



Universidad de Jaén

Escuela Politécnica
Superior de Jaén

LUNCH AT HOME

Autor: Francisco José Reyes Campos

Grado: Ingeniería Informática

Director: Prof. D. Miguel Ángel García Cumbreiras
Departamento del director: Informática

Fecha: 15/04/2024

Licencia CC



CREA



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Miguel Ángel García Cumbreras, tutor del Proyecto Fin de Carrera titulado:
Lunch At Home, que presenta Francisco José Reyes Campos, autoriza su
presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Abril de 2024

El alumno:

Los tutores:

Francisco José Reyes Campos

Miguel Ángel García Cumbreras

Agradecimientos

A mis padres, hermana y toda mi familia, gracias por su amor y apoyo incondicional en todo momento. Cada logro ha sido posible gracias a su guía y sacrificio. Este logro también es suyo.

A mis amigos, esta etapa ha sido mucho mas fructífera y divertida gracias a vosotros. Las vivencias compartidas quedarán siempre en mi memoria, recordadas con mucho cariño.

Índice

1.	Introducción	5
1.1.	Contexto y justificación del proyecto	5
1.2.	Objetivos y alcances del proyecto	5
1.3.	Análisis del mercado	6
1.4.	Antecedentes y estado del arte	9
1.4.1.	Glovo	9
1.4.2.	Uber Eats	12
1.4.3.	Just Eats.....	14
1.5.	Requisitos y necesidades.....	18
1.6.	Hipótesis y restricciones.....	19
1.7.	Estudio de alternativas y viabilidad de cada una de las mismas	19
1.7.1.	Alternativas de Frontend	19
1.7.2.	Alternativas de Backend.....	22
1.8.	Descripción de la solución propuesta.....	27
1.9.	Tecnologías utilizadas	27
1.10.	Metodología de desarrollo de software	28
1.11.	Estimación de tamaño y esfuerzo	29
1.11.	Diagrama de Gantt.....	31
2.	Inicio del proyecto.....	34
2.1.	Requisitos iniciales.....	34
2.2.	Casos de uso	36
2.3.	Análisis y diseño del sistema.....	42
2.4.	Diagrama de clases.....	44
2.5.	Diseño arquitectónico preliminar	45
3.	Desarrollo del proyecto.....	72
3.1.	Primera iteración	72
3.2.	Segunda iteración.....	88
3.3.	Tercera iteración.....	101
3.4.	Pruebas de usabilidad	115

4.	Resultados y discusión	119
4.1.	Resultado y resumen final del progreso de aprendizaje.....	119
5.	Conclusiones y trabajo futuro	121
BIBLIOGRAFÍA Y REFERENCIAS WEB.....		123

1. Introducción

1.1. Contexto y justificación del proyecto

En el ajetreo cotidiano de la vida universitaria, donde los estudiantes tienen poco tiempo y tienen muchos deberes, surge una necesidad básica: la búsqueda de una alimentación equilibrada. Siendo el tiempo un recurso escaso y donde la preparación de comidas saludables quedan en segundo plano, nace “Lunch At Home”, una aplicación diseñada con la premisa de satisfacer las demandas nutricionales de los estudiantes universitarios.

Como base de este trabajo, está la idea de que los estudiantes necesitan una dieta adecuada, esencial para el rendimiento académico y el bienestar general. “Lunch At Home” se presenta como una respuesta de conveniencia y salud, proporcionando a los estudiantes una solución para sus necesidades alimenticias. Este proyecto intenta ir más allá de ser un simple servicio de entrega a domicilio, facilita la vida de los estudiantes, permitiendo a los usuarios personalizar sus pedidos de acuerdo a criterios específicos, tales como calorías, carbohidratos, entre otras cosas.

Por ello, se ha desarrollado una aplicación web que facilite la orden y entrega de alimentos por parte de los usuarios, restaurantes y proveedores, convirtiéndola en una solución conveniente e innovadora para los estudiantes. Esta aplicación tiene el potencial de generar un efecto beneficioso en el ámbito de la alimentación y la salud, proporcionando comodidad a los usuarios y optimizando el tiempo de los mismos.

1.2. Objetivos y alcances del proyecto

La propuesta consiste en optimizar la experiencia del usuario tanto como sea posible. Ante la falta de información y alternativas, buscamos un enfoque tecnológico

para abordarlos. En concreto, la iniciativa implica el desarrollo de una aplicación tipo Marketplace con roles diferenciados, buscando integrar de manera efectiva la demanda de los usuarios y las ofertas de los restaurantes.

Por tanto, nuestro **objetivo principal** consiste en el estudio y desarrollo de una plataforma web dedicada y diseñada específicamente para satisfacer las necesidades nutricionales de los estudiantes universitarios. Se les proporcionará la mayor información posible sobre el producto y mediante nuestro Marketplace tendrán distintas alternativas para poder llevar una dieta que se desee.

En cuanto al alcance del proyecto se toman en consideración los siguientes elementos:

- Desarrollar una aplicación web full-stack que permita la compra-venta de productos alimenticios entre clientes y restaurantes.
- Diseñar e implementar una interfaz, tanto de usuario como de restaurante, que permita tanto a los usuarios buscar productos con facilidad como a los restaurantes administrar sus productos y poner toda la documentación necesaria para el usuario.
- Implementar una infraestructura de datos que permita almacenar y gestionar información de todo tipo relacionada con clientes o restaurantes.
- Evaluación completa de nuestra aplicación para garantizar la experiencia, usabilidad y experiencia del usuario.
- Implementar medidas de seguridad robustas para proteger la información sensible de los usuarios y restaurantes, cumpliendo con los estándares de privacidad y protección de datos.
- Permitir que los usuarios personalicen sus perfiles y preferencias para experiencias individualizadas

1.3. Análisis del mercado

En el proceso de análisis de mercado, se ha llevado a cabo una exhaustiva revisión de diversas fuentes especializadas que investigan los Marketplaces de alimentos más usados en la actualidad.

Nos hemos enfocado en un artículo de El Economista <https://www.eleconomista.es/retail-consumo/noticias/11995155/10/22/Glovo-acapara-ya-el-25-del-mercado-de-los-agregadores-de-comida-a-domicilio.html> encontrando Marketplaces que tienen mayor similitud con nuestro proyecto, a pesar de que este último se enfoque en un ámbito más específico.

Este enfoque técnico nos ha permitido obtener una perspectiva detallada sobre las tendencias, estrategias y dinámicas competitivas presentes en el segmento de Marketplaces de comida, proporcionando así una valiosa información para nuestra causa:

- El 61% de los consumidores usa agregadores de comida a domicilio en nuestro país.
- El porcentaje es aún mayor del 73%, cuando hablamos de una población de entre 18 y 24 años, edad que comprenden los usuarios destinados de nuestro proyecto.
- Aproximadamente el 25% de los usuarios utiliza plataformas de entrega de alimentos semanalmente, siendo común la lealtad a un único agregador en el mercado español. Esto nos sugiere que, al consolidarse nuestro proyecto, nos podría generar resultados significativos, dado nuestro cliente objetivo principal.
- Empresas como Glovo, acaparan ya el 25% del mercado de los agregadores de comida a domicilio, siguiéndoles de cerca Uber Eats, con el 21% de la cuota y Just Eats, que acapara el 18%.
- En cuanto a preferencias, los platos favoritos a domicilio en nuestro país siguen siendo la pizza (64%) y las hamburguesas (59%) con Burger King, McDonald's y Telepizza como las marcas más demandadas.
- Casi la mitad de los usuarios (49%) reconocen que toman su decisión una vez que acceden a su agregador preferido.

A continuación, unas gráficas con la información más importante:

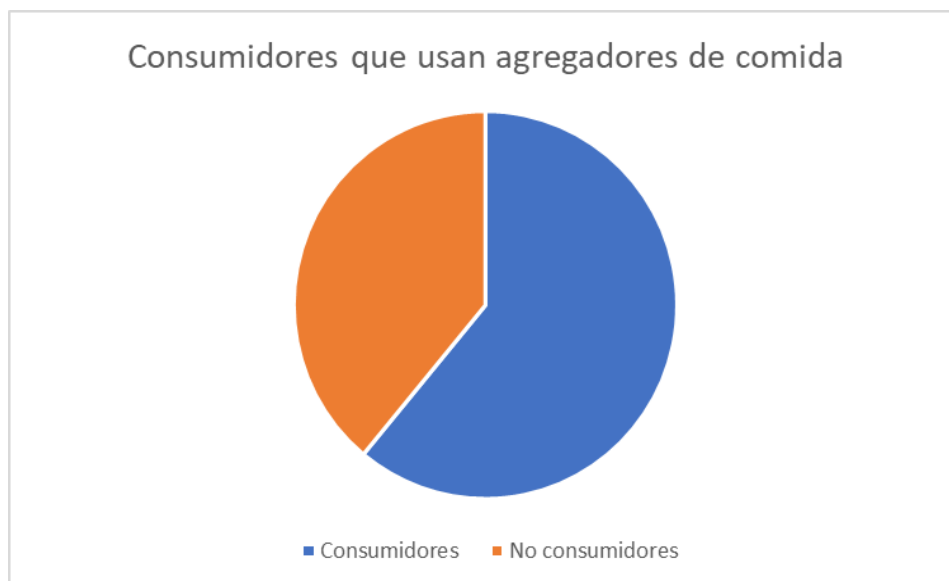


Figura 1: Gráfico de consumidores que usan agregadores de comida



Figura 2: Gráfico del mercado de agregadores

1.4. Antecedentes y estado del arte

Continuemos con un análisis de las diferentes páginas web de Marketplace de comida más populares. Exploraremos las 3 mencionadas anteriormente:

1.4.1. Glovo

Aquí tenemos una página web simple y sencilla con una interfaz intuitiva. Primero te pide mediante geolocalización una dirección (o se la puedes dar tú) y a continuación te muestran los restaurantes que hay en la zona. Dentro de estos restaurantes puedes elegir los distintos menús que hay. A continuación, pasas por la pasarela de pago y, por último, se tramita el pedido de manera segura mediante una transacción.

Las vistas principales son las siguientes:

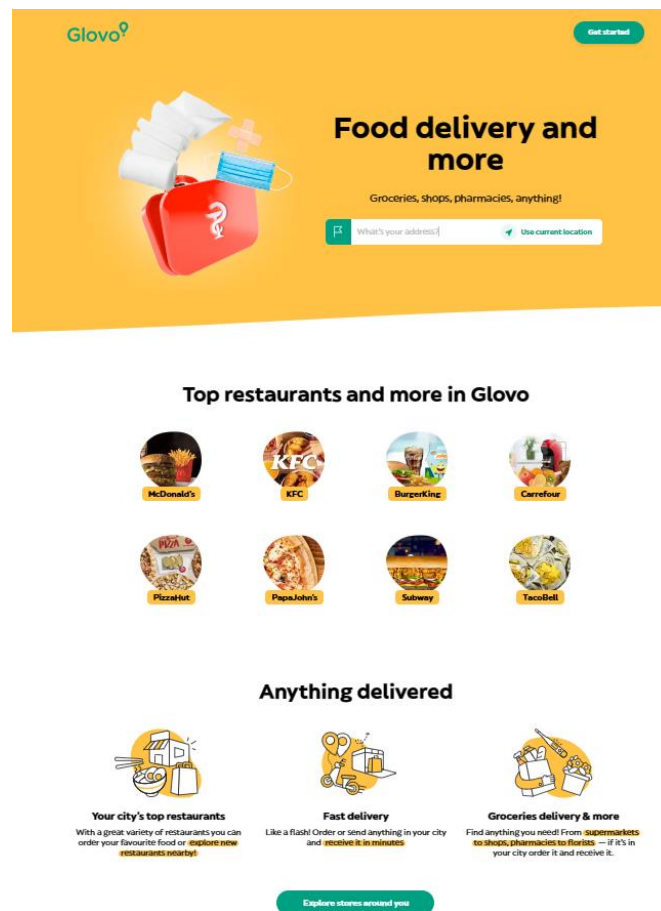


Figura 3: Vista principal de la plataforma web Glovo.

