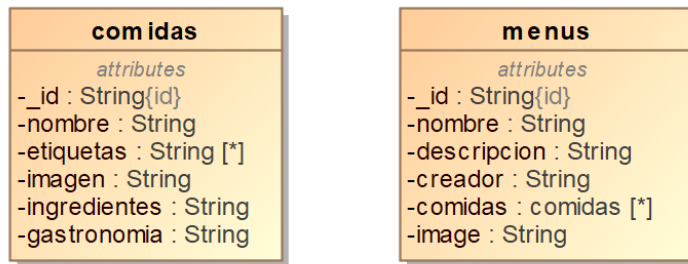
Tras lo enunciado anteriormente y lo visto en (MongoDB, 2022c, 2022b) se ha desarrollado la siguiente conclusión sobre cuál será el método de almacenamiento utilizado para este proyecto.

A pesar de que las bases de datos relacionales cuenten con muchas ventajas que podrían resultar de utilidad para este proyecto, la opción por la que se va a decantar finalmente es una base de datos no relacional basada en documentos debido a las siguientes razones:

- Se busca que los datos almacenados tengan un carácter flexible para que estos se puedan cambiar para cumplir alguna de las funcionalidades que se han visto que podrían ser de utilidad previamente en el punto 2, como podría ser la actualización de la API para que soportara diversos idiomas o el que las comidas pudieran contar con tutoriales de Internet.
- Disponibilidad para un aumento en el almacenamiento, aunque en esta iteración se contará con una muestra no muy elevada de comidas o menús desarrollados por los usuarios, se espera que en futuras iteraciones la base de datos contara con un número muy elevado de comidas y menús desarrollados por los usuarios.
- Una mejor gestión del tráfico y la necesidad de tiempo de inactividad cero, este tipo de base de datos permiten realizar cambios a tiempo real sin la necesidad de tener que estar en tiempo de espera mientras estas se realizan.
- Debido a su similitud con el formato de archivos json, se facilita la tarea de mostrar o escribir los datos de la API para los usuarios.
- Facilidad para el cambio de tipo de estructuración de los datos en mitad del desarrollo en caso que se vea que el enfoque por el que se ha optado inicialmente necesita algún tipo de refinamiento para adaptarse a algunos de los requisitos que se pretenden completar en esta iteración, adaptándose a la metodología usada en el desarrollo ágil de software.

## **4.2 Diseño**

A continuación se ha realizado un diagrama de la base de datos de acuerdo a los requisitos especificados previamente y a las características del sistema en el que se pretende desarrollar este:



**Figura 8** Diagrama que seguirá la base de datos

Como se puede ver en la Figura 8, el sistema constaría de dos tipos de documentos:

- **Menu:**

- **\_ID**  
Valor hexadecimal para identificar y diferenciar cada elemento entre sí.
- **nombre**  
Cadena para que los usuarios puedan identificar estos.
- **descripción**  
Breve cadena de texto opcional para que los usuarios puedan apuntar algún detalle que consideren importante.
- **creador**  
Correo del creador del menú, debido a que la aplicación usará el sistema de autenticación de Google, este se puede usar a modo de "clave foránea" con los usuarios que utilicen la aplicación.
- **comidas**  
Array de array de comidas de tamaño adaptable, este atributo representará el menú semanal por lo que será necesario que este se encuentre ordenado. Debido al tipo de base de datos que se está utilizando esto podría conllevar mucha información repetida, por lo que solo se almacenarían únicamente los campos que fueran necesarios para el menú.

- **Comida:**

- **\_ID**  
Código alfanumérico para identificar y diferenciar cada elemento entre sí.
- **nombre**  
Nombre que se utilizará para identificar a la comida.
- **gastronomía**  
Cadena que representará a que gastronomía pertenece el plato.
- **etiquetas**  
Lista de los alérgenos o consideraciones importantes que tengan los alimentos.
- **imagen**  
URL a la imagen que tiene asignada cada comida.
- **ingredientes**  
Cadena que contiene todos los ingredientes que contiene la comida.

# 5

## Tecnologías y Herramientas

### 5.1 Introducción

Una vez haber realizado el estudio y análisis inicial de los requerimientos del proyecto a realizar, se van a listar las tecnologías y herramientas que se van a utilizar en este junto con una breve descripción de ellas y de cuál va a ser su función concreta dentro de este.

### 5.2 Figma

Figma (Figma, 2022a) es una aplicación de desarrollo de prototipos y de edición de gráficos vectorizados, disponible en versión en web y versión de escritorio. Actualmente es una de las opciones más utilizadas en el campo del diseño y desarrollo de interfaces de usuario y de experiencia de usuario, debido a su versatilidad, todas sus funcionalidades para la colaboración a tiempo real y al continuo aumento de herramientas como plugin o plantillas desarrollados por la propia comunidad de esta. En este proyecto se utilizará esta herramienta para el desarrollo de ciertos elementos o diseños gráficos y para la elaboración de los prototipos de baja y alta fidelidad.

### 5.3 Python

Python (Python, 2022) es un lenguaje de programación de alto nivel habiéndose convertido en uno de los más utilizados en la actualidad. Dispone de herramientas y librerías para diferentes campos de desarrollo, entre ellos se encuentra el desarrollo web que es para lo que será utilizado en este proyecto.

## 5.4 Flask

Flask (Flask, 2022) es un micro framework ligero de Python para el desarrollo de aplicaciones web. Este está diseñado para que empezar a desarrollar sea algo rápido y fácil, con la capacidad de poder escalar a aplicaciones de carácter más complejo. Actualmente es uno de los frameworks de desarrollo web más populares dentro de la comunidad de Python.

Flask ofrece algunas sugerencias pero no impone ningún tipo de dependencia o disposición del proyecto, dejando libertad al desarrollador para que elija las herramientas y bibliotecas que desea utilizar. Además, existen muchas extensiones desarrolladas por la comunidad que hacen que añadir funcionalidades sea mucho más simple.

En este proyecto Flask será usado para el desarrollo del backend de la aplicación.

## 5.5 MongoDB

MongoDB (MongoDB, 2022a) es una base de datos no relacional basada en documentos muy utilizada en la actualidad que almacena la información en documentos BSON los cuales tienen mucha similitud con los archivos JSON por lo que cualquier tipo de conversión es muy simple.

## 5.6 HTML5

HTML (MDN contributors, 2022a) es un lenguaje de marcado de texto con etiquetas mediante el cual se estructuran y definen los componentes básicos de las aplicaciones web además de enlazar estas entre sí. Se suele acompañar de otras tecnologías para obtener tanto un aspecto como una funcionalidad de mayor complejidad.

En este proyecto formará parte clave del frontend de la aplicación web.

## 5.7 CSS

CSS (MDN contributors, 2021) es un lenguaje de estilos que se utiliza para definir cómo se van a renderizar los elementos a mostrar en pantalla de sitios web o aplicaciones que funcionen con páginas HTML o XML. Al igual que HTML, este formará parte fundamental de la página web junto con las herramientas que se van a mencionar a continuación.

## 5.8 JavaScript

JavaScript (MDN contributors, 2022b) es un lenguaje de programación muy popular en la actualidad y muy estandarizado en el campo de las aplicaciones web que permite al desarrollador programar aspectos de la interfaz del usuario respecto a diferentes interacciones del usuario junto con cómo reacciona la aplicación respecto a diferentes eventos lanzados por el usuario.

## 5.9 React

React (React, 2022) es un framework de JavaScript financiado por Meta para el desarrollo frontend de aplicaciones web interactivas. Ofrece un amplio abanico de funcionalidades para simplificar y acelerar la creación de interfaces de usuario y el desarrollo de SPA. Además cuenta con soporte tanto para el desarrollo con JavaScript como TypeScript, aunque en este proyecto solo se focalizará en el uso de JavaScript.

## 5.10 Next.js

Next.js(Vercel, 2022) es uno de los framework más potentes y populares de React debido a la gran cantidad de mejoras que ofrece sobre este. Entre sus funcionalidades más importante destaca que cuenta con *server side-rendering* de base (lo cual conlleva muchas ventajas en el rendimiento de la aplicación), no requiera configuración externa y sistema de rutas de bases.

## 5.11 GitHub

GitHub (GitHub, 2022) es un sitio web que mediante el uso interno de Git (Git, 2022) permite la creación, compartición y seguimiento del control de versiones de proyectos softwares de forma individual o colaborativa. Además debido a lo mencionado anteriormente, cuenta con una de las mayores comunidades de desarrolladores de software. En este proyecto se utilizará GitHub tanto para tener una copia de seguridad de la aplicación como para mantener un control de versiones sobre esta.

## 5.12 Visual Studio Code

Visual Studio Code (Visual Studio Code, 2022) es un poderoso y ligero editor de código que se ejecuta en el escritorio y se encuentra disponible para todos los sistemas operativos. Cuenta con soporte para JavaScript, TypeScript y Node.js además de contar un amplio de catálogo de extensiones para otros lenguajes e incluso para añadir nuevas herramientas al editor.

## 5.13 MagicDraw

MagicDraw (NoMagic, 2022) es una herramienta de escritorio CASE (*Computer Aided Software Engineering*), es decir una aplicación cuyo fin es mejorar el desarrollo de software reduciendo sus costes en tiempo y dinero, creada por NoMagic. Se centra principalmente en facilitar la elaboración de diagramas y modelos técnicos siguiendo el estándar UML.

## 5.14 Insomnia

Insomnia(Kong, 2022) es una herramienta utilizada para el control de calidad y diseño de API REST con el uso de interfaz simplificada e intuitiva que permite tanto probar consultas separadas, como crear colecciones de consultas, o el desarrollo de archivos de documentación.

# 6

## Diseño de la interfaz y experiencia de usuario

### **6.1 Introducción**

En este capítulo se describirá todo el proceso llevado a cabo para la elaboración del diseño del prototipo final de la interfaz de la aplicación web del proyecto. Primero se seleccionará el tipo de usuario al que se desea orientar la aplicación. Luego a continuación se elaborará un prototipo centrado en las funcionalidades y la navegabilidad de la aplicación y no en el diseño. Una vez hecho esto se llevará a cabo una lluvia de ideas para el aspecto de la aplicación. Por último se realizará el prototipado final de la aplicación en el que ya se mostrará el aspecto que se desea que tenga finalmente la aplicación.

### **6.2 Persona**

A continuación, como se ha mencionado anteriormente, se va a desarrollar una persona del tipo de usuario al que se va a dirigir la aplicación. Como se explica en (Faller, 2019) esto ayuda a no desarrollar un producto genérico y a resolver una de las grandes preguntas que se tienen que plantear los diseñadores: "¿Para quién se está diseñando?" De esta manera se comprenden las expectativas, las preocupaciones y las motivaciones de los usuarios objetivos, además de que se evita que el diseñador piense en sí mismo a la hora de realizar el diseño, consiguiendo así construir un producto que satisfaga las necesidades del cliente y tenga éxito.

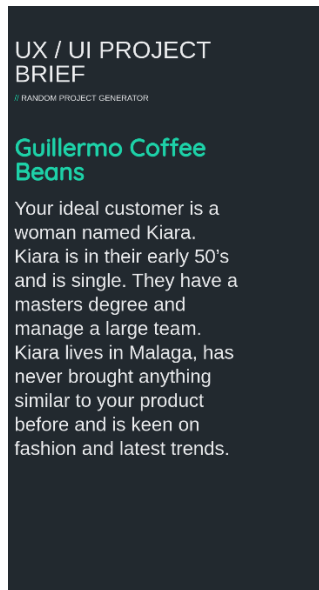


Figura 9 Persona ficticia de (Scott & Bring Your Own Laptop, 2020)

Debido a la escala de este proyecto, en vez de realizar una persona tan extensa como la mencionada en (Faller, 2019), se desarrollará una más simple parecida a la que se plantea en (Scott & Bring Your Own Laptop, 2020), la página elaborada por el diseñador Daniel Walter Scott en la figura 9.

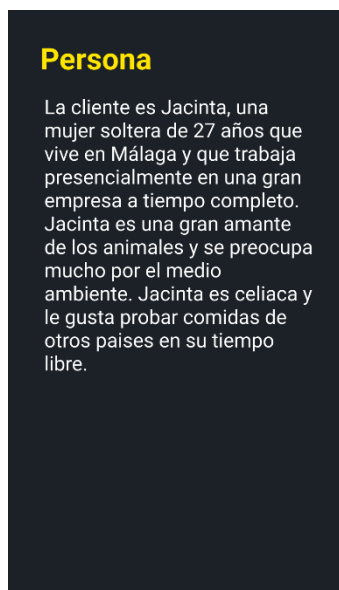


Figura 10 Persona a la que se va a dirigir el diseño

Como se muestra la figura 10, para la persona desarrollada se ha buscado pensar en alguien joven que no cuente con mucho tiempo para gestionar sus comidas, y que tenga intolerancias alimenticias. Por último, se han añadido algunas características que pueden afectar a su elección de comidas y que, por tanto, se deberán tener en cuenta para el desarrollo.

### 6.3 Prototipo de baja fidelidad

En esta sección se va a desarrollar el prototipo de baja fidelidad. Consistirá en la elaboración de las pantallas de una forma muy simplificada sin ningún tipo de estilos

solo centrándose en qué elementos tendrá cada pantalla y las funcionalidades y navegabilidad que tendrán estas.

Lo primero que se ha hecho ha sido optar por una metodología de *mobile first*. Esto como se ve en (Morales, 2021) tiene diversas ventajas a tener en cuenta:

- Al diseñar inicialmente para los dispositivos más pequeños, ayuda al diseñador a detectar los componentes más importantes de la experiencia del usuario con el producto y centrarse en ellos en vez de en elementos que pueden resultar secundarios. Una vez terminado el diseño en los dispositivos más pequeños ya se pueden añadir elementos para complementar el producto en dispositivos más superiores.
- En la actualidad, según (Handley, 2019) se estima que dos mil millones de personas acceden a Internet únicamente a través de sus dispositivos móviles y se estima que para 2025, el porcentaje de usuarios que accedan a Internet únicamente usando dispositivos móviles ascienda a 72,5%, lo cual equivale 3700 millones de personas. Tener una buena experiencia de usuario en móvil, por tanto, es algo vital para cualquier aplicación web en la actualidad. Además de es más probable que los usuarios compren y vuelvan a negocios cuya página sea amigable para móviles.
- Otro beneficio importante a la hora de hacer un producto con la metodología *mobile first* es que, como dijo Google (Google, 2022b), realizaron ajustes en su algoritmo de búsqueda de forma que se favorece a los sitios web que cuenten con una versión móvil para el contenido.
- Otro punto a favor de este enfoque es que desde principios de 2012 según (Xia, 2017) la venta de smartphones ha superado la venta de ordenadores personales, por lo que los consumidores están comprando más este tipo de dispositivos que ordenadores y los usuarios están pasando más tiempo en Internet aumentando la demanda de experiencias de usuario móvil agilizadas.
- Por último, otra razón para hacer uso de un diseño basado en *mobile first* reside en el hecho de que las compañías en la actualidad gastan más dinero en publicitarse en dispositivos móvil de lo que lo hacen en la televisión. Al igual que como se predecía en (Handley, 2019) y como se dice en (Dogtiev, 2022; Statista, 2022), en la actualidad se ha producido un gran aumento de la cantidad de dinero invertido en anuncios a través de aplicaciones y dispositivos móviles remplazando en parte a la popularidad que tenían antes los anuncios televisivos.

Una vez seleccionado la metodología a seguir a la hora de realizar el diseño, se seleccionarán los dispositivos que se usarán de base para los prototipos de la versión móvil y de la versión de ordenador. Para ello se ha realizado una búsqueda para encontrar qué resolución de pantalla es más popular y así contar con un tamaño genérico que pueda servir para la gran mayoría de dispositivos. Tras esta investigación se ha optado por usar para smartphones las dimensiones de un iPhone8 (375x667) ya que, según (Worship, 2021), esta es la más popular. Aunque dependiendo de la fuente consultada esto puede cambiar, no debería ser problemático ya que solo se busca uno de los muchos formatos genéricos que existen. En el caso de ordenador se ha optado por el formato Desktop (1440x1024) que ofrece Figma ya que los otros formatos disponibles son más específicos de Apple. Estas elecciones no conllevan un gran riesgo ya que en

futuras iteraciones, si se mantuviera conversaciones con un cliente real y este deseara que los prototipos estuvieran pensados en las dimensiones de los dispositivos que él usa, Figma ofrece herramientas para cambiar las dimensiones de las pantallas.

A continuación, antes de empezar a diseñar, es importante definir el flujo de tareas del usuario que se van a representar. Por ello se ha realizado el siguiente diagrama en la Figura 11, que muestra de forma simplificada como inicialmente pretende ser la navegación entre pantallas que deberá realizar el usuario para la creación de un menú y posteriormente añadir comidas a él.

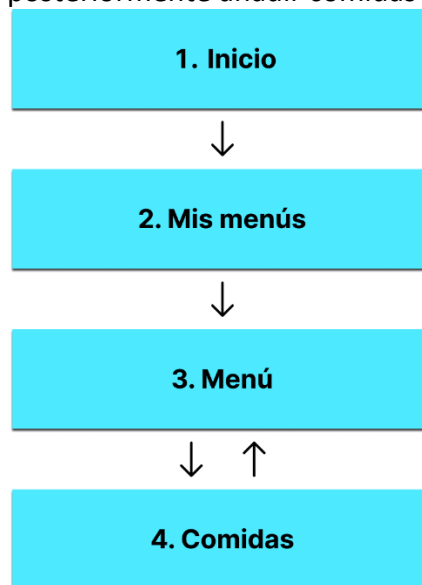


Figura 11 Flujo de tareas de creación de menú y añadir comida a menú

Una vez tomadas estas decisiones, se ha procedido a diseñar el prototipo de baja fidelidad para dispositivos móviles donde se han tomado varias decisiones para una mejor experiencia de usuario las cuales se desarrollarán a continuación:

- **Posición del botón con la acción primaria:**  
Como se menciona en (Babich, 2016; Syzonenko, 2019) no se existe una opción que sea claramente mejor ya que ambas cuentan con argumentos a favor y en contra. En este caso, como se ve en la Figura 12, se va a optar por poner siempre la opción afirmativa la primera. En el caso de disposición de botones en vertical esto significa que la opción primaria se encontrará arriba y en el caso de disposición en horizontal significa que será la que se encuentre más a la izquierda. Esta elección concuerda con el flujo normal de lectura; y con la estructura de las oraciones en la cultura occidental donde, cuando se realiza una pregunta la opción positiva va primero y la negativa después, por ejemplo "¿Quieres comer aquí, si o no? Además, suponiendo que los usuarios aceptarán con mucha más frecuencia de lo que cancelarán, esta opción es mejor para los usuarios que navegan por los formularios con el tabulador pues les supone menos pulsaciones llegar a esta opción.
- **Disposición de los menús individualmente:**  
Debido al tamaño de la pantalla del teléfono móvil se ha decidido que lo más cómodo es que se muestre en la pantalla individualmente solo un día de la semana del menú y que haya una opción para navegar por día,

en vez de mostrar toda la semana directamente porque se vería la información demasiado pequeña y compactada para el usuario.

- **Método de añadir comida a un menú:**  
Debido que la persona es alguien joven que no cuenta con mucho tiempo para cocinar se ha supuesto a la hora de diseñar que es posible que el usuario desee añadir varias veces la misma comida o comidas parecidas a su menú. Por ello se ha pensado que la forma más cómoda de hacer esto es directamente desde el buscador de comidas en vez de desde la propia pantalla de menú. Sin embargo se ha pensado que para mejorar la navegabilidad se añadirá un botón que redirija al buscador de comidas como se ve en la Figura 12.
- **Disposición de elementos en lista:**  
Para una mayor claridad visual se ha decidido optar por mostrar los elementos en una sola columna en una disposición horizontal para poder mostrar toda la información que contienen estos pero de una forma clara y concisa.
- **Aspecto del primer prototipo:**  
Debido a que esta etapa del diseño no se centra en el aspecto, se ha optado por representar las imágenes en forma de polígonos y circunferencias con cruces, y se ha optado por usar de forma arbitraria un color para este prototipo, pero estos no son representativos del prototipo final.
- **Menú de navegación:**  
Se ha decidido añadir un menú de navegación a pesar de contar con pocas pantallas a las que viajar para mejorar la navegabilidad del usuario y permitir a este llegar al mismo lugar de varias maneras diferentes. Debido a que actualmente se está solo considerando solo el aspecto de móvil, se ha decidido no añadir botones de volver atrás porque se cuenta con el botón móvil de marcha atrás.
- **Disponibilidad de la aplicación si no tienes iniciada sesión:**  
En principio se está diseñando la aplicación pensando en que para acceder a todas las funcionalidades necesites haber iniciado sesión una cuenta de Google, pero se valorará como funcionalidad extra poder acceder al catálogo de comidas sin necesidad de haber iniciado sesión.