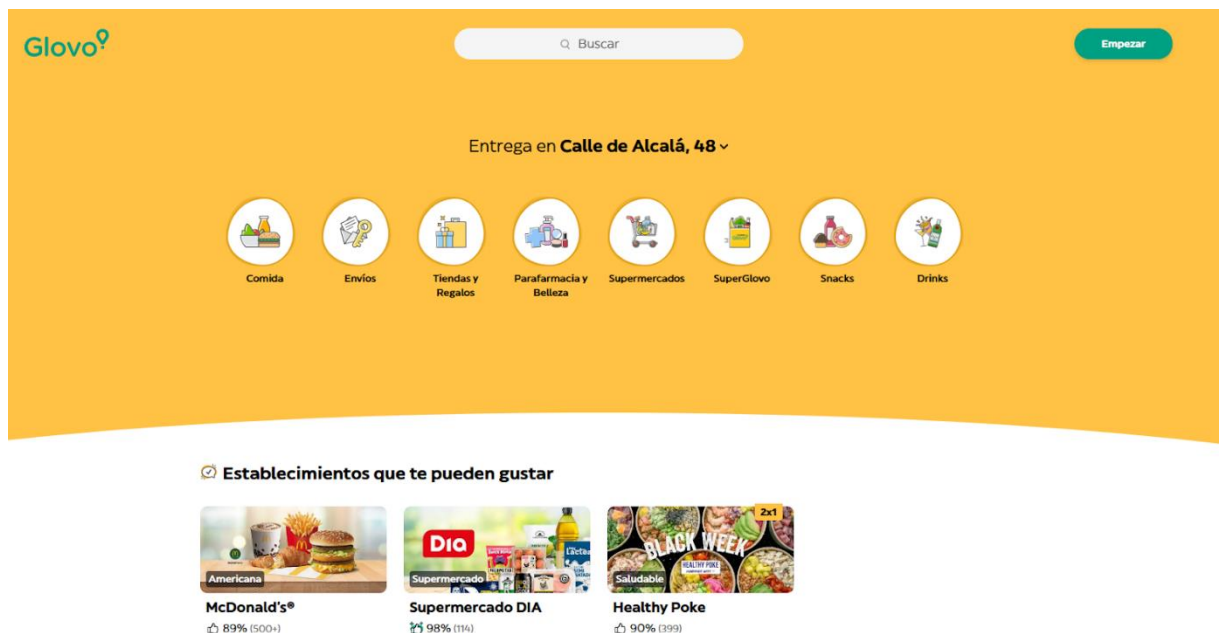


Figura 4: Vista personal de la plataforma web Glovo

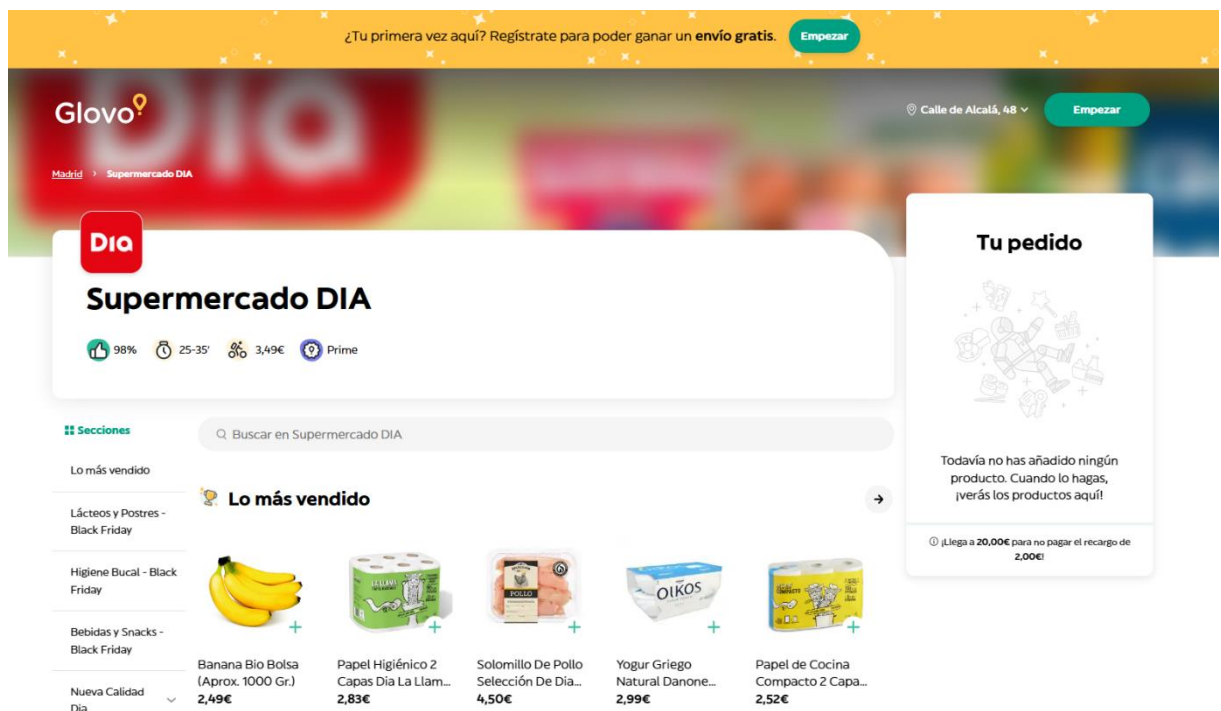


Figura 5: Vista de negocio de la plataforma web Glovo

Haciendo un análisis general, me ha gustado la sencillez de la interfaz y lo rápido que puedes realizar pedidos en Glovo. La disposición intuitiva de las opciones y la rapidez de la navegación contribuyen a una experiencia de usuario eficiente. Sin embargo, sería beneficioso profundizar en la personalización de perfiles de usuario para realizar algún sistema de recomendación en base a las preferencias del mismo.

Link de la plataforma web en cuestión: <https://glovoapp.com/es/en/jaen/>

1.4.2. Uber Eats

Esta página parece similar a la anterior. Te pide introducir una dirección y después de eso, te ofrecen los distintos restaurantes de la zona. Una vez haces click en uno de ellos puedes elegir entre los distintos menús.

Las vistas principales son las siguientes:

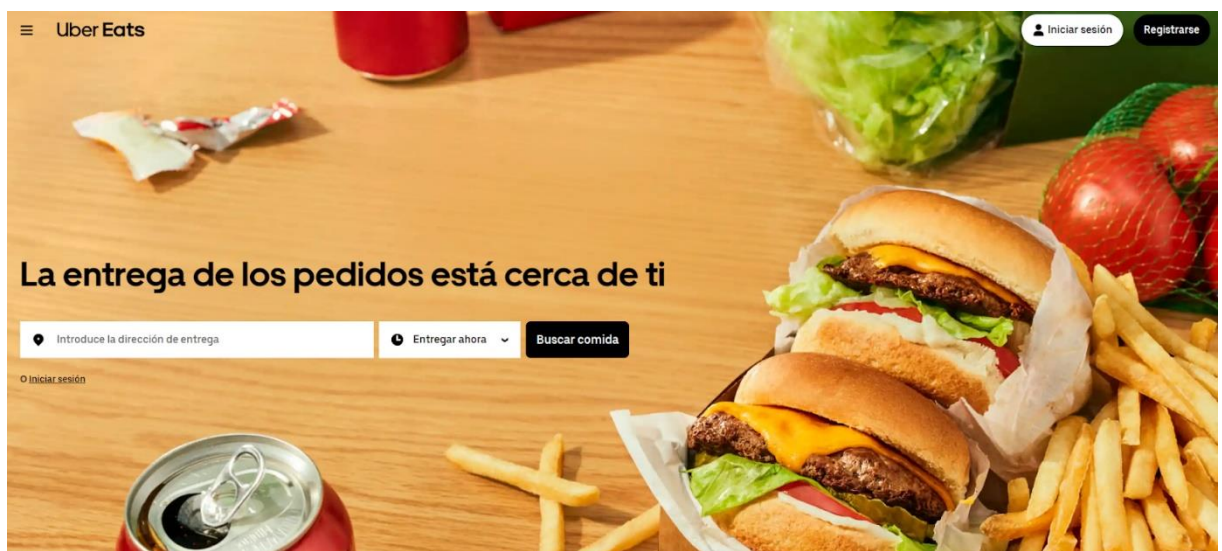


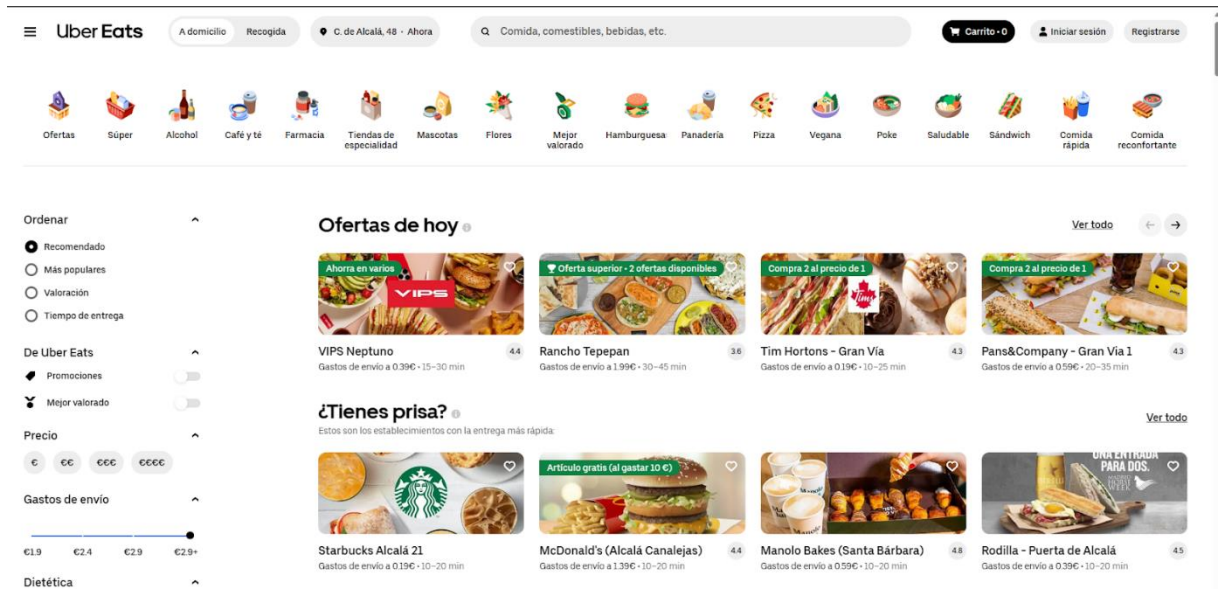
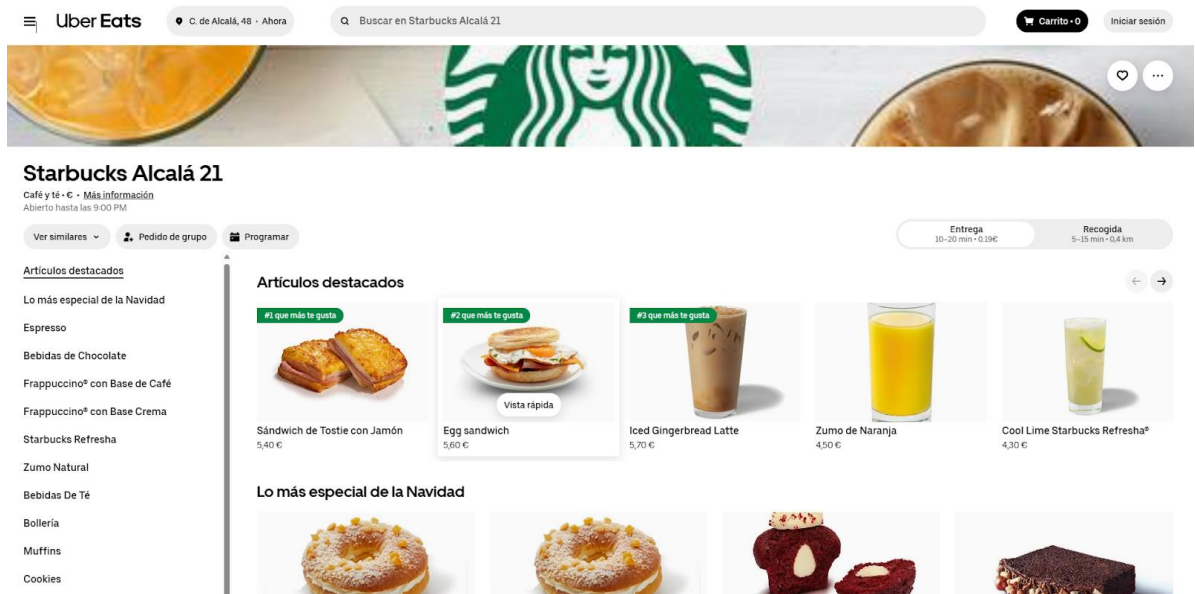
Figura 6: Vista principal de la plataforma web Uber Eats**Figura 7: Vista personalizada de la plataforma web Uber Eats**

Figura 8: Vista de negocio de la plataforma web Uber Eats

Haciendo un segundo análisis general, es igual a la anterior, pero estéticamente en lo personal, me gusta menos. Lo diferente que he visto y me gustaría integrar en mi página web, sería la primera barra de búsqueda, en la que puedes introducir directamente el nombre del restaurante si ya sabes lo que vas buscando. También contiene un recuadro con la opción de programar la hora de entrega, muy conveniente para lo que buscamos en nuestro proyecto.