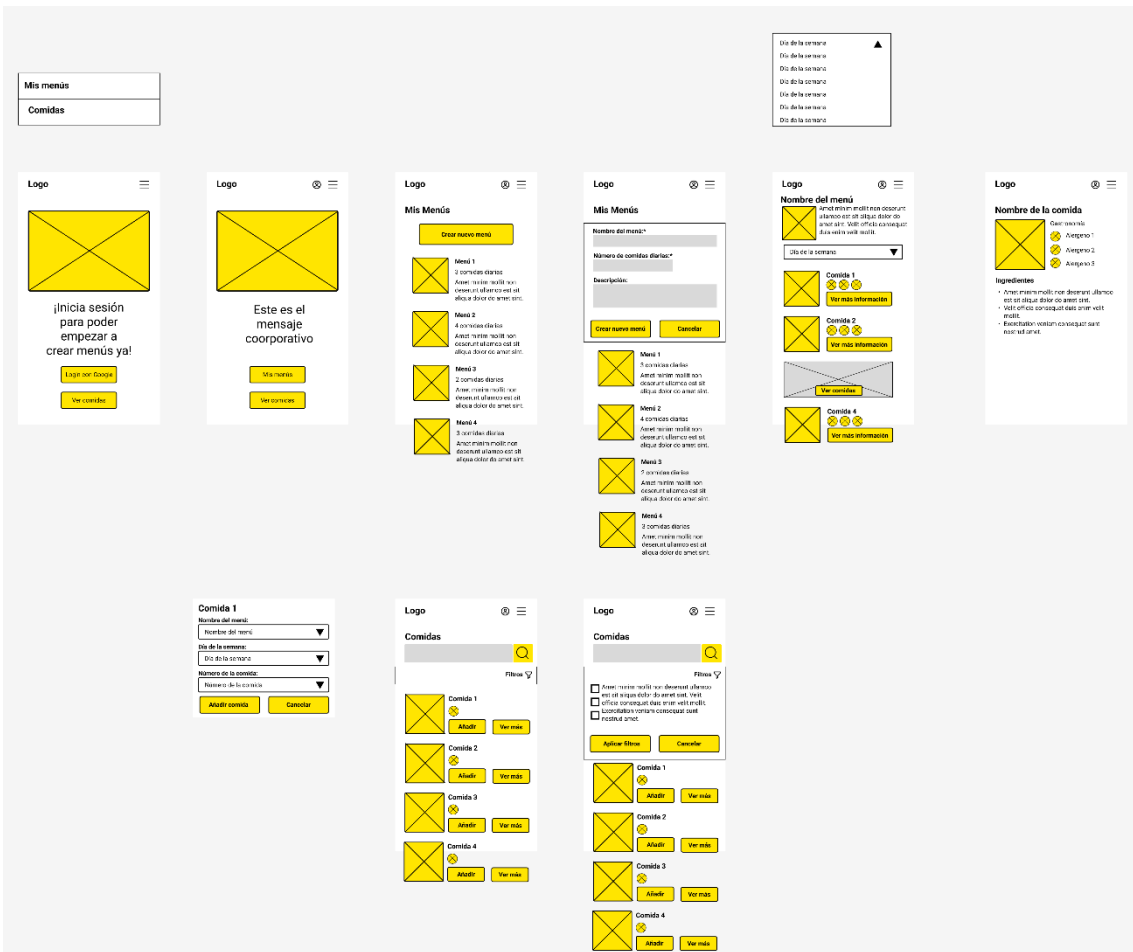


Figura 12 Prototipo de baja fidelidad versión móvil

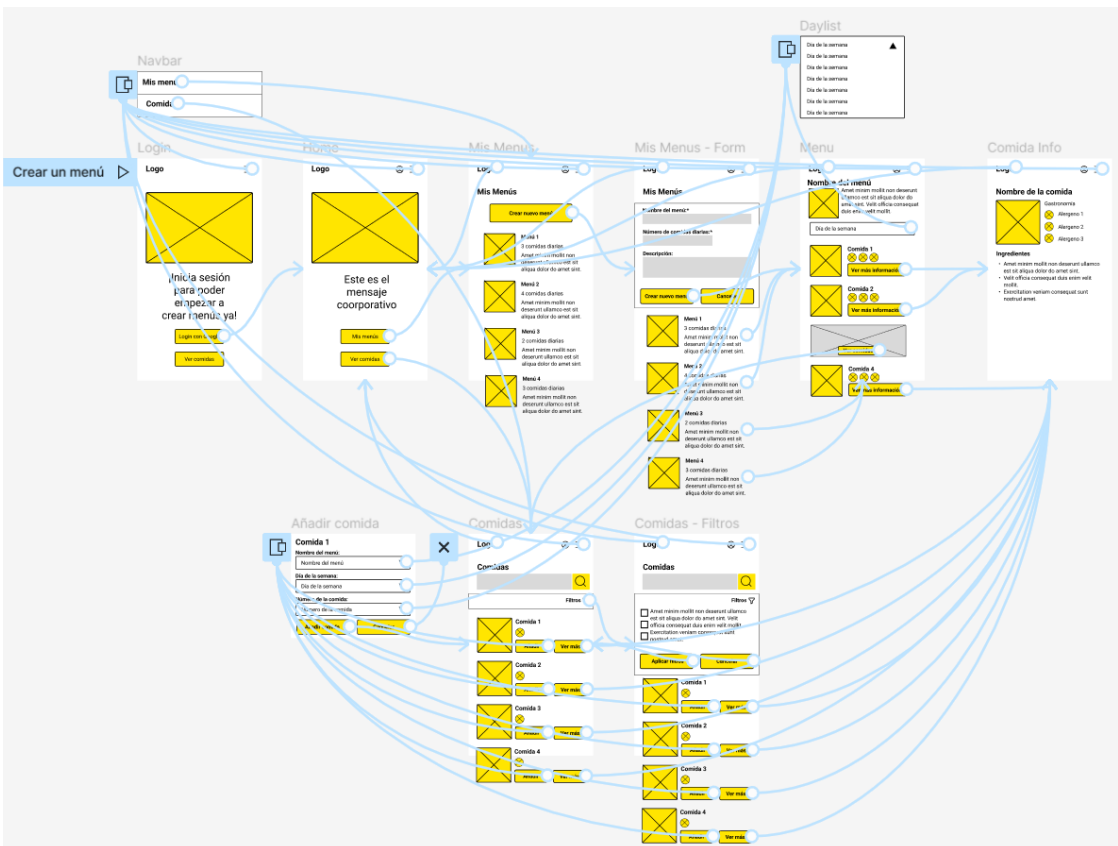


Figura 13 Navegabilidad del prototipo de baja fidelidad versión móvil

Como se ve en las Figura 13 y Figura 15, se ha representado también la navegación que tendría la aplicación para poder hacer una simulación correcta de cómo sería el uso de este prototipo. Para hacer una mejor comprobación del prototipo se ha hecho uso de la herramienta Mirror (Figma, 2022b) de Figma la cual permite probar el prototipo en un dispositivo móvil.

A continuación, como se ve en la Figura 14, se ha adaptado el diseño realizado para móvil a la versión de ordenador. Para ello se ha tenido en cuenta que había elementos que en una pantalla más grande que la de un smartphone dejarían demasiado espacio en blanco, como son la lista para seleccionar el día de la semana en la pantalla de menú o principalmente también las "tarjetas" utilizadas para representar alimentos y menús. Para esto último se ha tenido en cuenta lo mencionado en (Material Design, 2022). También se ha decidido probar por cambiar el menú desplegable que había en la barra de navegación de móvil por simplemente el texto de la ventana a la que se desea navegar, aunque se considera que esta opción es solo aplicable en casos con pocas ventanas a las que navegar, por lo que en el caso de que la página escalara, probablemente habría que usar también un menú desplegable como el de móvil.



Figura 14 Prototipo de baja fidelidad versión ordenador

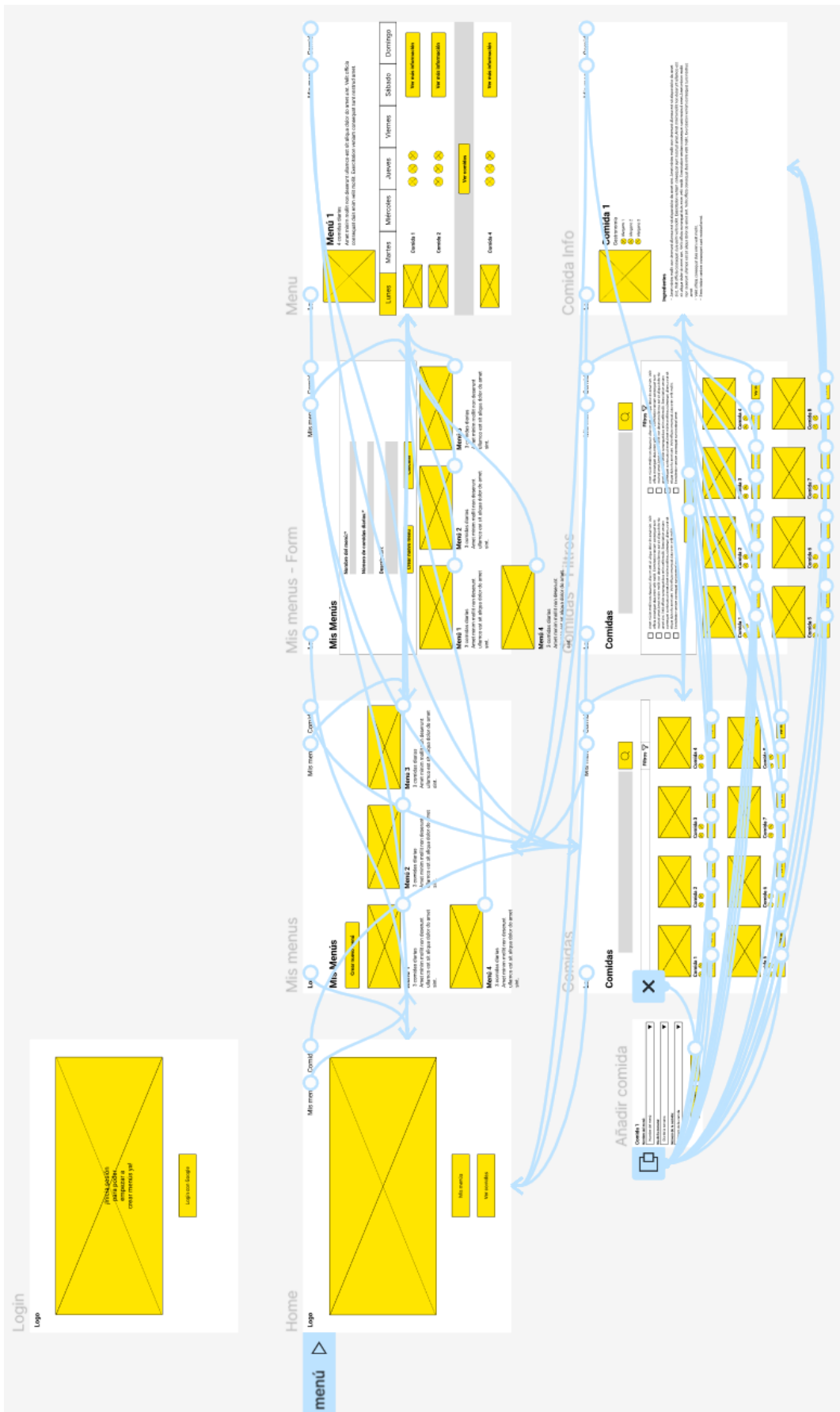


Figura 15 Navegabilidad del prototipo de baja fidelidad versión ordenador

## 6.4 Prototipo de alta fidelidad

Una vez finalizado el prototipado de baja fidelidad en el que se han definido la estructura, la usabilidad y el comportamiento de la página, se ha procedido a desarrollar el prototipo de alta fidelidad. Para esto primero se ha decidido cuál sería la paleta de colores que conformaría la aplicación. Para ello se ha seleccionado un color primario que sería el que le dará la personalidad a la aplicación y después uno secundario para completar a este junto a un color extra, por si se necesitara representar algún elemento de una forma especial o diferente. Para el color principal se ha optado por una tonalidad de verde ya que este se suele asociar con la naturaleza y la vida saludable lo cual compagina muy bien con la persona hacia la que estamos enfocando el producto. Después, para seleccionar los otros dos colores, se ha utilizado la opción de "Separación Complementaria" de (Adobe Color, 2022c), ya que el uso de colores complementarios, además de generar un acabado atractivo visualmente, también ayuda a que sea más fácil que la página sea apta para gente con daltonismo. Una vez seleccionados los colores, se comprobó que estos tenían un buen contraste para una correcta lectura con el uso de (Adobe Color, 2022a) y por último se aseguró que esta selección era apta para daltónicos usando (Adobe Color, 2022b). Una vez terminada la selección de colores se seleccionaron diferentes tonalidades de estos además de las fuentes procedentes de (Google, 2022a), ya que permite uso gratuito, para desarrollar la página de estilos de la aplicación, y se elaboró la página de estilos de la Figura 16.

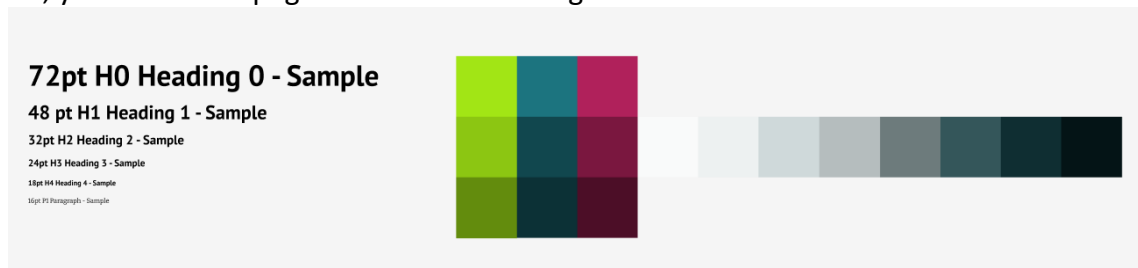


Figura 16 Página de estilos de la aplicación.

A continuación, antes de proceder a diseñar los prototipos, para una mayor facilidad a la hora de luego de diseñar y programar, se desarrolló la página de componentes de la aplicación de la Figura 17 haciendo uso de la herramienta de componentes que ofrece Figma.



Figura 17 Página de componentes que contiene la aplicación web

Una vez llevados a cabo estos pasos preliminares, se comenzó a realizar al igual que en el apartado anterior el prototipado de la versión móvil que se ve en la Figura 18. Primero, para ello, se tuvieron en cuenta los elementos explicados en (Material Design, 2022) para la elaboración de las tarjetas. También debido a que este trabajo final de grado no es un proyecto real no se cuenta con diseños personalizados para los alérgenos. Por esta razón en este prototipo se han usado *placeholders* ya que estos se obtendrán de una librería en el desarrollo. Para las imágenes se han usado Pexels (Pexels, 2022) y Unsplash (Unsplash, 2022) que ofrecen stock de fotos gratis. También al igual que para el prototipado de baja fidelidad se ha elaborado toda la navegabilidad de la aplicación, como se ve en la Figura 19.

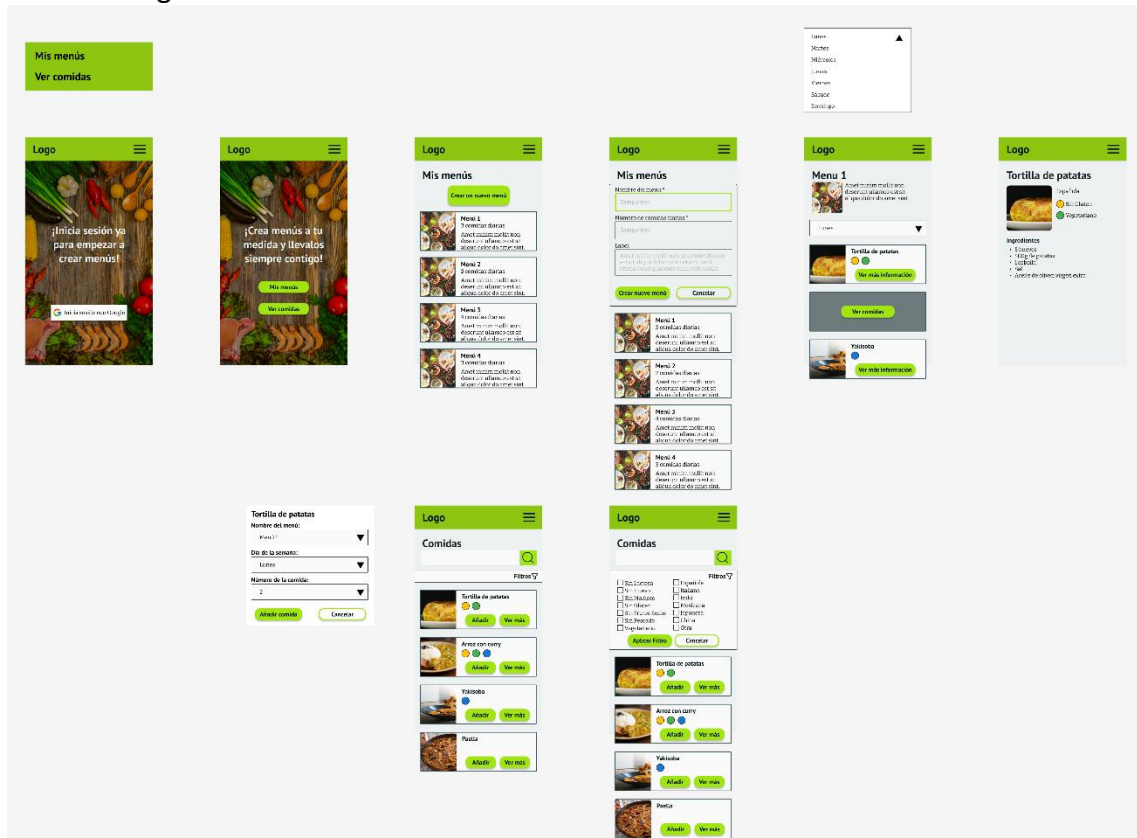


Figura 18 Prototipo de alta fidelidad versión móvil

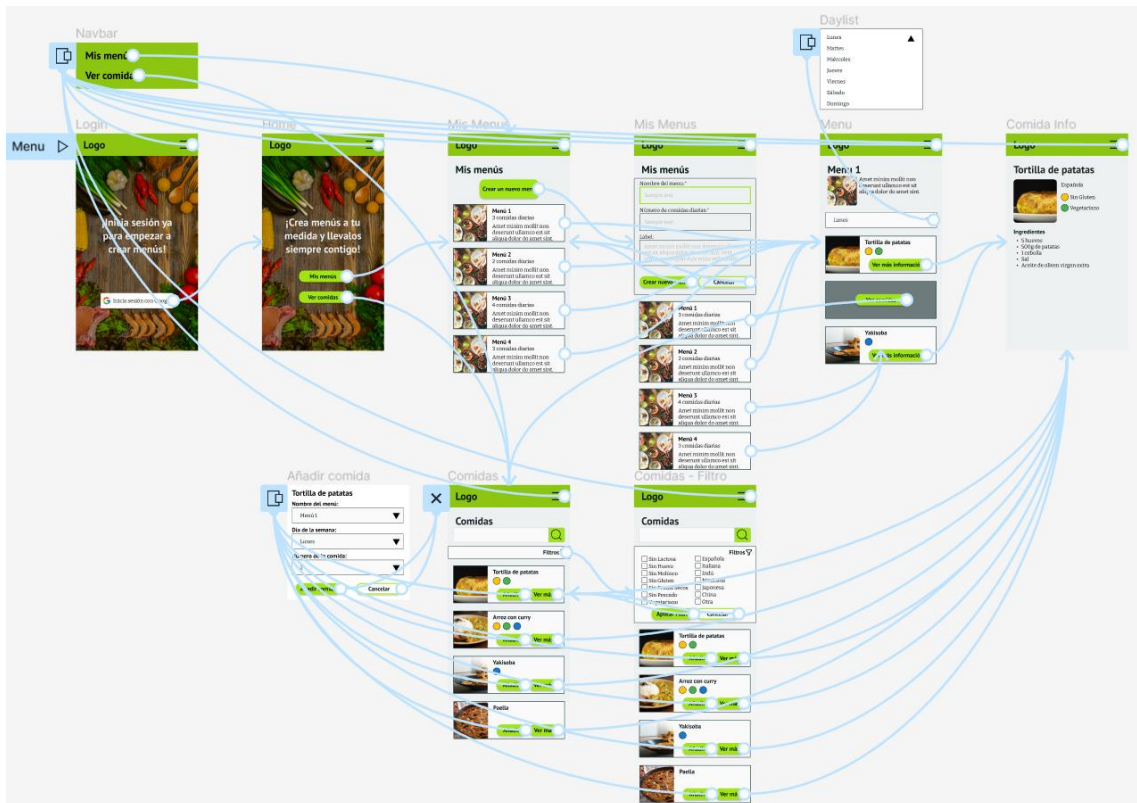


Figura 19 Navegabilidad del prototipo de alta fidelidad versión móvil

Una vez terminado el prototipado de la versión móvil se pasó a diseñar el de la versión de ordenador adaptando tanto el aspecto (Figura 20) como la navegabilidad (Figura 21) para que se adecuen a una pantalla de mayor tamaño, como se hizo en el apartador anterior. Sin embargo, se ha hecho un pequeño ajuste a la barra de navegación para que esta sea desplegable, como la de versión en móvil, en vez de la que se había propuesto inicialmente en el prototipo de baja fidelidad, debido a las consideraciones que se hicieron en el apartado anterior además de una mayor consistencia.

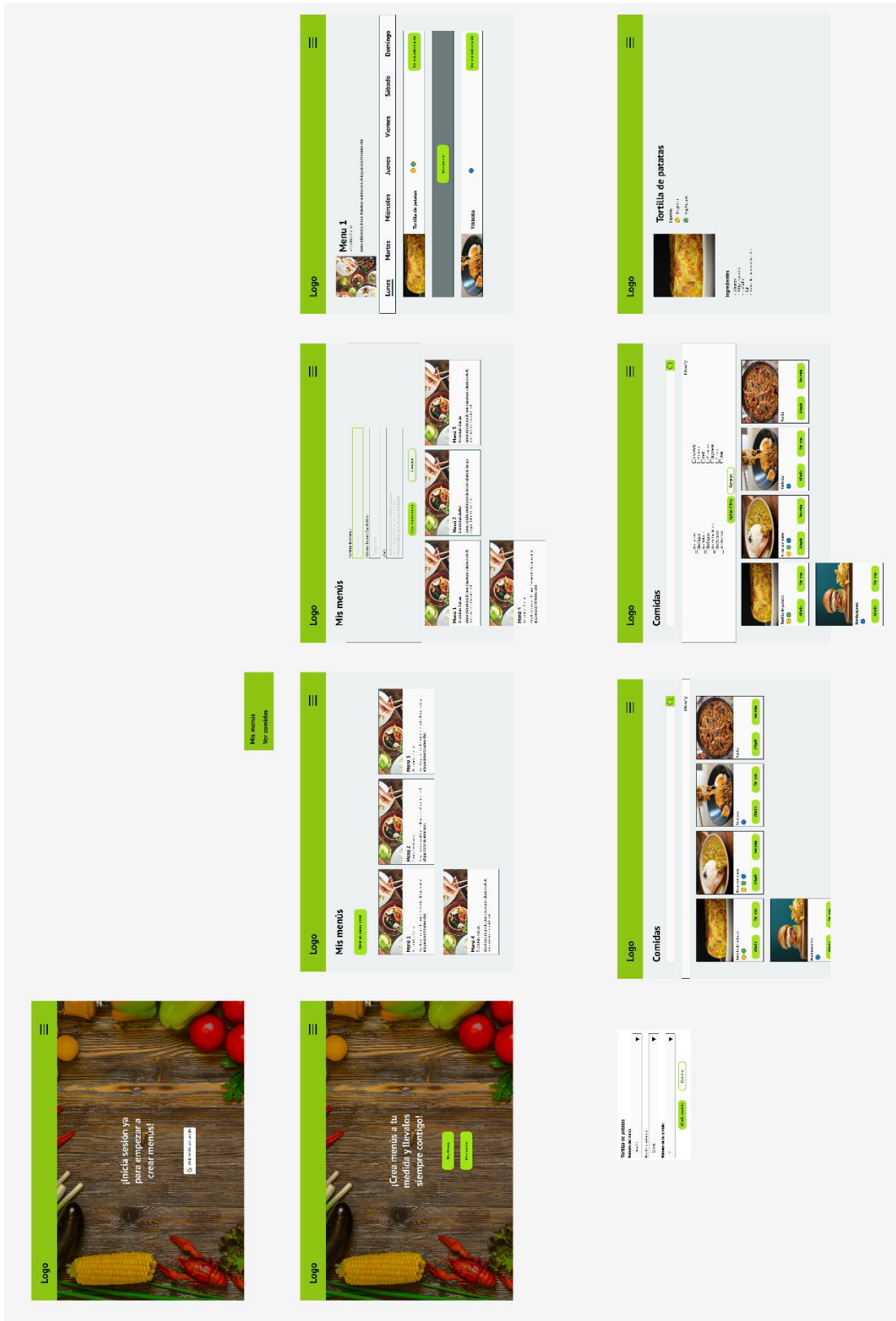


Figura 20 Prototipo de alta fidelidad versión ordenador



# 7

## Implementación

### 7.1 Introducción

En este capítulo se desarrollará todo el proceso llevado a cabo para la elaboración del funcionamiento del software de la aplicación web, primero se focalizará en preparar el conjunto de datos y la base de datos en la que se almacenarán los datos, luego se desarrollarán los diferentes métodos HTTP habituales en API REST para realizar las consultas necesarias junto con los diferentes filtrados o búsquedas específicas del catálogo de alimentos y por último se centralizará en el desarrollo del aspecto de la aplicación de acuerdo a los prototipos diseñados previamente.

### 7.2 Almacenamiento de datos

A continuación, debido a la naturaleza de MongoDB, la única configuración que se ha realizado de la base de datos local fue su creación y, dentro de esta, la creación de las dos colecciones de datos que serán utilizadas. No es necesario definir la estructura que tendrán estas debido a que los documentos, como se mencionó previamente, no son de un carácter fijo. Una vez realizado esto, se ha llevado a cabo la elaboración del script en formato json que almacenará toda la información relevante a las comidas. Para la creación de este se han tenido en cuenta varias consideraciones.

Para los alérgenos y gastronomías se ha optado por seleccionar arbitrariamente seis alérgenos y seis gastronomías que sean populares. Posteriormente, se ha decidido añadir una categoría extra, en el caso de los alérgenos para comidas que sean aptas para vegetarianos, y en el caso de gastronomías, una para identificar a los alimentos que no pertenezcan a ninguna. Luego a la hora de almacenar información inicialmente se barajó la idea de que ya que se iba a utilizar una librería de iconos para mostrar los alérgenos, almacenar también el nombre del componente referente a ese icono en la base de datos. Sin embargo, al final esa idea se descartó debido a que si en el futuro fuera necesario el uso de otra librería de componentes, el cambio de estos en la base de datos implicaría cambiar cada icono en cada elemento de la base de datos. Por ello, finalmente se ha optado por únicamente hacer la biyección en el frontend. Por último, para

almacenar todas las imágenes que utiliza la aplicación se ha decidido hacer uso de la plataforma Cloudinary (Cloudinary, 2022) ya que está pensada para desarrolladores y ofrece una API en caso de que en un futuro se quisiera añadir a la aplicación funcionalidades respecto al tráfico de imágenes.

Una vez terminada la redacción del archivo json, no es necesaria la elaboración de un fichero Python para la ingesta de los datos debido a que MongoDB cuenta con una funcionalidad para inyectar en una colección todo el contenido de un archivo en el formato json.

### 7.3 Desarrollo de la API de datos

Una vez terminada la construcción de los datos, se pasó al desarrollo de los métodos de petición HTTP. Para la elección de las diferentes rutas que tiene la API se decidió hacer uso de un método estandarizado en el desarrollo de estas que consiste en construir las rutas de la siguiente manera "/v{versión}/{colección}", por lo que las rutas de las dos colecciones en esta iteración son "v1/comidas" y "v1/menus". A continuación, para la elección de qué métodos http elaborar, se consideró que para la colección de comidas solo se desarrollarían los métodos de lectura y para la de menús métodos de lectura, creación y actualización. Para estos dos últimos debido a que son funcionalidades que afectan directamente a la "base de datos" se han protegido con las librerías de google-auth-http(Google Cloud Platform, 2021) y google-auth-oauth(Google Cloud Platform, 2022) de Python. Una vez realizados los métodos de petición básicos se añadieron los diferentes filtros y búsquedas especiales a estos.

Por último, una vez terminado todos los métodos de consulta se pasó a comprobar que su funcionamiento era correcto utilizando la herramienta Insomnia (Kong, 2022) como se ve en la Figura 22.