El link de la plataforma web en cuestión: <https://www.ubereats.com/es>

1.4.3. Just Eats

Ídem a las dos anteriores. Existe la barra buscadora para introducir una dirección o un restaurante y a continuación te muestran los restaurantes de la zona.

Las vistas principales son las siguientes:

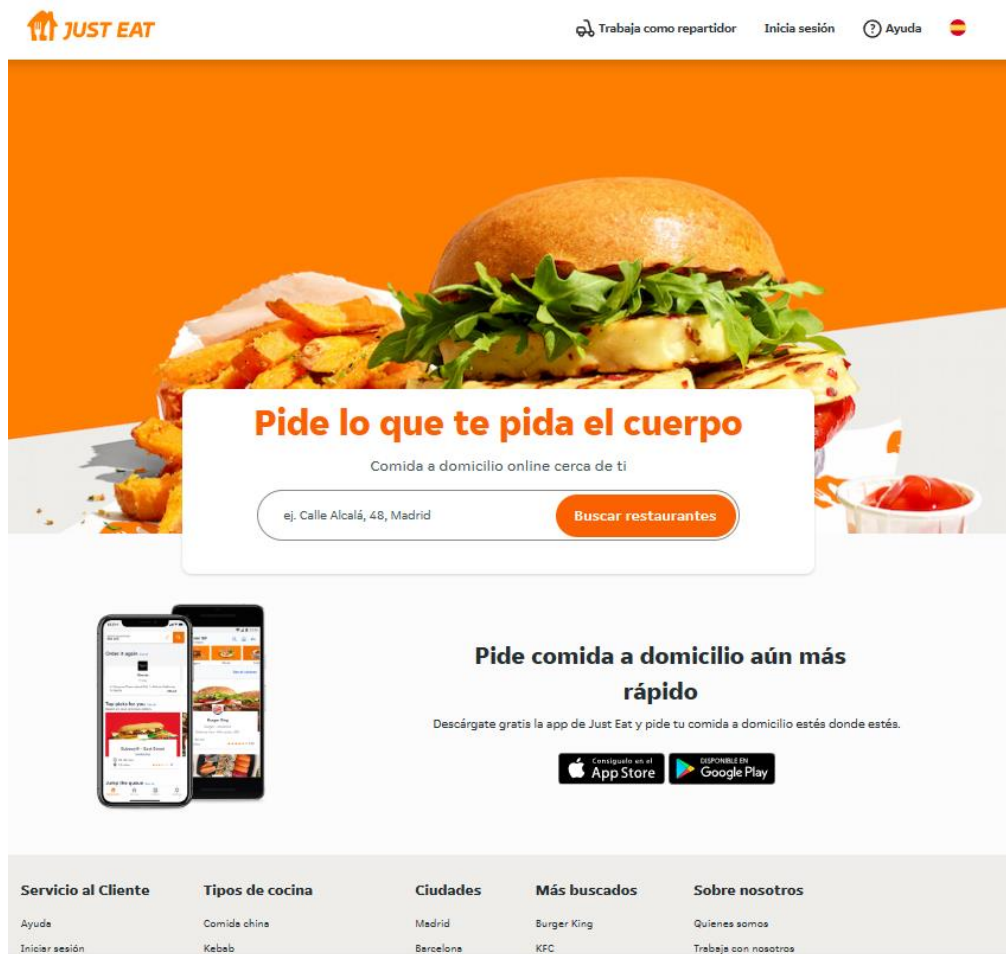


Figura 9: Vista principal de la plataforma web Just Eats

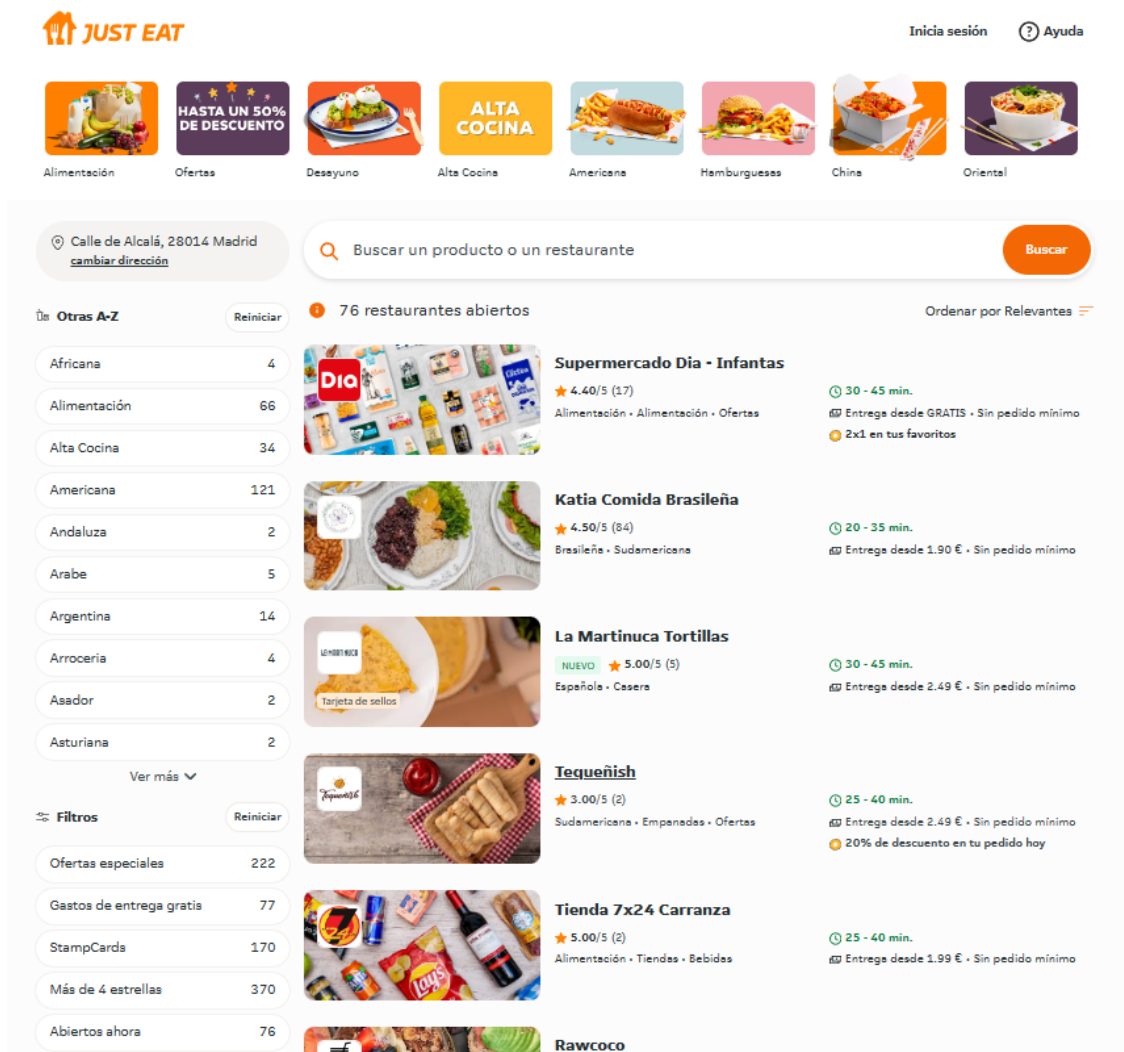


Figura 10: Vista principal de la plataforma web Just Eats

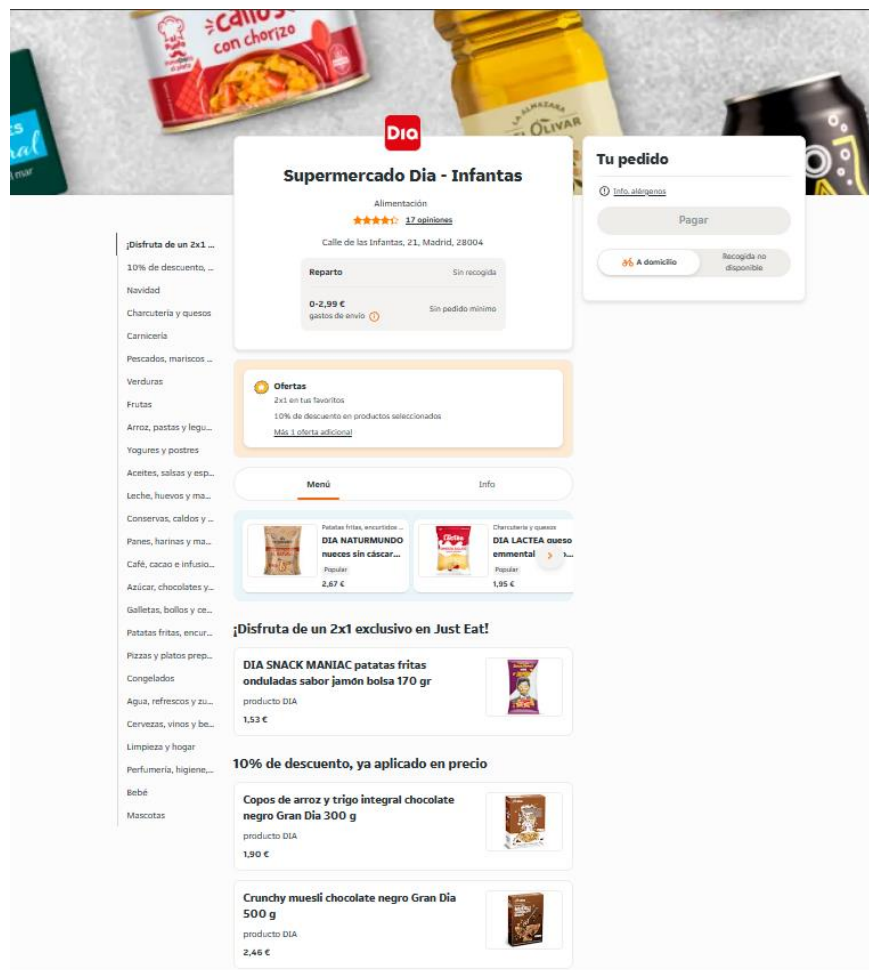


Figura 11: Vista de negocio de la plataforma web Just Eats

En el último análisis, veo claramente que existe un estilo predefinido a la hora de hacer un Marketplace de distintos restaurantes donde los usuarios buscan la comida que ellos quieren. Lo único que varía entre las 3 páginas web son los distintos precios debido a las distintas políticas que hay entre ellos, estilo artístico y presentación de la interfaz y por último se pone mucho énfasis en esa barra principal de búsqueda nada más entrar.

El link de la plataforma web en cuestión: <https://www.just-eat.es/>

Ahora tenemos una visión más clara del panorama general y sabemos qué estilo debe llevar nuestra página, así como los elementos más importantes a resaltar.

1.5. Requisitos y necesidades

Las necesidades de los usuarios serán nuestros factores clave para que nuestra aplicación tenga éxito. Por supuesto, también se tiene en cuenta los restaurantes en nuestra aplicación, pero nuestra búsqueda se basa, principalmente, en aportar información al estudiante y hacerle su etapa universitaria más sencilla. Estas son algunas necesidades a tener en cuenta:

- Para clientes:
 - Buscar, seleccionar restaurantes y comprar menús de forma fácil e intuitiva.
 - Obtener información detallada del valor nutricional del producto.
 - Realizar una búsqueda eficiente por aporte nutricional en específico.
 - Disfrutar del sistema de recomendación basado en las preferencias del usuario.
 - Acceder a promociones exclusivas y ofertas especiales de los restaurantes asociados.
- Para los restaurantes:
 - Registro y gestión sencilla de su establecimiento y productos.
 - Proporcionar información detallada sobre el valor nutricional de sus productos.
 - Acceso a herramientas eficientes para gestionar y actualizar la información nutricional.

- Participar en un sistema de recomendación que destaque sus establecimientos basándose en las preferencias de los usuarios.
- Facilidad para ajustar la disponibilidad de productos y gestionar pedidos de manera eficiente.

1.6. Hipótesis y restricciones

El trabajo de fin de grado consta de 12 créditos, por lo que la duración del mismo, basado en otras materias y en base a las horas lectivas de la universidad, es de 300 horas. En estas horas debe estar incluido todo el proyecto, desde diseño, desarrollo e implementación hasta las pruebas exhaustivas de que nuestro proyecto sea consistente, seguro y funcional para cualquier usuario.