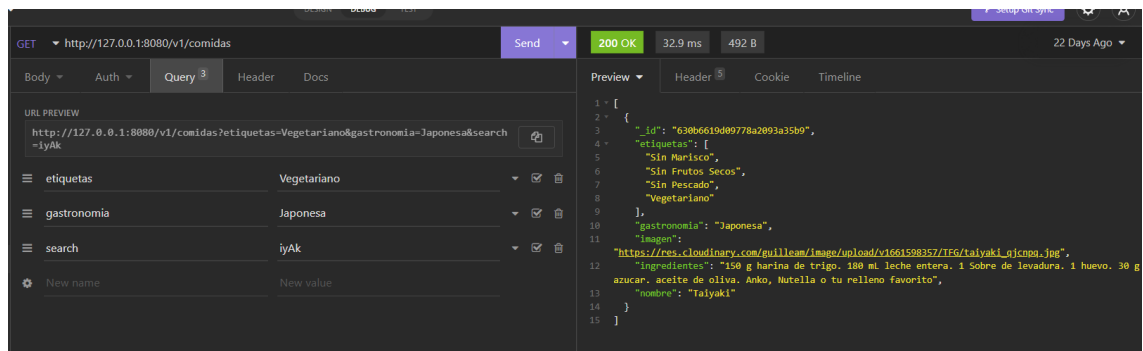


Figura 22 Comprobación del correcto funcionamiento de una consulta de filtro con Insomnia

### 7.4 Desarrollo del frontend de la aplicación

Por último, una vez elaborado todo el funcionamiento interno de la aplicación, se procedió a elaborar el frontend de esta. Debido al uso del framework "Next.js" lo primero en lo que se focalizó fue en la estructuración en carpetas del proyecto de la Figura 23. A continuación se describirá brevemente el cometido de cada carpeta:

- **components:** En esta carpeta se almacenan todos los componentes de la aplicación que serán utilizados por la aplicación, pero no son rutas de la misma:

- **AppLayout:** Este es un componente especial que ofrece el framework en el cual se encuentra código que queremos que se encuentre disponible en toda la aplicación. En este caso específico, aquí es donde se encuentra la barra de navegación y el espacio en blanco donde se depositará el contenido de cada página.
- **Modal:** Este componente representa la ventana de diálogo que aparecerá cuando se quiera añadir una comida. Debido a que esta es de carácter dinámico, dependiendo de los parámetros que reciba, se decidió que lo más correcto era en un componente externo que se pudiera reutilizar para cada comida.
- **Pages:** Esta es la carpeta principal de los proyectos realizados en *Nextjs*. Cualquier carpeta que se cree dentro de esta con un fichero *index.js* (o en su defecto *[parametro].js* para rutas dinámicas, siendo *parámetro* será el elemento dinámico) se convertirá en una ruta de la aplicación. En el caso de este proyecto se ha decidido almacenar en cada carpeta, además de su fichero *index.js* correspondiente, un fichero *nombre.module.css* donde se encuentran los estilos de cada página. El uso de *module* permite asegurar que no habrá sobreescritura de estilos entre un componente y otro, al añadir un *hash* único a los identificadores. Además de lo mencionado anteriormente, en esta carpeta también se alojan los archivos de carácter especial que ofrece *Nextjs* listados a continuación:
  - **\_app.js:** Este archivo está relacionado con *AppLayout* mencionado previamente, ya que aquí es dónde se deberá referenciar a este y a los estilos globales de la aplicación. Toda la información que se encuentra en este fichero es global para el resto de la página.
  - **404.js y 500.js:** Estos archivos se refieren a las rutas de error cuando ocurra un fallo 404 o 500 en la aplicación.
  - **index.js:** Este archivo representa a la *landing page* de la aplicación, es decir la pantalla que aparecerá por primera vez al entrar a la aplicación y que representará a la ruta *"/*".
- **public:** En esta carpeta solo se almacenan elementos multimedia que utilizará la aplicación. En este caso solo se ha contado con el icono de la página que se muestra en la ventana del navegador.
- **styles:** En esta carpeta se encuentran todas las hojas de estilo que afectan globalmente a la página o que afectan a los archivos especiales como las páginas de error.

Una vez terminada la estructuración del proyecto, se comenzó con su desarrollo. Para ello primero se tuvieron que instalar dos extensiones extra de React: React OAuth Google (Momen, 2022), para el sistema de identificación dentro de la aplicación y React Icons (React Icons, 2022), para tener una librería de iconos. Por último, se adaptaron todas las pantallas de los prototipados de alta fidelidad elaborados en el capítulo anterior, siguiendo en el desarrollo la misma metodología de *mobile-first* que se utilizó para el diseño. En este caso, primero se elaboró la aplicación entera para dispositivos móviles de las dimensiones

establecidas, y después se realizaron los cambios necesarios tanto en los archivos de JavaScript como en las hojas de estilo necesarios para que también estuviera disponible para versión de escritorio. Durante este proceso se llevaron a cabo las siguientes consideraciones que no se había reflejado o especificado explícitamente previamente:

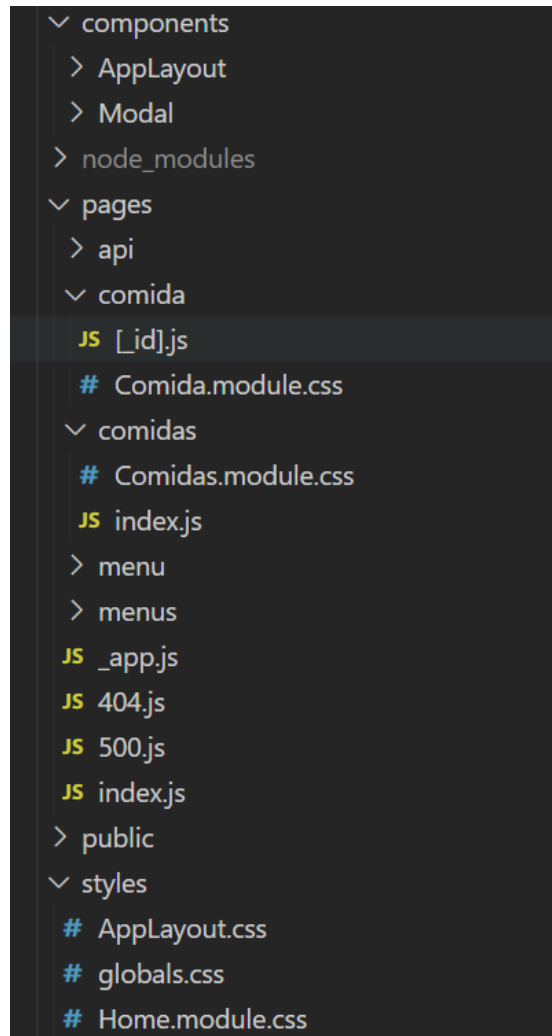


Figura 23 Estructura de carpetas del frontend

- Se encontró que a la hora de elaborar el diseño no se había tenido en cuenta que el usuario necesitaría un botón para cerrar sesión. Por ello, para una experiencia intuitiva, se decidió colocar este en la página principal de la aplicación que aparece justo cuando inicias sesión. De esa manera el usuario sabría dónde está nada más comenzar a hacer uso de la aplicación.
- Se observó que no se había considerado las páginas de error por lo que se elaboraron dos páginas simples para los errores 404 y 500. En ellas se notifica que se ha producido un fallo, el tipo de este y un botón redireccionando a la página principal.
- Respecto a la visibilidad de las páginas, para los usuarios que no hayan iniciado sesión, se ha decidido optar porque todo usuario con sesión no iniciada pueda ver todas las páginas, pero con todas las funcionalidades de creación y ocultas. De esta manera, cualquier usuario podrá hacer una

navegación inicial por la aplicación antes de decidir si quiere usarla o no. Así también se permite que un usuario pueda compartir su menú con otro a través de la url.

- Por último, para la elección de los filtros se ha decidido que lo más intuitivo y cómodo para el usuario es que los filtros de gastronomía actúen en forma del operador OR lógico, pero sin embargo que los filtros de alérgenos actúen en forma de AND lógico. Es imposible que un alimento tenga más de una gastronomía por lo que al seleccionar varias lo que se busca es obtener resultado de varias. Sin embargo, en cuanto a componentes alimenticios el usuario quiere que cuando busque por sus intolerancias los resultados cumplan todas sus necesidades, es decir, si un usuario es celíaco e intolerante a la lactosa busca que todos los alimentos que obtenga con el filtro los pueda comer. Para ello es necesario que se cumplan las dos condiciones a la vez.

# 8

## Conclusión y trabajos futuros

### 8.1 Introducción

Para concluir la redacción de la memoria de este proyecto, a continuación se reflexionará sobre las conclusiones y objetivos cumplidos. Después se procederá a mencionar diferentes apartados que se considera que se podrían extender o modificar en trabajos futuros sobre esta aplicación.

### 8.2 Conclusiones y objetivos cumplidos

Como se explicó en las motivaciones del trabajo, las intolerancias alimenticias son un problema muy importante en la actualidad. Por ello, el objetivo principal de este proyecto siempre ha sido proporcionar a las personas que conviven con estos problemas una herramienta para organizar sus menús semanales. En el transcurso de este proyecto se han propuesto varias metas que en conjunto puedan solventar ese problema.

Desde el punto de vista de diseño y experiencia de usuario, se definió una persona de ejemplo en la que centrarse para satisfacer a todos los clientes de ese sector de la población. Se considera que tanto el aspecto como la experiencia de usuario resultante cumplen con las cualidades que este tipo de usuario requeriría y por tanto que los objetivos del diseño se han cumplido.

Desde el punto de vista de los requisitos y casos de usos, también se considera que se han cumplido correctamente con estos pues todos los requisitos y casos de usos derivados de ellos, propuestos al inicio del proyecto, se han implementado.

Por último, pero no menos importante, desde el punto de vista académico y tecnológico se considera que se han logrado completar las metas previstas con en el desarrollo de este proyecto, pues debido al carácter multidisciplinar de este trabajo se considera que se han aplicado muchos conocimientos estudiados en la carrera. Además se han

aprendido muchos otros nuevos gracias a la elaboración de esta aplicación. Gracias a que las tecnologías utilizadas son muy populares en la actualidad, estos conocimientos son de gran utilidad y, si se deseara seguir aumentando las funcionalidades de este producto, se simplificaría mucho la tarea debido al hecho de que hay muchos recursos donde poder apoyarse.

### 8.3 Dificultades encontradas

Debido a la gran escala de este proyecto se han encontrado varias dificultades que se han tenido que superar durante el desarrollo. Inicialmente una vez elegida la temática de la aplicación fue complicado elegir cómo enfocar y estructurar el proyecto pues era la primera vez que el autor de este trabajo se veía en la situación de desarrollar un trabajo de carácter profesional, de forma individual. Esto también se vio reflejado a la hora de seleccionar cuáles serían las tecnologías a utilizar, pues no se poseía experiencia con muchas de ellas y era necesario realizar una selección, a pesar de no contar con mucho conocimiento sobre ellas, y aprenderla.

A continuación, una vez sentadas tanto las bases como las tecnologías que usaría el proyecto, se afrontaron dificultades a la hora de aprender diversas tecnologías y conocimientos de diversos campos de forma simultánea. Respecto a aprender conocimientos nuevos, lo más costoso fue desarrollar por primera vez una interfaz gráfica teniendo en cuenta de forma directa factores de diseño gráfico y otros como la experiencia que tendría el usuario al utilizar la aplicación. Respecto a tecnologías, debido a que antes de aprender a usar Nextjs se aprendió a usar React, al principio fue un poco difícil hacer la transición de una metodología a otra pues este primero añade funcionalidades que hacen que cambie la forma en la que se desarrolla el código como es tener en cuenta el *server side rendering*.

Por último, pero no menos importante, el proceso de aprender a organizar las tareas en el tiempo fue largo ya que el ritmo de trabajo no siempre era el mismo, debido al propio constante proceso de aprendizaje. A veces tareas aparentemente simples que parecían requerir menos tiempo al final llevaban más por la necesidad de aprender conocimientos nuevos o a hacer las cosas de maneras diferentes.

### 8.4 Trabajos futuros

Como ya se ha mencionado en el apartado anterior se considera que la temática del trabajo es muy actual, además de hacer uso de tecnologías muy innovadoras. Por ello hay muchas maneras en las que este proyecto se podría seguir actualizando. A continuación, se mencionarán varias de acuerdo a la experiencia asimilada realizando el proyecto:

- Sería interesante añadir algunas de las funcionalidades que se observaron en el estado del arte de otras herramientas parecidas pero que, por razones de tiempo o importancia, no se han incluido en este proyecto. Algunas de ellas serían: la recomendación de alimentos, la disponibilidad de la aplicación en más de un idioma o la posibilidad de poder compartir menús. Referente a esta última, dado que los menús ya son compartibles vía enlace, se buscaría mejorar esta funcionalidad mediante la implementación de una pantalla donde poder buscar menús por usuario. Alternativamente, se podría buscar directamente por usuario, acceder a su perfil y ver sus menús.

- Respecto a la personalización de la herramienta, se podrían añadir en futuras iteraciones la funcionalidad de que cada usuario contara con una pestaña de perfil o que los usuarios pudieran modificar la información base de sus menús (cambiar foto, editar nombre y descripción, borrar un menú...). También se podría añadir la disponibilidad de un modo oscuro de la interfaz.
- En lo referente a la base de datos, se podría aumentar el catálogo de alimentos que ofrece la aplicación, así como la variedad de alérgenos, gastronomías y etiquetas que ofrece. También, después de terminar con el desarrollo de esta versión, se considera que cambiar el sistema de almacenamiento de no relacional a mixto podría ser beneficioso. De esta manera, hay información que se podría estandarizar, como los alérgenos, pasando a almacenarlos en un sistema relacional o simplemente cambiar la manera en la que se relacionan los elementos. Así se aprovecharían las ventajas tanto del sistema no relacional, como ya se hace en el proyecto, y sumar las ventajas que podría ofrecer hacer uso de un sistema relacional.
- A nivel visual de la herramienta, sería beneficioso que se añadieran pantallas de carga para cuando se están haciendo transiciones entre páginas. Esta funcionalidad tendrá más importancia a medida que el contenido de la base de datos se expanda, pues los tiempos de carga pasarán a ser mayores.
- Otra actualización que podría ser positiva para la aplicación será el desplegarla en la nube tanto a nivel de código para no tener que hacer uso de un servidor local para desplegarla como a nivel de base de datos ya que en la versión actual, ambos se encuentran a nivel local.
- Por último, pero no menos importante también se considera que en futuras iteraciones cuando el número de pantallas y funcionalidades fuera más elevado se podría añadir análisis de calidad y pruebas al proyecto.

# Referencias

- Adobe Color. (2022a). *Herramientas de accesibilidad - Comprobador de contraste*.  
<https://color.adobe.com/es/create/color-contrast-analyzer>
- Adobe Color. (2022b). *Herramientas de accesibilidad - Seguro para daltónicos*.  
<https://color.adobe.com/es/create/color-accessibility>
- Adobe Color. (2022c). *Rueda Cromática*. <https://color.adobe.com/es/create/color-wheel>
- Arranz Santamaria, J. M. (2015, September 21). *The Single Page Interface Manifesto*.  
[http://itsnat.sourceforge.net/php/spim/spi\\_manifesto\\_en.php](http://itsnat.sourceforge.net/php/spim/spi_manifesto_en.php)
- Babich, N. (2016, May 7). *Primary & Secondary Action Buttons*.  
<https://uxplanet.org/primary-secondary-action-buttons-c16df9b36150>
- Cloudinary. (2022). *Cloudinary*. <https://cloudinary.com/>
- Dogtiev, A. (2022, June 30). *Mobile Advertising Rates (2022)*.  
[https://www.businessofapps.com/ads/research/mobile-app-advertising-cpm-rates/?zd\\_source=mta&zd\\_campaign=13041&zd\\_term=matto%20connor](https://www.businessofapps.com/ads/research/mobile-app-advertising-cpm-rates/?zd_source=mta&zd_campaign=13041&zd_term=matto%20connor)
- Easy Menu Planner. (2022). *Easy Menu Planner*.  
<https://easymenuplanner.app/es.html>
- Enkilu. (2022). *Enkilu*. <https://enkilu.com/es>
- Faller, P. (2019, December 17). *Putting Personas to Work in UX Design: What They Are and Why They're Important*. <https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design/>
- Figma. (2022a). *About Figma*. <https://www.figma.com/about/>
- Figma. (2022b). *Figma Mirror*. <https://www.figma.com/mirror>
- Fiocchi, A., & Fierro, V. (2017, March). *WAO Food Allergy*.  
<https://www.worldallergy.org/education-and-programs/education/allergic-disease-resource-center/professionals/food-allergy>
- Flask. (2022). Flask Project. In <https://github.com/pallets/flask>

Git. (2022). *Git*. <https://git-scm.com/>. <https://git-scm.com/>

GitHub. (2022). *GitHub*. <https://github.com/>. <https://github.com/>

Google. (2022a). *Google Fonts*. <https://fonts.google.com/>. <https://fonts.google.com/>

Google. (2022b, April 21). *Prácticas recomendadas para la indexación centrada en los móviles*. <https://developers.google.com/search/mobile-sites/mobile-first-indexing>. <https://developers.google.com/search/mobile-sites/mobile-first-indexing>

Google Cloud Platform. (2021, March 4). *google-auth-httpplib*. <https://pypi.org/project/google-auth-httpplib2/>. <https://pypi.org/project/google-auth-httpplib2/>

Google Cloud Platform. (2022, September 14). *google-auth-oauthlib*. <https://pypi.org/project/google-auth-oauthlib/>. <https://pypi.org/project/google-auth-oauthlib/>

Handley, L. (2019, January 24). *Nearly three quarters of the world will use just their smartphones to access the internet by 2025*. <https://www.cnbc.com/2019/01/24/smartphones-72percent-of-people-will-use-only-mobile-for-internet-by-2025.html>. <https://www.cnbc.com/2019/01/24/smartphones-72percent-of-people-will-use-only-mobile-for-internet-by-2025.html>

Ignatiev, I., & Penev, V. (2022). *Edamam*. <https://www.edamam.com/>. <https://www.edamam.com/>

Kong. (2022). *Insomnia*. <https://insomnia.rest/>. <https://insomnia.rest/>

Material Design. (2022). *Cards*. <https://material.io/components/cards>. <https://material.io/components/cards>

MDN contributors. (2021, July 7). *CSS*. <https://developer.mozilla.org/es/docs/Web/CSS>. <https://developer.mozilla.org/es/docs/Web/CSS>

MDN contributors. (2022a). *HTML: Lenguaje de etiquetas de hipertexto*. <https://developer.mozilla.org/es/docs/Web/HTML>. <https://developer.mozilla.org/es/docs/Web/HTML>

MDN contributors. (2022b, May 30). *JavaScript*. <https://developer.mozilla.org/es/docs/Web/JavaScript>. <https://developer.mozilla.org/es/docs/Web/JavaScript>

Momen, S. (2022, September 19). *React OAuth2 | Google*. <https://github.com/MomenSherif/React-Oauth>. <https://www.npmjs.com/package/@react-oauth/google>

MongoDB. (2022a). *Mongo Database*. <https://www.mongodb.com/atlas/database>. <https://www.mongodb.com/atlas/database>

MongoDB. (2022b). *Relational vs. Non-Relational Databases*. <https://www.mongodb.com/compare/relational-vs-non-relational-databases>. <https://www.mongodb.com/compare/relational-vs-non-relational-databases>

MongoDB. (2022c). *When to Use a NoSQL Database*. <https://www.mongodb.com/nosql-explained/when-to-use-nosql>. <https://www.mongodb.com/nosql-explained/when-to-use-nosql>

Morales, J. (2021, February 16). *Mobile First Design Strategy: The When, Why and How*. <https://xd.adobe.com/ideas/process/ui-design/what-is-mobile-first-design/>. <https://xd.adobe.com/ideas/process/ui-design/what-is-mobile-first-design/>

Nestle. (2022). *Nestle Menú Planner*. <https://Www.Nestlemenuplanner.Es/Menu-Semanal.Html>. <https://www.nestlemenuplanner.es/menu-semanal.html>

NoMagic. (2022). *Software Engineering Solutions*. <https://Www.3ds.Com/Products-Services/Catia/Products/No-Magic/Solutions/Software-Engineering/>. <https://www.3ds.com/products-services/catia/products/no-magic/solutions/software-engineering/>

Notion. (2022, June 21). *Notion*. <https://Www.Notion.so/Es-Es>. <https://www.notion.so/es-es>

Nutrition Addition. (2022). *Nutrition Addition*. <https://Www.Nutritionaddition.Com/>. <https://www.nutritionaddition.com/>

Pexels. (2022). *Pexels*. <https://Www.Pexels.Com/Es-Es/>. <https://www.pexels.com/es-es/>

Planifood. (2022). *Planifood*. <https://Www.Planifood.Com/>. <https://www.planifood.com/>

Python. (2022). *Applications for Python*. <https://Www.Python.Org/about/Apps/>. <https://www.python.org/about/apps/>

React. (2022). *React Una biblioteca de JavaScript para construir interfaces de usuario*. <https://Es.Reactjs.Org/>. <https://es.reactjs.org/>

React Icons. (2022, September 4). *React Icons*. <https://React-Icons.Github.Io/React-Icons>. <https://react-icons.github.io/react-icons>

Recipe Calendar. (2022). *Recipe Calendar*. <https://Www.Recipe-Calendar.Com/>. Recipe Calendar

Recipe Keeper. (2022). *Recipe Keeper*. <https://Www.Recipekeeperonline.Com/>. <https://www.recipekeeperonline.com/>

Scott, D., & Bring Your Own Laptop. (2020). *Random Project Generator*. <https://Randomprojectgenerator.Com/>. <https://randomprojectgenerator.com/>

Sicherer, S. H. (2011). Epidemiology of food allergy. *Journal of Allergy and Clinical Immunology*, 127(3), 594–602. <https://doi.org/10.1016/j.jaci.2010.11.044>

Statista. (2022, June). *Mobile advertising spending worldwide from 2007 to 2024*. <https://Www.Statista.Com/Statistics/303817/Mobile-Internet-Advertising-Revenue-Worldwide/>. <https://www.statista.com/statistics/303817/mobile-internet-advertising-revenue-worldwide/>

Syzonenko, A. (2019, May 26). *Buttons on the web: placement and order*. <https://Uxdesign.Cc/Buttons-Placement-and-Order-Bb1c4abadfcb>. <https://uxdesign.cc/buttons-placement-and-order-bb1c4abadfcb>

Tang, M. L. K., & Mullins, R. J. (2017). Food allergy: is prevalence increasing? *Internal Medicine Journal*, 47(3), 256–261. <https://doi.org/10.1111/imj.13362>

TheDataDB. (2022). *TheMealDB*. <https://Www.Themealdb.Com/Api.Php>. <https://www.themealdb.com/api.php>

Unsplash. (2022). *Unsplash*. <https://Unsplash.Com/Es>. <https://unsplash.com/es>

Vercel. (2022). *The React Framework for Production*. <https://Nextjs.Org/>. <https://nextjs.org/>

Visual Studio Code. (2022). *Visual Studio Code Docs*. <https://Code.VisualStudio.Com/Docs>. <https://code.visualstudio.com/docs>

Worship. (2021). *Mobile screen sizes for 2021 based on data from 2020*. <https://Worship.Agency/Mobile-Screen-Sizes-for-2021>. <https://worship.agency/mobile-screen-sizes-for-2021>

Xia, V. (2017, December 21). *What is Mobile First Design? Why It's Important & How To Make It?* <https://Medium.Com/@Vincentxia77/What-Is-Mobile-First-Design-Why-Its-Important-How-to-Make-It-7d3cf2e29d00>.  
<https://medium.com/@Vincentxia77/what-is-mobile-first-design-why-its-important-how-to-make-it-7d3cf2e29d00>



# Apéndice A

# Manual de Instalación

## Requerimientos:

Para hacer uso de la herramienta en sus dispositivos es necesario contar con las siguientes especificaciones:

- MongoDB instalado (es recomendable tener la versión gráfica llamada Compass instalada para una mejor visualización de los datos).
- Contar con una cuenta de Google para poder realizar todos los procesos de autenticación de la herramienta.
- Python con la versión 3.6.7 instalada.
- Node con la versión 6 npm instalado.
- Visual Studio Code o cualquier otro editor de código; esta instalación asumirá que se está utilizando este.
- El sistema operativo también es opcional, pero en esta instalación se supondrá que se está utilizando Windows.
- Un proyecto de la API de Google creado en el que estén permitidas las IPs locales de los puertos 8080, 5000 o el puerto en el que se lance tanto el backend como el frontend, con una página de autenticación de Google Oauth creada.

## Instalación base de datos

Para esta parte solo será necesaria que se cree en MongoDB una base de datos llamada TFG, donde se crearán a su vez dos colecciones menús y comidas. En esta última se deberá introducir el fichero food.json de la carpeta backend/datasets.

## Instalación del backend

- Primero se deberá abrir el proyecto y dirigirse en la terminal a la carpeta backend. Para hacer esto se puede escribir `cd backend` en la terminal.
- Una vez se encuentre en la carpeta, creará un entorno virtual mediante el comando `python -m venv env`.
- Una vez haya creado el entorno virtual, deberá abrir la paleta de comandos. Para ello se deberá pulsar en vista y luego en paleta de comandos, como se ve en la Figura 24.
- A continuación, se seleccionará "Python: Seleccionar intérprete" y ahí el que empieza con `/env` o `\env`. Si no se visualiza, se deberá pulsar sobre

"Introducir directorio del intérprete" y "Buscar", e introducir la ruta ".\backend\env\Scripts\python.exe" o "\env\Scripts\python.exe".