La duración ideal será de unos 4 meses, como un periodo lectivo de la universidad, aunque la idea es acabarlo en mucho menos. Siendo la idea dar un mes para el desarrollo de la parte teórica y otro mes para la parte práctica, siendo ampliable a 1 mes y medio o dos meses. Para el desarrollo de la aplicación se intentará usar software libre, aunque para otros casos, como es este mismo con Open Office será inevitable.

1.7. Estudio de alternativas y viabilidad de cada una de las mismas

Para seleccionar los recursos que usaremos, vamos a realizar una investigación para seleccionar las tecnologías más óptimas para nuestro proyecto.

1.7.1. Alternativas de Frontend

- **React.js.** Biblioteca de JavaScript para construir las interfaces de usuario interactivas en nuestra página web. Utiliza componentes reutilizables que facilitan la creación y actualización eficiente de la interfaz. Hemos escogido esta alternativa por su enfoque declarativo y su capacidad para crear aplicaciones de manera fácil y rápida.



Figura 12: Logo React

- **Node.js.** En el contexto Frontend, es una herramienta de desarrollo que facilita la gestión de dependencias y facilita su automatización. En particular, en nuestro proyecto web hemos usado npm (Node Package Manage) o Yarn, dos administradores de paquetes comunes de Node.js. Con ello hemos instalado, actualizado y gestionado las bibliotecas y dependencias necesarias para nuestro proyecto.



Figura 13: Logo Node.js

- **Vue.js.** Framework progresivo de JavaScript que se centra en la creación de interfaces de usuario interactivas de y una sola página (SPA). Vue.js es conocido por su facilidad de aprendizaje y flexibilidad, convirtiéndolo en una opción popular para desarrolladores de plataformas web.



Figura 14: Logo Vue.js

Angular. Framework de desarrollo frontend creado por Google. Está diseñado para la creación de aplicaciones web SPA complejas y escalables.



Figura 15: Logo Angular

Tras analizar detenidamente las distintas alternativas y evaluar nuestros requisitos nos hemos decantado por la tecnología React. Nuestro proyecto será sencillo, será escalable y hemos buscado una tecnología que tenga una amplia comunidad y facilidad de aprendizaje, además de ser código abierto y haber trabajado antes con ella, la hemos considerado como la mejor opción.

1.7.2. Alternativas de Backend

- **Python:** Lenguaje de programación de alto nivel y propósito general, usado ampliamente en aplicaciones web. Se destaca por su sintaxis clara y legible, facilitando nuestro mantenimiento y desarrollo de código. Tiene software gratis y se integra bien en todos los tipos de sistemas.

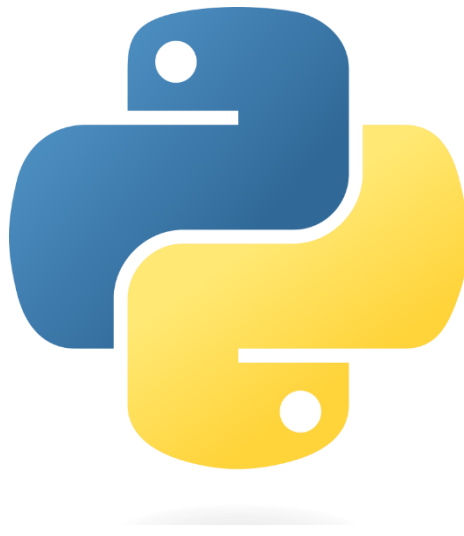


Figura 16: Logo Python

- **Flask:** Marco de desarrollo web para Python. Ligero y fácil de usar, proporcionando las herramientas necesarias para construir nuestro lado servidor. Con él se pueden definir rutas, y manejar las solicitudes HTTP del lado cliente. Debido a que nuestro proyecto es pequeño buscábamos la simplicidad y flexibilidad.



Figura 17: Logo Flask

- **Django:** Marco de desarrollo web para Python que ofrece una estructura sólida para la construcción de aplicaciones web a gran escala. Se pueden construir

aplicaciones web complejas y altamente funcionales gracias a sus características integradas.



Figura 18: Logo Django

- **Express.js:** Marco de desarrollo web para Node.js que se centra en la creación de aplicaciones web simples y escalables. Es minimalista y flexible, ideal para proyectos más pequeños.



Figura 19: Logo Express.js

Para la tecnología backend nos hemos enfocado en los mismos requisitos que para la parte frontend, facilidad de aprendizaje, una amplia comunidad y que cumpla con las restricciones planteadas por nuestro proyecto. Tras explorar las diferentes alternativas, nos hemos decantado por usar Flask junto con Python el cual se integra de manera sencilla con la alternativa escogida Frontend.

- **MySQL WorkBench:** Herramienta gráfica de administración y diseño de bases de datos para MySQL. Con un software libre y gratuito, proporciona una interfaz visual para realizar tareas como modelado de datos, diseño de esquemas entre otras cosas. Con este programa, hemos gestionado nuestra base de datos con muy buenos resultados.

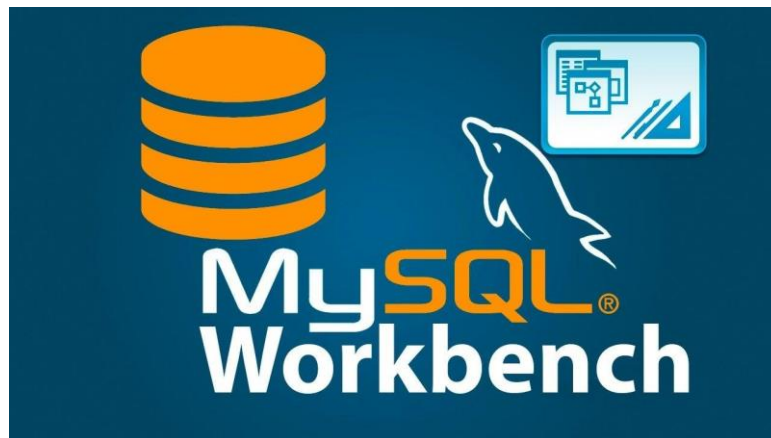


Figura 20: Logo MySQL Workbench

Por último, para una base de datos relacional buscamos programas de código abierto que usaran MySQL. La primera de ellas en aparecer fue PostgreSQL o Microsoft SQL Server, pero investigando un poco más, di con la explicada anteriormente, y por ser más cómoda e intuitiva, hemos elegido MySQL Workbench.

Una vez realizada una investigación sobre las diferentes tecnologías, una breve definición de los términos y un resumen de las seleccionadas:

- **Frontend:** El frontend, o desarrollo del lado del cliente, se refiere a la creación de la parte visual de un sitio web o aplicación. Este es el aspecto con el que los usuarios interactúan directamente en sus navegadores, mediante HTML, el esqueleto de la página, CSS, colores, márgenes y otros estilos y por último JavaScript que proporciona la interactividad. Se han escogido:

- **React:** Enlace al sitio oficial: <https://es.react.dev/>
- **Node.js:** Enlace al sitio oficial: <https://nodejs.org/en>

En resumen, cuando un usuario visita nuestro sitio web, el frontend se encarga de mostrar nuestra información de manera intuitiva y atractiva para que la experiencia del usuario sea funcional y agradable.

- **Backend**, o desarrollo del lado del servidor, está detrás de escena y gestiona las operaciones que el usuario no ve directamente. Se ocupa de procesar todas las solicitudes del usuario, hacer llamadas a bases de datos y realizar otra clase de tareas lógicas. Se han escogido:

- **Python:** Enlace al sitio oficial: <https://es.python.org/>
- **Flask:** Enlace al sitio oficial: <https://flask.palletsprojects.com/en/3.0.x/>

El backend sería como el motor de una aplicación, se encarga de realizar y procesar y manejar las solicitudes hechas por el usuario. La comunicación entre la parte frontend y la backend se realiza a través de solicitudes y respuestas HTTP

- **Base datos relacional:** Se encargará de almacenar y organizar los datos de manera estructurada, permitiendo cualquier tipo de consulta de manera eficiente. La información se organiza en tablas con filas y columnas. Cada fila es una entrada única y cada columna define un tipo de información concreta para el tipo de dato que estamos trabajando. Se ha escogido:

- **MySQL Workbench 8.0:** Enlace al sitio oficial: <https://www.mysql.com/products/workbench/>

Por último, la base de datos sería como el almacén de información de nuestra aplicación. La interacción entre la aplicación (frontend y backend) se gestiona mediante consultas SQL ejecutadas en el backend, desencadenadas por acciones del usuario, como, por ejemplo, registrarse o actualizar su perfil.

1.8. Descripción de la solución propuesta

Como arquitectura de la aplicación web, hemos escogido un modelo cliente-servidor. En este enfoque, el cliente es el navegador del usuario y el servidor puede estar alojado en una máquina dedicada de manera continua, encargándose de procesar solicitudes pertinentes. Mediante esta separación de responsabilidades entre el cliente y el servidor, podremos escalar y mantener la aplicación de manera efectiva.

En el enfoque anteriormente mencionado, las tecnologías usadas en el lado cliente, se han usado las más estándar: HTML para la estructura, CSS para el estilo y JavaScript, concretamente usando React.js para la interactividad. Para el lado servidor hemos usado Flask, un marco de desarrollo web ligero de Python. Nos proporcionará una estructura simple pero eficaz para manejar las solicitudes HTTP por parte del usuario. Con ello tenemos tecnologías fáciles de usar y muy modulares, permitiéndonos un desarrollo ágil y eficiente en los dos lados.

Por último, MySQL WorkBench como herramienta de gestión de nuestra base de datos, con una interfaz gráfica sencilla e intuitiva, la hemos escogido como nuestra mejor opción.

Con esta arquitectura y selección de tecnologías, obtenemos un diseño modular, ágil y eficiente para el desarrollo y mantenimiento de nuestra aplicación web.

1.9. Tecnologías utilizadas

Las tecnologías usadas en nuestro proyecto han sido las siguientes:

- Entorno de desarrollo:

- **PyCharm:** He utilizado PyCharm como mi entorno principal para el desarrollo Python. Complementé este entorno junto con las herramientas de desarrollo del navegador, como el Web Inspector, para analizar y depurar tanto los elementos del lado cliente. Esto me ha permitido ver en detalle la estructura del DOM, monitorear las solicitudes de red y realizar depuración en el código JavaScript.
- **Herramientas:**
 - **GitHub:** Plataforma principal para el control de versiones y colaboración en el desarrollo.
 - **Postman:** Esencial en mi flujo de trabajo para probar y validar las peticiones realizadas por los usuarios y comprobar que toda la información contenida en las peticiones HTTP era correcta.
 - **Microsoft Word:** Procesador de textos para redactar la memoria del proyecto.
 - **Google Docs.:** Herramienta colaborativa en tiempo real de documentos. Usado con el fin de incluir comentarios y sugerencias por parte de compañeros para mejorarlo en tanto fuera posible
 - **Yarn y Npm:** En el desarrollo del lado cliente, hice uso de estas dos herramientas para administrar paquetes y para gestionar dependencias en JavaScript.

1.10. Metodología de desarrollo de software

Como metodología, hemos escogido un enfoque ágil, concebido originalmente para equipos de desarrollo pero que hemos adoptado de manera efectiva a un contexto individual. Este enfoque nos proporciona la flexibilidad necesaria para ajustar los plazos de desarrollo, tomando como base experiencias anteriores y tiempos realistas.

En este marco, damos prioridad a la consecución de metas claras a corto plazo, con entregas iterativas y plazos bien definidos, con el fin de tener una gestión eficiente de retrasos, plazos o dependencias entre las distintas tareas y también, una adaptación ágil a los cambios y retroalimentación constante.

El correcto seguimiento de esta metodología asegura un progreso continuo y fluido del proyecto, respaldado por un seguimiento y control adecuados. Este enfoque no solo se centra en la eficacia del desarrollo, sino que promueve el aprendizaje continuo y mejora iterativa, aspectos fundamentales para un proyecto de investigación y desarrollo como es un Trabajo de Fin de Grado.

1.11. Estimación de tamaño y esfuerzo

Teniendo en cuenta que soy el único desarrollador del proyecto, y en general, de todo el proyecto, y además soy un estudiante, no disponemos de muchos recursos económicos y además tenemos límite de tiempo. Por tanto, más que contar el dinero gastado, contaremos el tiempo empleado en el proyecto.