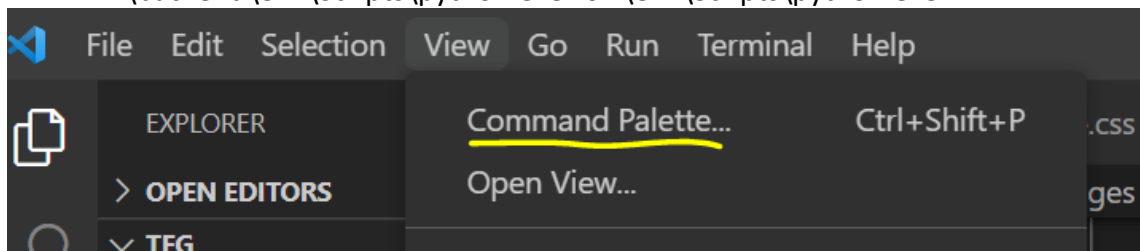


Figura 24 Seleccionando la Paleta de comandos en Visual Studio Code

- Después de haber realizado esto, se creará una nueva terminal mediante (Ctrl+Shift+`) o pulsando "Terminal: Crear nuevo terminal" de la paleta de Comandos.
- Nota: En Windows, si el terminal por defecto es PowerShell, a lo mejor se visualiza un error diciendo que no puede lanzar `activate.ps1` porque lanzar scripts está desactivado en el sistema. El error proporciona un enlace con información para activar scripts. Si no, se usará Terminal: Seleccionar Shell como predeterminado para configurar Command Prompt(cmd) o Git bash como su terminal por defecto. Si se sigue teniendo problemas, se puede crear manualmente una terminal cmd y usar el comando `env/scripts/activate` para activar el entorno virtual.
- Una vez hecho todo esto deberá aparecerá el nombre del entorno virtual entre paréntesis, como se ve en la Figura 25.

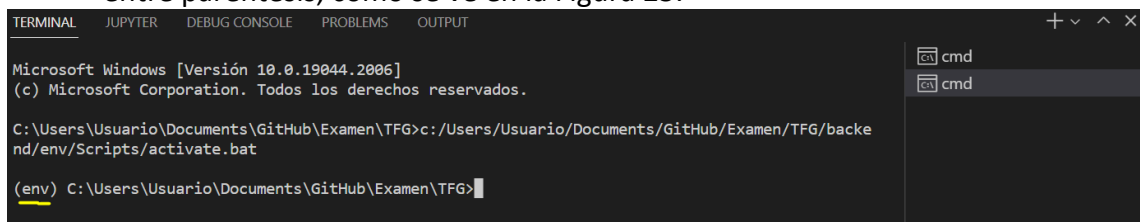


Figura 25 Entorno virtual activado en la terminal de Visual Studio Code.

- Una vez realizados estos pasos previos y garantizado que se encuentra en el entorno virtual adecuado, se deberán instalar los paquetes necesarios para usar la aplicación. Para esto simplemente se deberá ejecutar el comando `pip install -r requirements.txt` verificando que se está en la carpeta de backend.
- A continuación, se crea en app un archivo `config.py` en la carpeta app de backend con el contenido de la Figura 26. Tras el símbolo del igual, se introducirá el id del cliente de Google Auth en formato string.

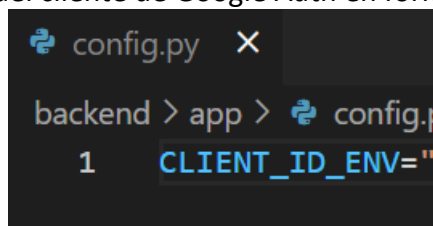


Figura 26 Fichero de `config.py` necesario para usar la API de Google

- Una vez realizados todos estos pasos, se podrá utilizar el backend de la aplicación.

## Instalación del frontend

- Primero se deberá abrir el proyecto y dirigirse en la terminal a la carpeta `frontend/tfg`. Para hacer esto se puede escribir `cd frontend/tfg` en la terminal.
- Una vez se encuentre en esta carpeta, se deberá usar el comando `npm install` para instalar todas las dependencias necesarias del proyecto.
- Una vez realizado esto, se deberá crear un archivo `.env.local` con el contenido de la Figura 27. Tras el símbolo del igual irá la id del cliente de Google Auth.

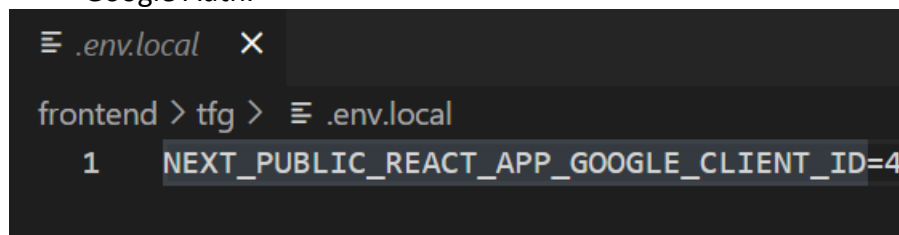
A screenshot of a terminal window with a dark background. The window title is ".env.local" with a close button (X) on the right. The terminal shows the current directory path as "frontend > tfg > .env.local". Below the path, there is a single line of code: "1 NEXT\_PUBLIC\_REACT\_APP\_GOOGLE\_CLIENT\_ID=4". The text "NEXT\_PUBLIC\_REACT\_APP\_GOOGLE\_CLIENT\_ID=4" is highlighted with a light blue selection box.

Figura 27 Fichero `.env.local` en el que se almacena la id del cliente de Google Auth

- Una vez realizados todos estos pasos, se podrá utilizar el frontend de la aplicación, asegurándose siempre que previamente se ha lanzado el backend de esta.

## Despliegue de la aplicación

Para desplegar la aplicación, se deberá tener iniciado MongoDB con la base de datos y tener dos terminales. En la primera se encontrará en el fichero `frontend/tfg` y en la segunda el fichero `backend`. Una vez realizado esto, se realizará lo siguiente en el orden indicado:

- Lanzar el script `openbackend` en la terminal del backend.
- Lanzar el comando `npm run dev` en la terminal del frontend.

# Apéndice B

## Manual de usuario

### Introducción

A continuación, se hará una descripción de las funcionalidades y del funcionamiento de cada página de la aplicación para que el usuario pueda hacer un uso correcto de esta.

### Página principal de la aplicación

Nada más entrar a la aplicación el usuario se encontrará con el botón de inicio de sesión con Google (Figura 28 y Figura 29). Dado que la aplicación permite el acceso aun sin registrarse, si el usuario lo desea podrá acceder mediante la navegación del menú que se encuentra en la barra superior. Dicha barra de navegación contiene redirecciones a la página de "Mis menús" y "Comidas", aunque si no se ha iniciado sesión se tendrá una vista limitada de los contenidos de ambas. De encontrarse fuera de la pantalla principal, el usuario también podrá volver a esta a través del logo.



Figura 28 Pantalla de inicio sin sesión versión móvil



Figura 29 Pantalla de inicio sin sesión con barra navegación abierta versión ordenador



Figura 30 Barra de navegación abierta versión móvil

En el caso de que el usuario inicie sesión, el estado de la página principal se actualizará al estado de las Figura 31 y Figura 32, y serán visibles dos botones para redireccionar a las mismas páginas que ofrecía la barra de navegación, además de uno adicional para cerrar sesión en caso de que así se desee.



Figura 31 Pantalla de inicio con sesión versión móvil



Figura 32 Pantalla de inicio sin sesión versión ordenador

## Página de Mis menús

Lo primero que verá el usuario al entrar a esta página será su lista de menús, como se ve en las Figura 33 y Figura 34. En caso de que desee crear un nuevo menú, solo tendrá que pulsar sobre el botón de las Figura 35 y Figura 36, y rellenar los campos que son obligatorios. En caso contrario, la aplicación le notificará que estos deben ser rellenados. También si el usuario así lo deseará, podría acceder a la información de cualquiera de sus menús con solo pulsar en uno de ellos y sería redireccionado a la página con la información específica de este. Por último, en caso de que se accediera a esta página sin una sesión iniciada, se le notificara al usuario, como se ve en la Figura 37.



Figura 33 Página de Mis Menús versión móvil

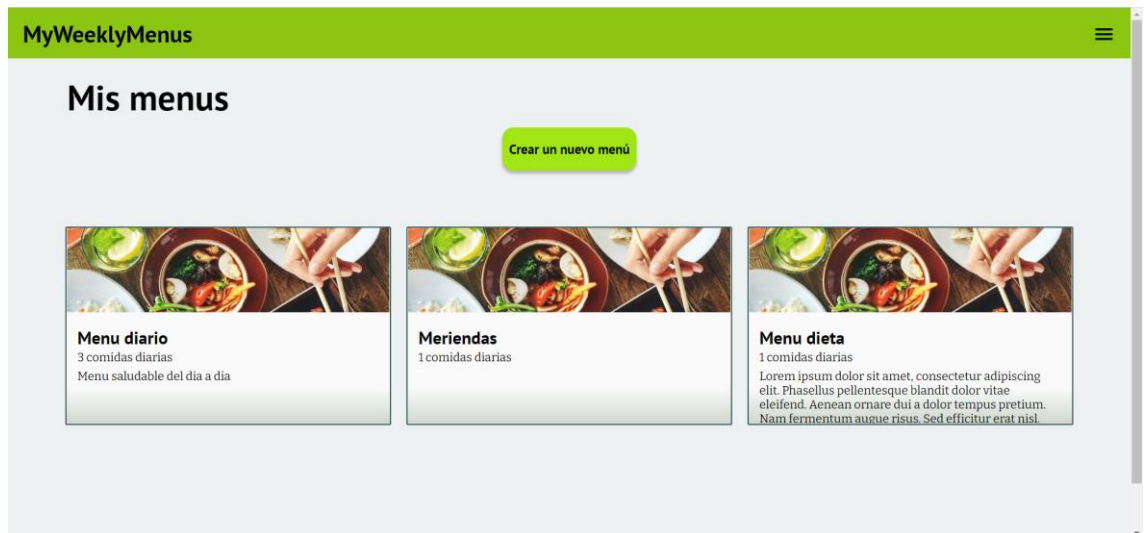


Figura 34 Página de Mis Menús versión ordenador

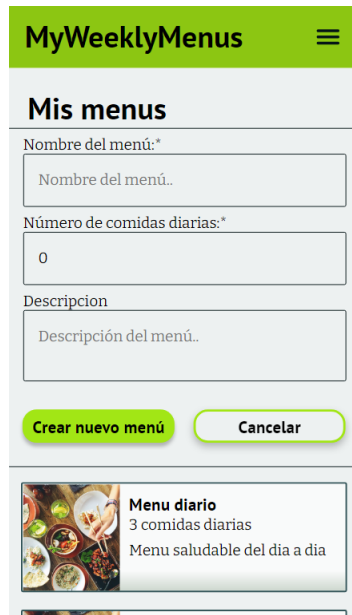


Figura 35 Página mis menús formulario abierto versión móvil

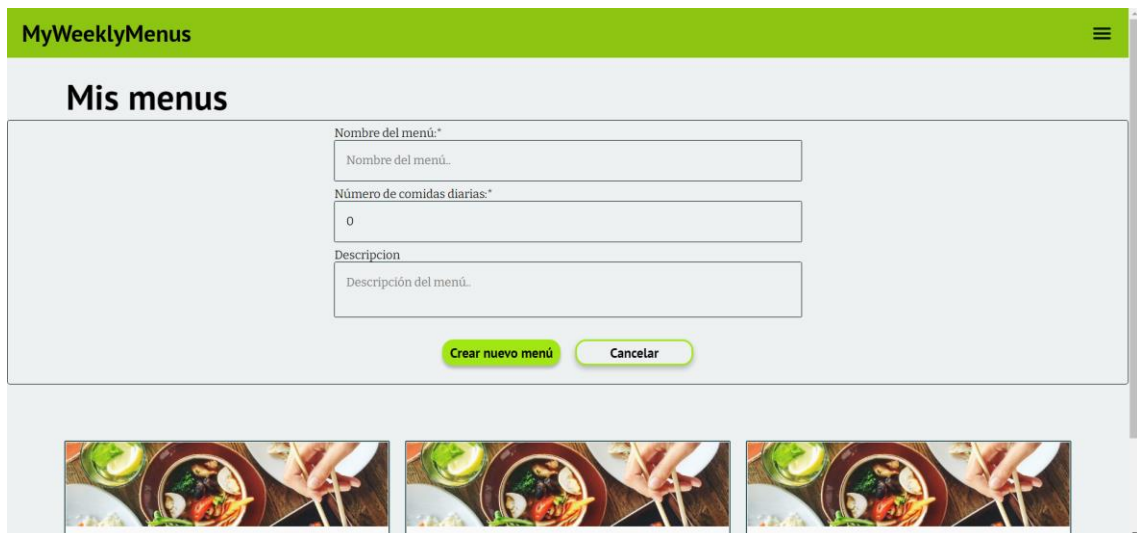


Figura 36 Página mis menús formulario abierto versión ordenador

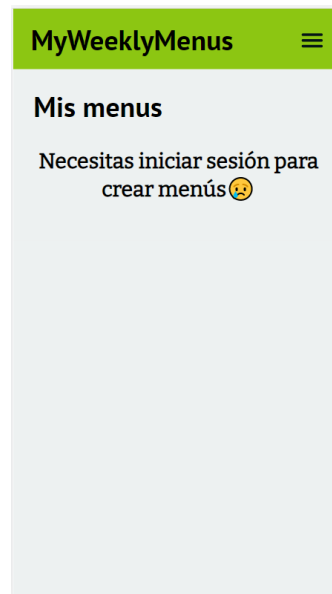


Figura 37 Página Mis Menús sesión sin iniciar

## Página información de un menú

Cuando el usuario accede a la información de un menú será redireccionado a una página como la de las Figura 38 y Figura 39, donde podrá navegar por los días de la semana mediante una desplegable en versión móvil o una barra de navegación en ordenador. En caso de que el hueco de una comida estuviera vacío, dicha tarjeta aparecerá vacía con un botón que le redirecciona al catálogo de comidas, como se observa en la Figura 40 y Figura 41. Por último, en caso de que se introdujera un enlace de un menú que no existiera, la aplicación redirigiría al usuario a la pantalla de error 404 de la Figura 42.