1.11.1. Presupuesto

RECURSO	UNIDADES	TIEMPO DE USO	COSTE GLOBAL	COSTE PARCIAL
Microsoft Office	Licencia	3 meses	149 €	149 €
Ordenador sobremesa	Licencia	3 meses	1500 €	25€
Pycharm	Licencia	3 meses	0 €	0 €
Teclado	Unidad	3 meses	100 €	2€
Ratón	Unidad	3 meses	50 €	1€

Monitor	Unidad	3 meses	100 €	2€
Ingeniero informático y Programador full stack	Mes	3 meses	30000€	2134€
Coste total				6581€

Tabla 1: Presupuesto de Lunch At Home

Para el software Pycharm, si declaras que eres universitario y que son fines académicos, te dejan usar un año Pycharm por lo que he puesto su costo en mi caso personal, 0 euros.

Para saber el sueldo de un programador full stack con un perfil junior como el mío, he buscado cuál es su sueldo medio en alguna página conocida, tras explorar e investigar las distintas opciones, la página “Talent” me ha parecido la que más información aporta, proporcionando cualquier tipo de salario para cualquier tipo de trabajo en la franja de tiempo que se desee.



Figura 21: Salario programador full stack con un perfil junior

1.11. Diagrama de Gantt

Un diagrama de Gantt es una herramienta de planificación y gestión de proyectos que proporciona una representación visual de la planificación de las actividades. Explicándolo un poco más en detalle, este tipo de diagrama contiene barras horizontales que representan las tareas y actividades a lo largo del tiempo. La longitud de la barra indica la duración de una tarea específica, permitiendo una clara comprensión de la secuencia de actividades e incluso la dependencia entre ellas. También se puede detallar aún más este tipo de diagramas y pueden mostrar hitos claves y eventos importantes o las distintas fases que puede contener el proyecto.

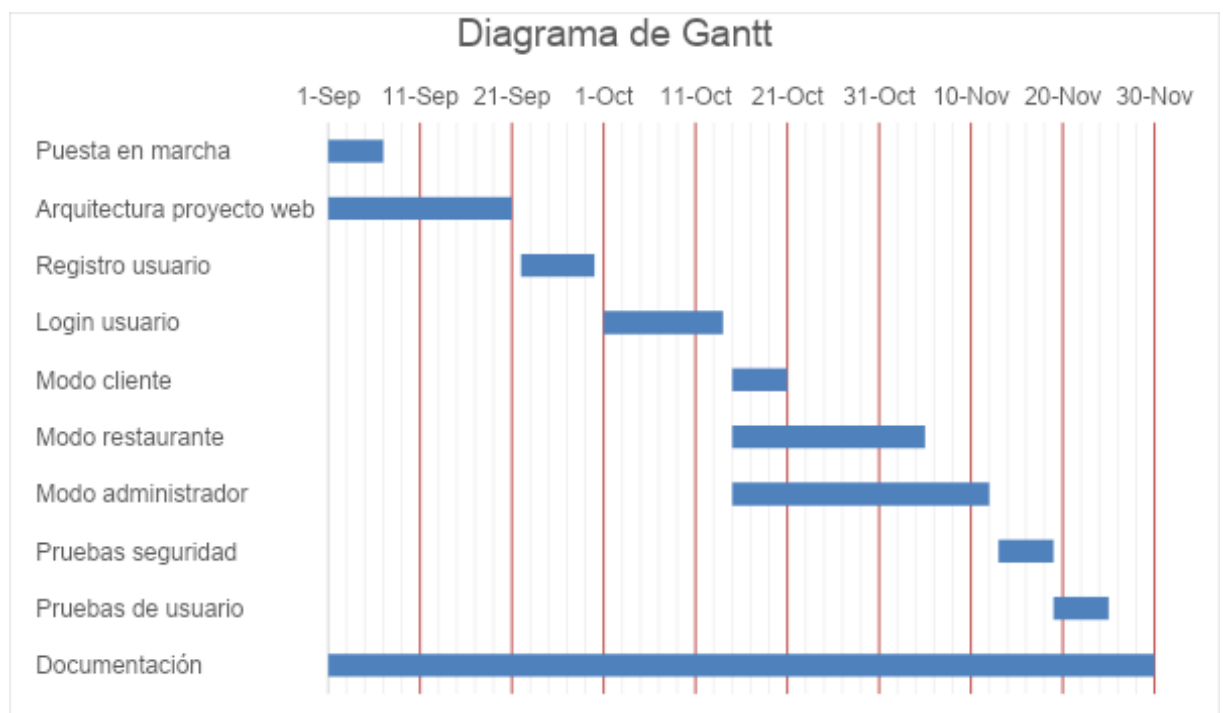
Se ha optado por una planificación temporal inicial en cuanto a horas reales, y lo hemos dividido en pequeñas subtareas que cuentan con una serie de puntos generales a seguir y le hemos asignado puntos de esfuerzo, de cuanto mas o menos creemos que nos va a costar conseguir esos objetivos, como nunca nos hemos embarcado en un proyecto así, a priori vamos a suponer que todas las subtareas cuestan el mismo esfuerzo.

Quedando de la siguiente manera:

Puntos generales	Subtareas	Puntos de historia por subtareas	INICIO	FINAL	Primera iteración						Segunda iteración			Tercera iteración			Total
					01/09	11/09	18/09	25/09	02/10	09/10	16/10	23/10	30/10	06/11	13/11	20/11	
Inicio del proyecto	Análisis del sistema	1	01/09	11/09	1												1
	Diseño de la arquitectura y desarrollo de los componentes	1	01/09	11/09	1												1
	Instalación y configuración de PyCharm	1	01/09	11/09		1											1
	Instalación y configuración de Flask + Python y React	1	01/09	18/09		1											1
	Instalación de SQL WorkBench	1	01/09	18/09			1										1
	Instalación de otros softwares necesarios	1	01/09	18/09			1										1
Cliente	Registrar usuario	1	25/09	02/10				1									1
	Iniciar sesión	1	25/09	02/10				1									1
	Editar usuarios	1	02/10	09/10					1								1
	Eliminar usuarios	1	02/10	09/10					1								1
	Buscar usuario	1	02/10	09/10					1								1
Restaurante	Crear restaurante	1	16/10	23/10							1						1
	Editar restaurante	1	23/10	30/10								1					1
	Eliminar restaurante	1	23/10	30/10								1					1
	Buscar restaurante	1	30/10	06/11									1				1
Menú	Crear menú	1	06/11	13/11										1			1
	Editar menú	1	06/11	13/11										1			1
	Eliminar menú	1	06/11	13/11										1			1
	Buscar menú	1	13/11	20/11											1		1
Cierre del proyecto	Medidas de seguridad datos de usuario	1	20/11	27/11												1	1
	Medida de seguridad sesión del usuario	1	20/11	27/11												1	1
	Medidas de seguridad reglas ACL	1	20/11	27/11												1	1
	Normas de estilo	1	20/11	27/11												1	1
Resumen	Puntos totales	23	01/09	27/11	2	2	2	2	3	0	1	2	1	3	1	3	23
	Puntos por días/semanas		01/09	27/11	0.28	0.28	0.28	0.28	0.14	0	0.14	0.28	0.14	0.42	0.14	0.42	0.23
																	1.84

Tabla 2: Planificación temporal Lunch At Home

Tener en gran detalle la planificación, se traduce en la mejora de la eficiencia de la gestión del proyecto, pudiendo identificar posibles cuellos de botella y gestionar mejor los recursos. A continuación, un gráfico donde se visualiza mejor los puntos generales y el tiempo estimado.

**Figura 22: Diagrama de Gantt de Lunch At Home**

2. Inicio del proyecto

Una vez hemos dejado claro las bases de nuestro proyecto, toca poner en marcha el siguiente paso, una tarea fundamental, la definición de un conjunto de historias de usuario iniciales, o lo que es lo mismo, los requisitos iniciales de nuestra aplicación. El propósito es lograr un desarrollo eficiente mediante la utilización de la metodología incremental, lo cual, junto con otras metodologías anteriormente descritas, nos ayudará en nuestro desarrollo gracias a los elementos definidos en esta sección.

2.1. Requisitos iniciales

Los requisitos iniciales de un proyecto son fundamentales para establecer una base sólida y comprender la dirección que tomará el desarrollo. Tenemos dos tipos:

- **Requisitos funcionales:** Describen las funciones y características específicas que debe tener un proyecto para satisfacer las necesidades del usuario y cumplir con los objetivos del mismo. Se centran en lo que se debe hacer.
- **Requisitos no funcionales:** No están relacionados directamente con las funciones específicas como en el apartado anterior, si no cómo debería comportarse un sistema o producto en términos de características más amplias como rendimiento, usabilidad o seguridad.

2.1.1. Requisitos funcionales

- **Registro de usuarios.** Los usuarios podrán crear cuentas en la plataforma proporcionando información básica como nombre, correo electrónico y contraseña de manera segura.

- **Búsqueda de productos.** Los usuarios podrán buscar productos disponibles en la plataforma, dando resultados de búsqueda relevantes y filtrables.
- **Detalles del menú.** Los usuarios podrán ver la información del menú de cada menú de cada restaurante. Esta información incluirá precios, descripción, información nutricional e incluso imágenes de los mismos.
- **Registro de restaurantes.** Los usuarios que así lo quieran también podrán crear su propio restaurante dentro de nuestra plataforma. Parecido al registro de usuarios, proporcionando información básica de su alojamiento, pronto podrán disfrutar de nuestra plataforma.
- **Gestión de restaurante.** Los restaurantes tendrán su propia gestión en la plataforma. Podrán poner, actualizar y eliminar tanto menús como información de los mismos.

2.1.2. Requisitos no funcionales

- **Seguridad.** Implementación de protocolos de seguridad para proteger la información del usuario y garantizar la privacidad de los datos almacenados.
- **Fiabilidad.** Garantizar la estabilidad del sistema para evitar caídas inesperadas, realizando pruebas de robustez para manejar situaciones inesperadas.
- **Disponibilidad.** Mantener la plataforma disponible la mayor parte del tiempo, programando mantenimientos y actualizaciones.
- **Adaptabilidad.** Garantizar que la plataforma sea accesible desde cualquier tipo de pantalla, es decir, que se visualice correctamente y desde cualquier tipo de navegador.
- **Rendimiento.** Optimizar la velocidad de carga de la plataforma, búsqueda de productos y manejar el tráfico de usuarios de manera eficiente.

- **Escalabilidad.** Diseñar la arquitectura para escalar fácilmente ante un aumento de la cantidad de usuarios en nuestro sistema.
- **Documentación.** Proporcionar documentación clara y completa sobre el funcionamiento de nuestra plataforma, para desarrolladores, administradores y usuarios finales.

Con estos requisitos, hemos establecido las funcionalidades clave y los criterios de rendimiento de nuestra aplicación web. Por supuesto, mediante auditorías y en el desarrollo del proyecto, se incluirán requisitos que se consideren importantes, aunque no lo hayamos considerado inicialmente.

2.2. Casos de uso

En esta sección, vamos a entrar más en detalle en los diferentes casos de uso del proyecto, la descripción detallada de las acciones del usuario. Para ello vamos a seguir el Marco de Desarrollo de la junta de Andalucía (MADEJA) <https://www.juntadeandalucia.es/servicios/madeja/>

CU-01	<i>Registrar usuario</i>		
Versión	<i>1.0 (01/09/2023)</i>		
Dependencias	• <i>El usuario debe tener una cuenta creada.</i>		
Precondición	<i>El usuario accede a nuestra página web y hace click en el botón de registrarse</i>		
Descripción	Permite al usuario crear una cuenta en nuestra aplicación y acceder a todas las funcionalidades.		
Secuencia Normal	Paso	Acción	
	1	El usuario accede a nuestra aplicación e intenta acceder a la misma.	
	1.1	<i>El usuario accede directamente a al botón de registro</i>	
	1.2	El usuario accede a iniciar sesión, pero al no tener cuenta, pincha en “Regístrate aquí”	

	2	El sistema muestra un formulario de registro
	3	El usuario completa los campos requeridos del formulario con la información que desee proporcionar.
	4	El usuario hace click en el botón “Registrarse”
	5	El sistema valida la información y comprueba la lógica pertinente.
	6	Si la información es correcta, el sistema crea una cuenta para el usuario y almacena su información.
	7	El sistema muestra un mensaje de confirmación y lo redirige a la pantalla de inicio de sesión.
Postcondición	<i>El usuario ha creado una cuenta exitosamente y puede iniciar sesión usando sus credenciales.</i>	
Excepciones	Paso	Acción
	3	El usuario no completa correctamente los campos requeridos del formulario
	E.1	<i>El sistema muestra un mensaje de error indicando los campos obligatorios que faltan por completar</i>
	E.2	<i>El sistema redirige al usuario al formulario de registro para que complete los campos faltantes.</i>
	5	<i>El correo electrónico proporcionado ya está registrado, se muestra un mensaje de error indicando que ya está en uso.</i>
	E.1	<i>El sistema verifica la existencia del correo electrónico en la base de datos.</i>
	E.2	<i>El sistema muestra un mensaje de error indicando que el correo electrónico ya está en uso y sugiere al usuario iniciar sesión con ese correo en lugar de crear una cuenta nueva.</i>

Tabla 3: Caso de uso – Registrar usuario

CU-02	<i>Iniciar sesión</i>
Versión	<i>1.0 (01/09/2023)</i>
Dependencias	<i>El usuario debe tener una cuenta en el sistema</i>
Precondición	<i>El usuario hace click en iniciar sesión o es redirigido desde el botón “Registrarse”</i>
Descripción	Permite al usuario iniciar sesión en la aplicación utilizando los datos con los que se ha registrado previamente.

Secuencia Normal	Paso	Acción
	1	<i>El usuario introduce su información en los campos correspondientes.</i>
	2	El usuario presiona el botón de “Iniciar sesión”
	3	El sistema valida los datos ingresados.
	4	<i>El sistema autentica al usuario y lo redirige a la página principal de la aplicación.</i>
	5	<i>El sistema valida la información y verifica si el correo electrónico y verifica si el correo electrónico ya está registrado en el sistema</i>
	4	<i>El sistema autentica al usuario y lo redirige a la página principal de la aplicación.</i>
	5	El sistema valida la información y verifica si el correo electrónico ya está registrado en el sistema
Postcondición	<i>El usuario crea correctamente una cuenta en nuestro sistema y puede iniciar sesión</i>	
Excepciones	Paso	Acción
	3.1	Si la dirección del correo electrónico no está registrada en el sistema
	E.1	<i>El sistema muestra un mensaje de error indicando que la dirección de correo electrónico no está registrada.</i>
	E.2	<i>El sistema ofrece la opción de registrarse para crear una nueva cuenta.</i>
	3.2	Si la contraseña dada no coincide con la registrada en nuestra base de datos.
	E.1	<i>El sistema muestra un mensaje de error indicando que la contraseña es incorrecta y vuelve a ofrecer el formulario de inicio de sesión.</i>
	E.2	<i>El sistema ofrece al usuario la opción de restablecer su contraseña en caso de haberla olvidado.</i>

Tabla 4: Caso de uso – Iniciar sesión

CU-03	<i>Buscar producto</i>
Versión	1.0 (01/09/2023)
Dependencias	• <i>Navegador web</i>
Precondición	<i>El usuario accede a nuestra página web</i>
Descripción	Permite al usuario buscar un restaurante en específico en nuestra aplicación web y ver sus productos.

Secuencia Normal	Paso	Acción
	1	<i>El usuario hace click en la barra de búsqueda de nuestra aplicación y usa unas palabras clave</i>
	2	El usuario presiona el botón “Buscar”
	3	El sistema realiza una búsqueda con las palabras clave usadas en nuestra base de datos.
	4	<i>Se muestra un listado con los diferentes resultados encontrados por el sistema.</i>
	5	El usuario ahora puede ver los resultados diferentes proporcionados por su búsqueda en nuestro sistema e interactuar con ellos.
Postcondición	<i>El usuario ha encontrado el restaurante deseado y puede ver todo el catálogo ofrecido por el mismo.</i>	
Excepciones	Paso	Acción
	3	Si no se encuentran resultados que coinciden con las palabras clave introducidas por el usuario.
	E.1	<i>El sistema no devuelve valores y muestra un mensaje indicando que no se encontraron resultados</i>
	E.2	<i>El usuario utiliza la barra de búsqueda para buscar con otras palabras clave.</i>

Tabla 5: Caso de uso – Buscar un producto

CU-04	<i>Gestionar restaurante</i>	
Versión	<i>1.0 (01/09/2023)</i>	
Dependencias	<i>El usuario debe haber iniciado sesión como administrador o restaurante.</i>	
Precondición	<i>Debe haber tener uno de los roles designados y haber accedido la sección de gestión de restaurante.</i>	
Descripción	Permite al usuario gestionar sus restaurantes y modificar sus datos pertinentes.	
Secuencia Normal	Paso	Acción
	1	<i>El usuario se dirige al inicio de la vista principal de su negocio</i>
	2	El sistema ya muestra cuales son todos los restaurantes asociados, pudiendo interactuar con ellos
	3	El usuario realiza sus tareas, como modificar la información del restaurante, así como

		añadir o quitar menús.
	4	<i>El sistema responderá ante estas modificaciones, actualizando la información mostrada en nuestra aplicación.</i>
Postcondición	<i>El usuario ha gestionado el restaurante de manera exitosa, actualizando el estado.</i>	
Excepciones	Paso	Acción
	3	Si el usuario no proporciona correctamente información a la hora de realizar la tarea conveniente para su restaurante:
	E.1	<i>El sistema muestra un mensaje de error indicando que se deben completar los campos obligatorios.</i>
	E.2	<i>El usuario debe corregir los campos requeridos y volver a intentar rellenar el formulario.</i>

Tabla 6: Caso de uso – Gestionar un restaurante

CU-05	<i>Gestión de administrador</i>	
Versión	<i>1.0 (01/09/2023)</i>	
Dependencias	<i>Rol de administrador proporcionado por desarrolladores</i>	
Precondición	<i>El usuario accede con el botón de inicio de sesión con sus credenciales</i>	
Descripción	Permite al usuario una vista completa de toda la aplicación pudiendo realizar cualquier acción que desee.	
Secuencia Normal	Paso	Acción
	1	<i>El usuario debe dirigirse a la pestaña deseada, según lo que busque gestionar: usuarios o restaurantes</i>
	2	El sistema muestra una lista con todos los usuarios o restaurantes.
	3	El usuario puede tanto agregar o eliminar usuarios como restaurantes, así como modificar la información de los mismos.
	4	El sistema responderá ante estas modificaciones, actualizando la información guardada y mostrada en nuestra aplicación web.
Postcondición	<i>El usuario ha realizado sus tareas pertinentes en el sistema de manera correcta y eficiente.</i>	
Excepciones	Paso	Acción
	2	Si no existen usuarios o restaurantes

		E.1	Mostrará un mensaje con "No existen usuarios o restaurantes"
		E.2	El usuario vuelve a la vista anterior

Tabla 5: Caso de uso – Gestión de administrador

CU-06	Realizar comprar			
Versión	1.0 (01/09/2023)			
Dependencias	El usuario debe haber iniciado sesión en nuestra aplicación y haber seleccionado varios productos.			
Precondición	El usuario ha seleccionado uno o varios productos y ha iniciado el proceso de compra.			
Descripción	Permite al usuario pedir productos de uno o varios restaurantes que él haya seleccionado.			
Secuencia Normal	Paso	Acción		
	1	El usuario, en el producto deseado, presiona el botón: “Añadir a la cesta”		
	2	El usuario presiona el botón: “Ver cesta”		
	3	El sistema muestra un resumen de la compra, con toda la información necesaria para el usuario y la posible opción de poder cambiarla.		
	4	El usuario presiona el botón “Confirmar y Pagar”		
Postcondición	El usuario ha completado exitosamente el proceso de compra y es confirmado por el sistema.			
Excepciones	Paso	Acción		
	3.1	Si el usuario no tiene ningún producto en la cesta		
		E.1	La cesta sale en blanco con un mensaje de “La cesta está vacía”	
		E.2	El usuario vuelve a un paso anterior o a la página principal	
	3.2	El usuario no ha proporcionado dirección de envío		
		E.1	El sistema muestra un mensaje de error indicando que la información de envío es obligatoria.	
		E.2	El usuario vuelve al paso anterior.	

Tabla 7: Caso de uso – Realizar compra

2.3. Análisis y diseño del sistema

2.3.1. Análisis

Para nuestra plataforma, van a intervenir entidades relacionadas con un Marketplace de restaurantes y comida. Hemos trasladado estas entidades a diferentes clases del proyecto.

1. Usuario

Representa a los usuarios de nuestro sistema en su totalidad. Es donde centraremos todo nuestro esfuerzo. La información a almacenar será la siguiente:

- userEmail (Email de usuario)
- userPassword (Contraseña del usuario)
- userRole (Rol del usuario)
- userPreference (Preferencias del usuario)

Los distintos roles de los usuarios serán los siguientes en la plataforma: Cliente, Restaurante y Administrador. Según el rol de usuario, los usuarios podrán realizar distintas actividades dentro de la plataforma.

En cuanto a las preferencias de usuario, un usuario podrá elegir qué tipo de comida quiere según su preferencia, para que se muestren primero unos restaurantes antes que otros, pudiendo elegir entre: Kcal, Protein, Carbs y Fats.

2. Restaurante

Representan los negocios donde los usuarios comprarán productos. Un restaurante podrá mostrar cualquier tipo de información sobre sus menús y los usuarios podrán hacer cualquier tipo de búsqueda al respecto. La información a almacenar será la siguiente:

- userEmail (Email del usuario, un usuario puede tener varios restaurantes)
- restaurantID (Identificador del restaurante)
- restaurantName (Nombre del restaurante)
- restaurantImage (Imagen del restaurante)
- restaurantDescription (Descripción del restaurante)
- restaurantType (Tipo de restaurante)

Un restaurante puede enfocarse en un tipo de comida en concreto, para poder ser destacado a los usuarios con ese mismo tipo de preferencias. Las distintas opciones que se pueden elegir son las siguientes: Kcal, Protein, Carbs y Fats.

Con esto hemos creado un sistema de recomendación simple para un usuario según su preferencia.

3. Menú

Cada restaurante, va a tener una serie de menús, donde tendrán una información detallada del mismo y su información nutricional. Se podrá filtrar por el valor nutricional que se necesite. Esta entidad no tendría sentido si no existiera la entidad restaurante, ya que los menús no pueden existir por si solos. La información a almacenar será la siguiente:

- menuID (Identificador del menú)

- menuName (Nombre del menú)
- restaurantID (Identificador del restaurante, necesitamos saber a qué restaurante pertenece el menú)
- menuPrice (Precio del menú)
- menuImage (Imagen del menú)
- menuKcal (Calorías del menú)
- menuProtein (Proteína del menú)
- menuCarbs (Carbohidratos del menú)
- menuFats (Azúcares del menú)

2.4. Diagrama de clases

Para una vista general hemos realizado un diagrama de clase. Estos diagramas son una representación gráfica de la estructura y las relaciones entre las distintas clases de un sistema software, proporcionando una vista estructurada y organizada de las distintas entidades.

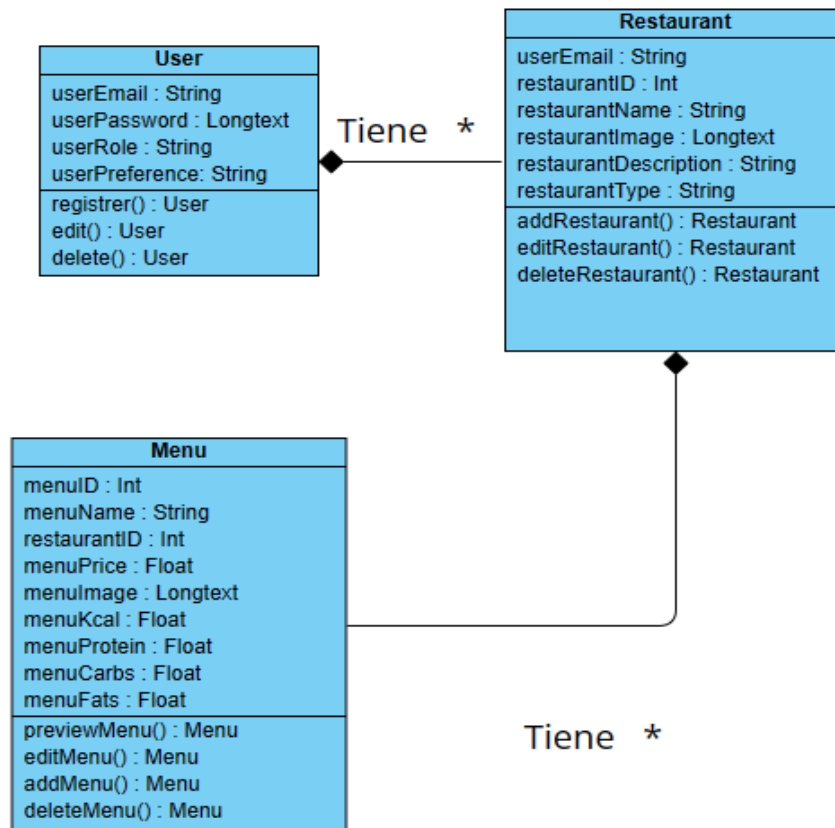


Figura 23: Diagrama de clases de Lunch At Home

2.5. Diseño arquitectónico preliminar

En esta fase inicial del proyecto, vamos a establecer la arquitectura de la aplicación, definiendo la estructura general y los principios clave para guiar el desarrollo. Para comenzar a trabajar, es importante establecer la base de nuestra aplicación, los lenguajes y los marcos de trabajo generales, a fin de tener un criterio arquitectónico general.

En esta etapa, vamos a delinear la arquitectura de nuestra aplicación web y establecer los principios que se seguirán durante el desarrollo de la misma.

2.5.1. Arquitectura del sistema

La arquitectura de nuestra aplicación está compuesta por capas, ya mencionadas anteriormente. Tenemos, por un lado, el lado cliente, que se encargará de mostrar la información, interactuar con el usuario, la lógica visual entre otras cosas y tenemos el lado servidor, encargado del almacenamiento de los datos de usuarios, gestión de sus peticiones y todo lo relacionado con la lógica de datos, pero de ellas hablaremos más en profundidad más adelante.

Nuestra aplicación será de una sola página, conocida como Single Page Application (SPA), un enfoque de diseño web donde toda la interacción del usuario ocurre en una única página. En lugar de cargar páginas completamente nuevas desde el servidor del usuario, una SPA carga solo los recursos necesarios y actualiza la interfaz del usuario de manera dinámica. Este enfoque mejora la experiencia del usuario al proporcionar transiciones rápidas y suaves entre las diferentes secciones de la aplicación sin, como hemos dicho antes, tener que cargar nuevas páginas desde el lado servidor, ahorrando tiempo y mejorando la experiencia del usuario.

Vamos a ver más en profundidad las capas de nuestra aplicación:

- **Capa de presentación (Frontend):** Siendo la que nos hemos referido hasta ahora como “lado cliente” esta capa se encarga de presentar la interfaz del usuario al usuario final. Se implementa la lógica de presentación usando React.js para lograr interfaces de usuario interactivas. Esta capa de presentación se comunica con el “lado servidor” a través de una API RESTful para solicitar y enviar datos.

- **Capa de Lógica de Negocio (Backend):** Identificada como el “lado servidor” esta capa alberga la lógica de negocio y procesamiento de datos. Desarrollada usando el framework Flask de Python, esta capa responde a las solicitudes del lado cliente y realiza las operaciones de autenticación, autorización y gestión de datos. Aquí, se establece la conexión con la base de datos, la siguiente capa, en nuestra arquitectura.
- **Capa de Almacenamiento (Base de datos):** Esta capa maneja la persistencia y recuperación de datos. Usamos MySQL como sistema de gestión de bases de datos relacional, garantizando la integridad y disponibilidad de los datos. La “Capa de Lógica de Negocio” interactúa con ésta para realizar operaciones de lectura y escritura.

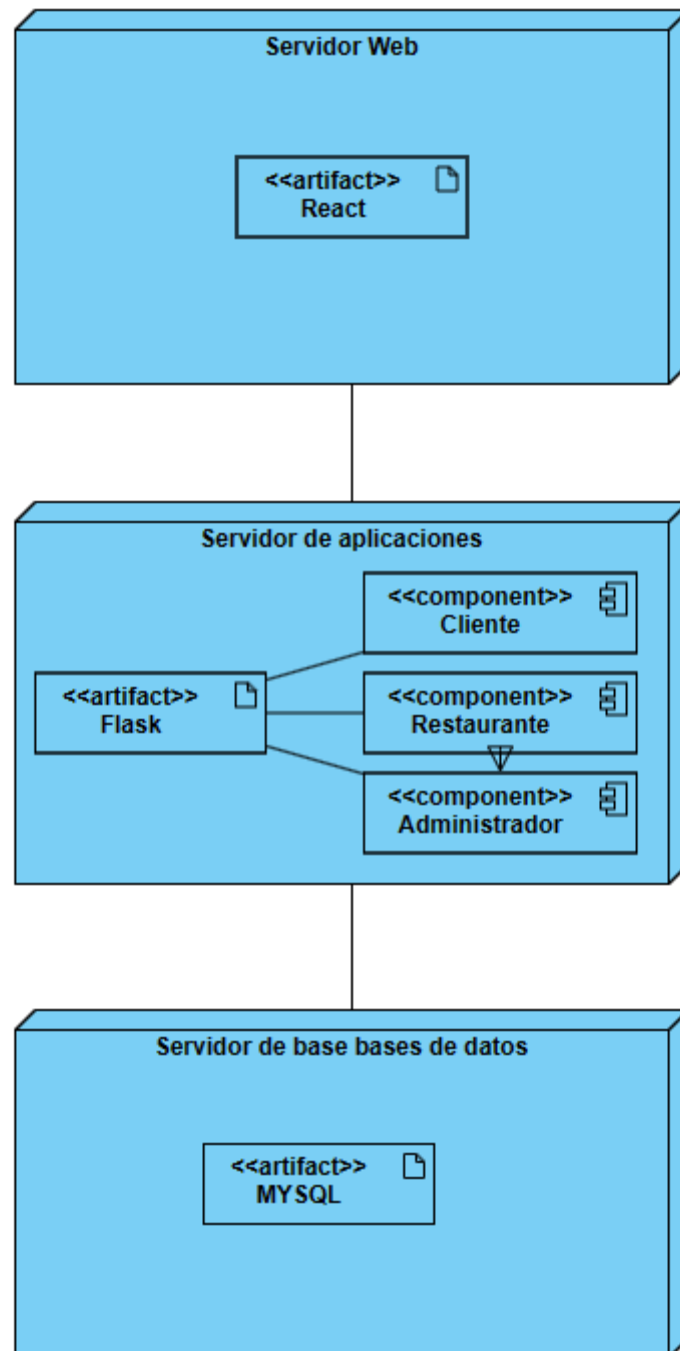


Figura 24: Arquitectura de Lunch At Home

Esta arquitectura proporciona varios beneficios clave, entre los cuales vamos a destacar:

- **Modularidad:** Las SPAs se diseñan normalmente en módulos independientes, donde cada sección de la aplicación se puede considerar un módulo separado. Al desarrollarse estos módulos de manera aislada, facilita la colaboración y gestión del código.
- **Escalabilidad:** La arquitectura SPA maneja eficientemente un gran número de vistas y funcionalidades sin recargar la página, todo debido a que se cargan los recursos necesarios.
- **Reutilización de código:** El modularidad hablado anteriormente, tiene un segundo beneficio, y es su reutilización en diferentes partes de la aplicación. Con ello reducimos la redundancia y mejoramos la consistencia.
- **Mantenibilidad:** Esta separación entre módulos crea la capacidad para gestionar el código y facilita las tareas de mantenimiento, debido a que las actualizaciones y correcciones pueden realizarse sin afectar a las otras partes de la aplicación.
- **Rendimiento mejorado:** El efecto inmediato es una carga inicial más rápida. Para llevar este apartado a su máxima expresión, las SPAs pueden aprovechar técnicas como la carga diferida y el almacenamiento en caché para mejorar aún más su rendimiento. Para nuestro proyecto, se han propuesto implementarlas a largo plazo.
- **Experiencia del usuario continua:** Al no necesitar recargar la página cada vez que hay una nueva vista, los usuarios tienen una experiencia más fluida y continua, similar a las aplicaciones de escritorio.

2.5.2. Cliente

La parte servidor web aporta una interfaz de usuario que se encuentra ajustada a la función que nos ofrece el servidor. Esta información se mostrará al usuario de forma modular, dependiendo del usuario y de los requisitos que este necesite.

Para el rol de Restaurante

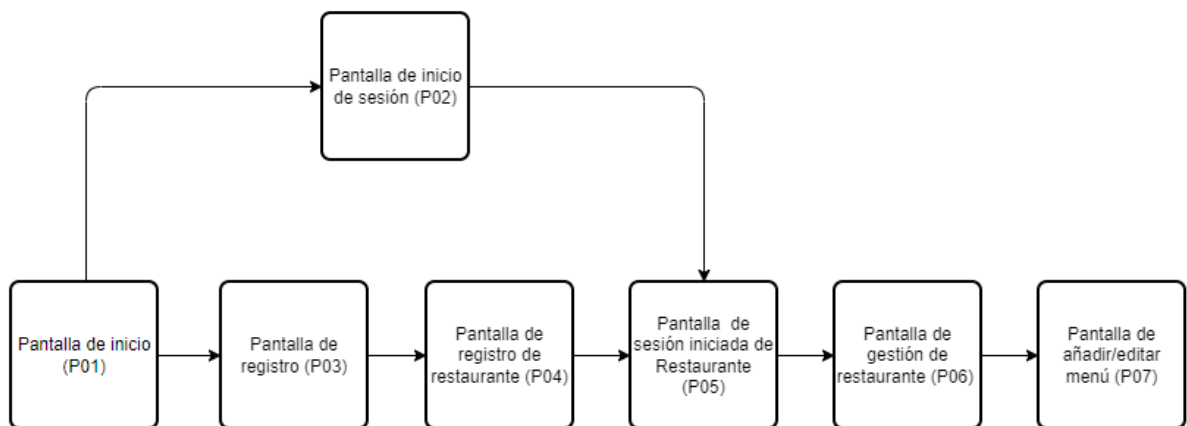


Figura 25: Diagrama de flujo del rol Restaurante

Para el rol de administrador

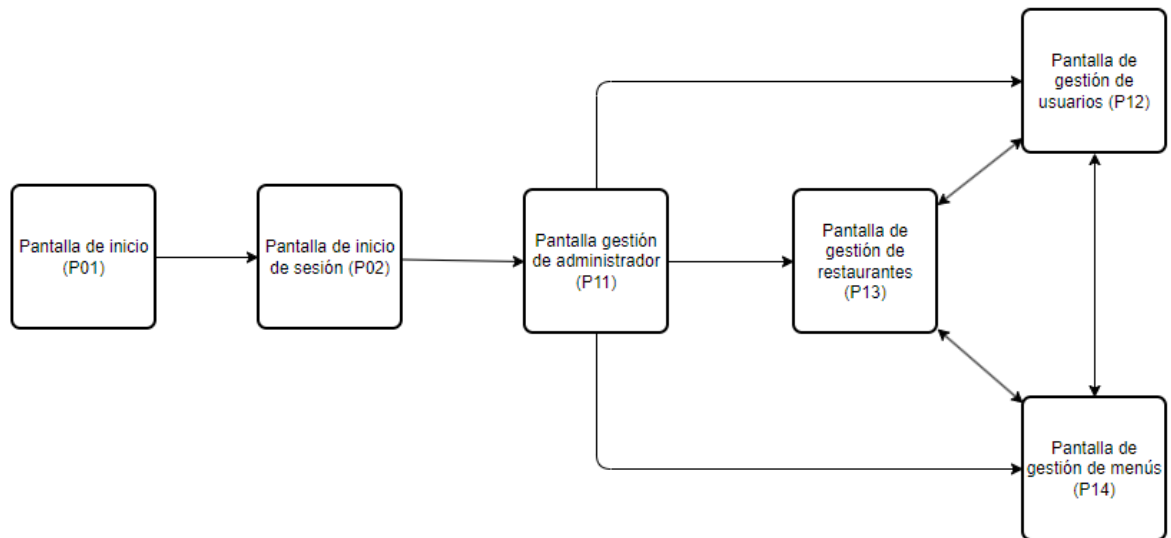


Figura 26: Diagrama de flujo del rol administrador

Para el rol de Cliente

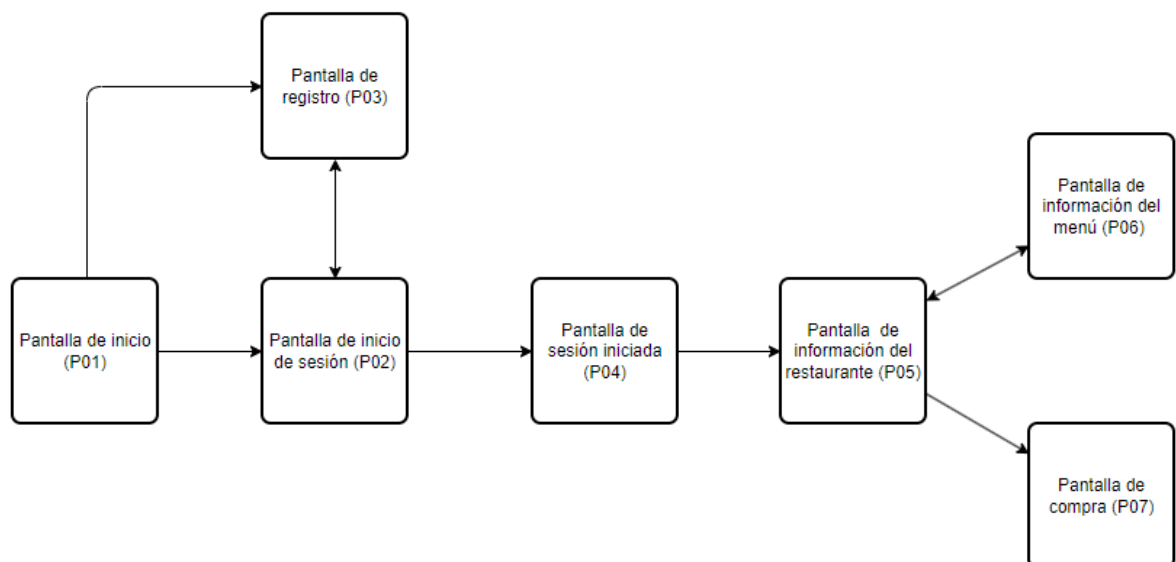


Figura 27: Diagrama de flujo del rol cliente

Diseño de pantallas

En este subapartado, vamos a mostrar un diseño de pantallas de nuestra aplicación. Para visualizar mejor nuestra aplicación, se muestra un esquema general de navegación de pantallas.

- **Pantalla de inicio (P01):** La página web tendrá una cabecera con el logo de la aplicación y un par de botones para iniciar sesión y realizar el registro de usuarios. El cuerpo de la página contendrá la información principal del portal antes de iniciar sesión, como los restaurantes más populares o la barra de búsqueda. El pie de página tendrá información sobre el equipo desarrollador de la aplicación.

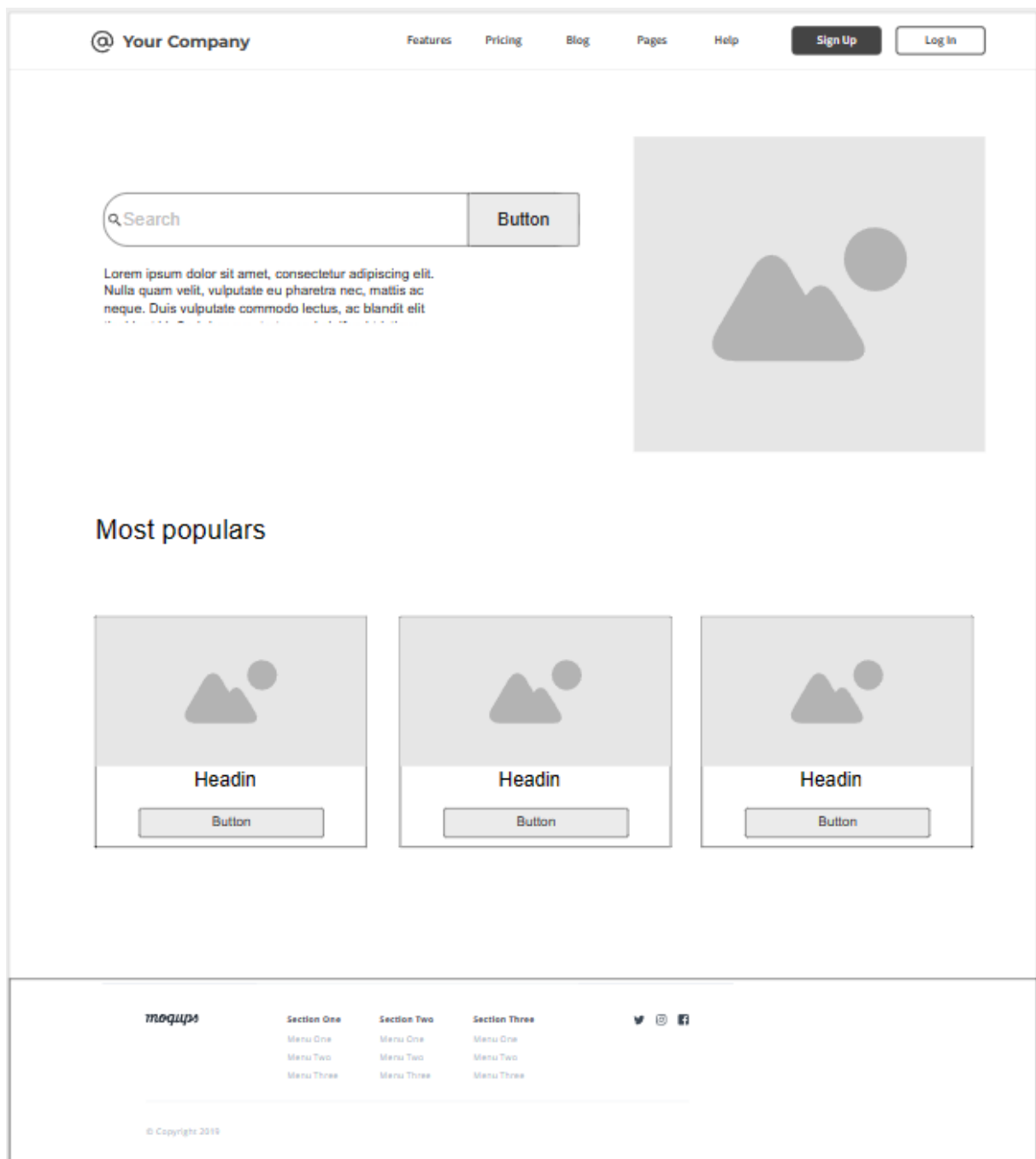


Figura 28: Diseño de pantalla de inicio

- **Pantalla de inicio de sesión (P02):** Al pulsar el botón mencionado anteriormente mencionado en la cabecera, con la etiqueta de “Iniciar sesión”, accederemos a la pantalla principal de inicio de sesión.

Podremos iniciar sesión a partir de nuestro correo y contraseña que hayamos registrado anteriormente. Tendremos también un link en caso de que no nos hayamos registrado.

The image shows a web page mockup for a login form. At the top, there is a navigation bar with the company logo 'Your Company' on the left and links for 'Features', 'Pricing', 'Blog', 'Pages', 'Help', 'Sign Up', and 'Log In' on the right. The main content area has a large heading 'Heading' centered. Below the heading is a login form box containing two input fields: 'Usuario' and 'Contraseña'. Below these fields are a checkbox labeled 'Checkbox' and a link labeled 'Link'. At the bottom of the form box is a button labeled 'Button'. The footer of the page includes the 'mockups' logo, three columns of placeholder text under the headings 'Section One', 'Section Two', and 'Section Three', social media icons for Twitter, Instagram, and Facebook, and a copyright notice '© Copyright 2019'.

Figura 29: Diseño del form para iniciar sesión

- **Pantalla de registro (P03):** Con el segundo botón de la cabecera, accederemos a la pantalla de registro que contendrá los siguientes campos para registrarse. Una vez hayamos completado todos los campos con la información pertinente, podremos confirmarlos con el botón de confirmar.

The image shows a web page layout for a registration form. At the top is a header with the company logo '@ Your Company' on the left, and navigation links 'Features', 'Pricing', 'Blog', 'Pages', and 'Help' in the center. On the right of the header are two buttons: 'Sign Up' (highlighted in dark grey) and 'Log In' (white with a grey border). Below the header is a large white area with a heading 'Heading' in a large, bold, black font. Centered below the heading is a registration form box. Inside this box, there are five input fields: 'Usuario', 'Rol' (with a dropdown arrow), 'Contraseña', 'Repita contraseña', and 'Preferencias' (with a dropdown arrow). Below these fields is a checked checkbox labeled 'Checkbox'. At the bottom of the form box is a 'Button' labeled 'Button'. The footer of the page contains the 'mequps' logo on the left, followed by three columns of links: 'Section One' (Menu One, Menu Two, Menu Three), 'Section Two' (Menu One, Menu Two, Menu Three), and 'Section Three' (Menu One, Menu Two, Menu Three). To the right of these columns are social media icons for Twitter, Instagram, and Facebook. At the bottom left of the footer is the copyright notice '© Copyright 2019'.

Figura 30: Diseño del form para registrarse

- **Pantalla de sesión iniciada (P04):** A grandes rasgos es igual a la pantalla **P01**, pero con algunos pequeños cambios. Los botones de inicio de sesión habrán cambiado y ahora se podrá cerrar la sesión y los restaurantes mostrados están basados en la preferencia de los usuarios, información obtenida previamente en el registro del mismo.

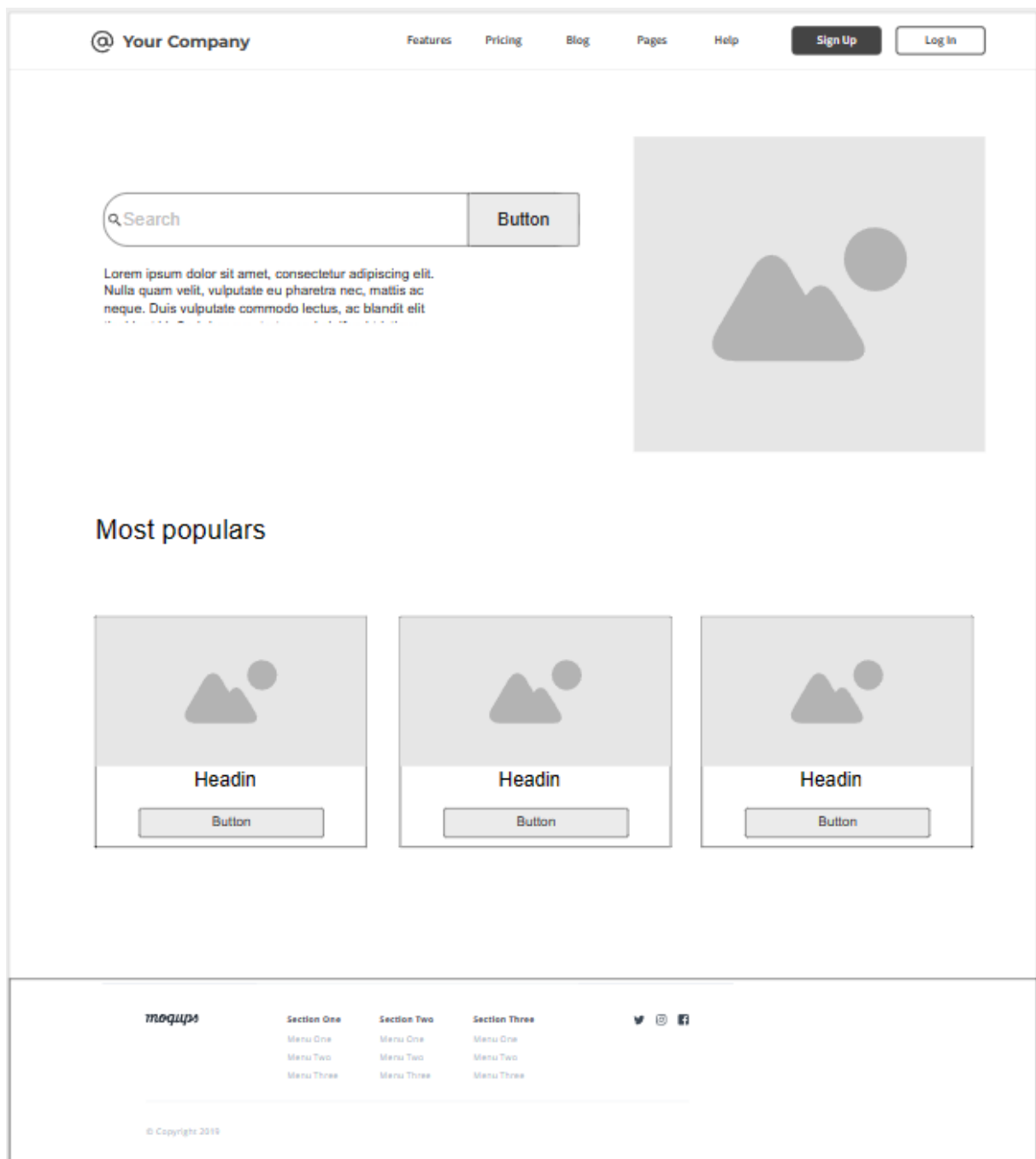


Figura 31: Diseño de pantalla de inicio con sesión iniciada

- **Pantalla de información del restaurante (P05):** Pantalla donde se muestran los menús del restaurante junto a los botones de añadir al

carrito y la información del menú como su nombre y precio. Para entrar en detalles, presionando el botón de “Vista rápida” se pueden ver los ingredientes y sus diferentes aportes nutricionales