Figura 38 Información de un menú versión móvil



Figura 39 Información de un menú versión ordenador

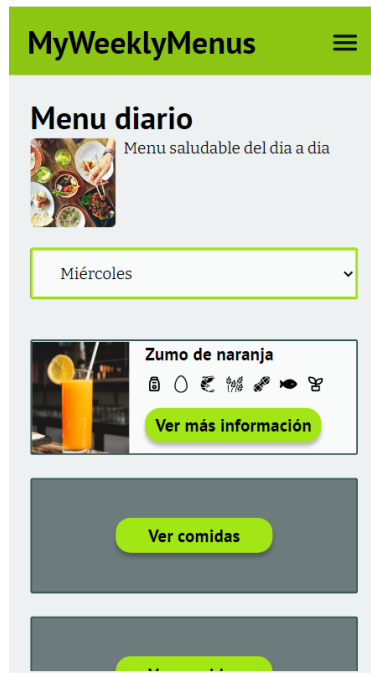


Figura 40 Información de un menú versión móvil (comidas vacías)



Figura 41 Información de un menú versión ordenador (comidas vacías)



Figura 42 Página de error 404

## Página de Comidas

Al entrar en la página de comidas inicialmente se verán todas las comidas que ofrece la aplicación, como se observa en las Figura 43 y Figura 44. Si el usuario lo desea, podrá buscar alimentos introduciendo la cadena de caracteres que quiere que contengan estos, en el nombre en la barra de búsqueda. Si en algún momento deseara volver a ver todo el catálogo de alimentos de nuevo, solo tendrá que buscar la cadena vacía. También si el usuario lo desea podrá hacer búsquedas más intensivas mediante el uso de los filtros. Pulsando en el botón de embudo, podrá acceder al formulario de filtros de las Figura 45 y Figura 46. A través de él, filtrará los alimentos por el texto que se encuentre en la barra de búsqueda y por las categorías que seleccionen en el formulario. En el caso de alérgenos, buscará solo las que cumplan todos esos requisitos. Sin embargo, en el caso de gastronomía, buscará todos los alimentos que pertenezcan a alguna de esas categorías.

Por otro, todos los alimentos cuentan con dos botones: botón de *Ver más*, que si es pulsado será redireccionado a una página con la información de ese alimento, y botón de *Añadir* (solo visible si el usuario tiene sesión iniciada) que si lo pulsa, se le abrirá un cuadro de diálogo, como el de las Figura 47 y Figura 48, donde seleccionará el menú, el día de la semana y la comida del día en el que se desea añadir el alimento.



Figura 43 Página de comidas versión móvil

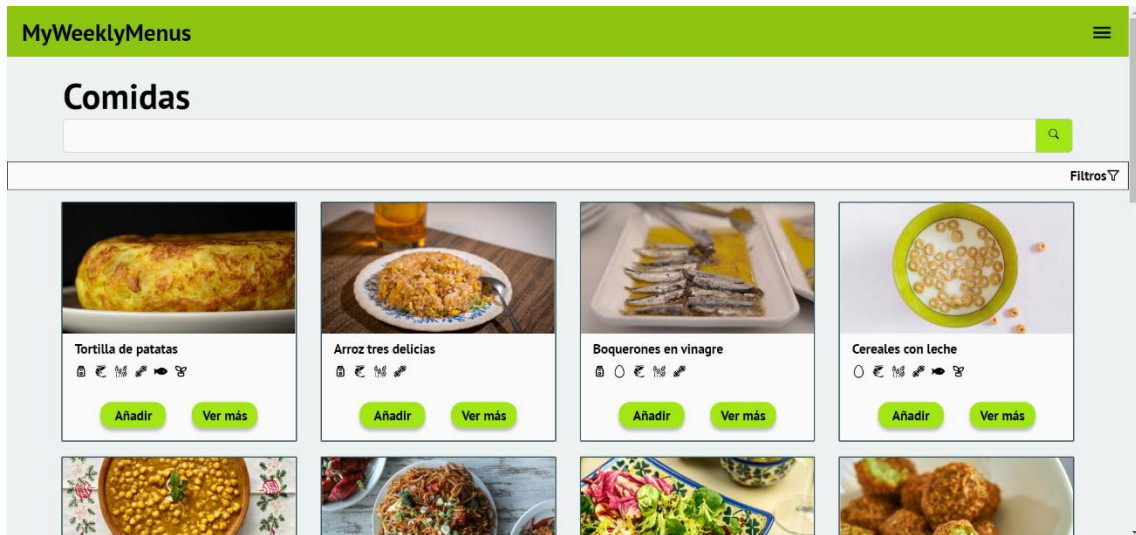


Figura 44 Página de comidas versión ordenador



Figura 45 Filtros de comidas versión móvil

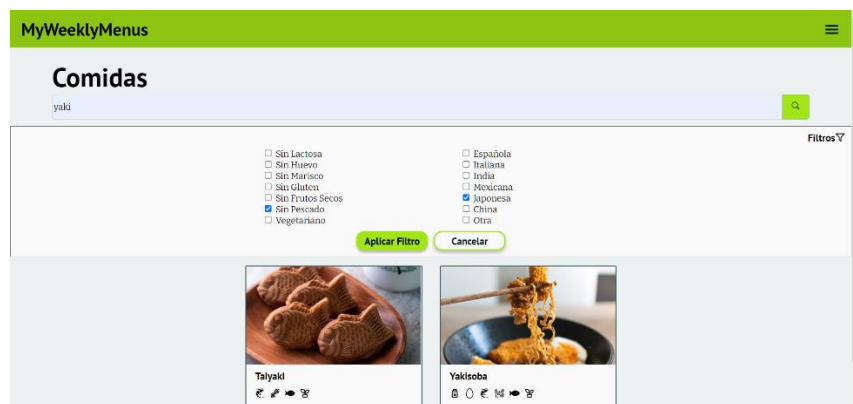


Figura 46 Filtros de comidas versión ordenador

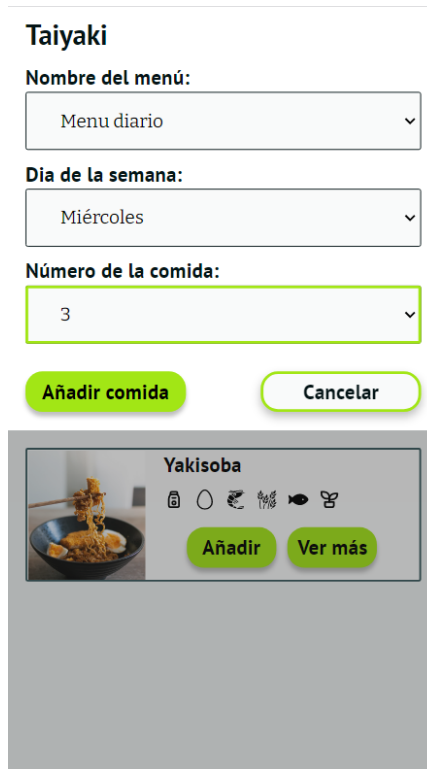


Figura 47 Ventana de diálogo para añadir comida versión móvil

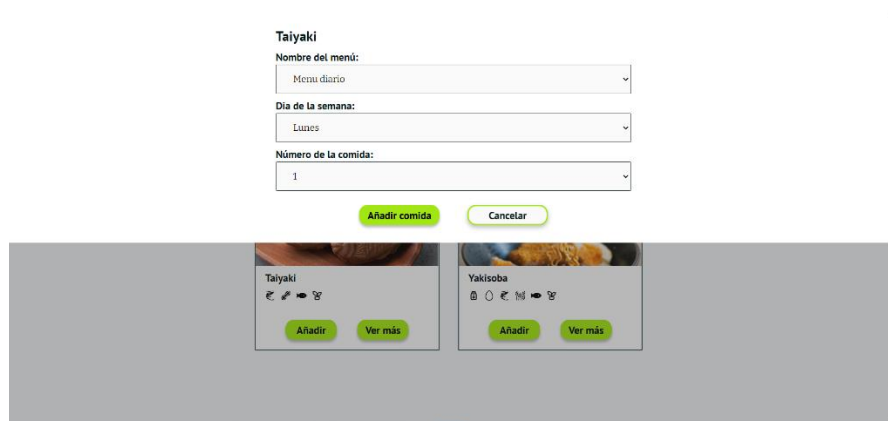


Figura 48 Filtros de comidas versión móvil

## Página de información de una comida

Al entrar en la página de información de una comida se mostrará toda la información respectiva a ella, como se ve en las Figura 49 y Figura 50. Si se intentara acceder a una dirección de una comida que no existe, al igual que con las páginas de información de los menús se redireccionaría al usuario a una página de error 404 como la vista anteriormente.



Figura 49 Página de información de una comida versión móvil

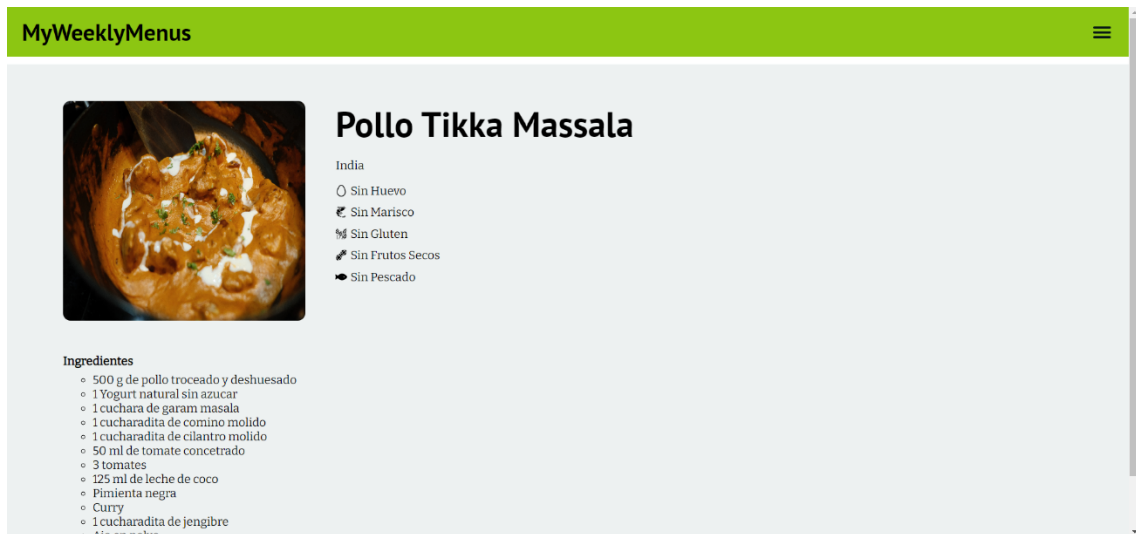


Figura 50 Página de información de una comida versión ordenador



UNIVERSIDAD  
DE MÁLAGA

| [uma.es](http://uma.es)

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA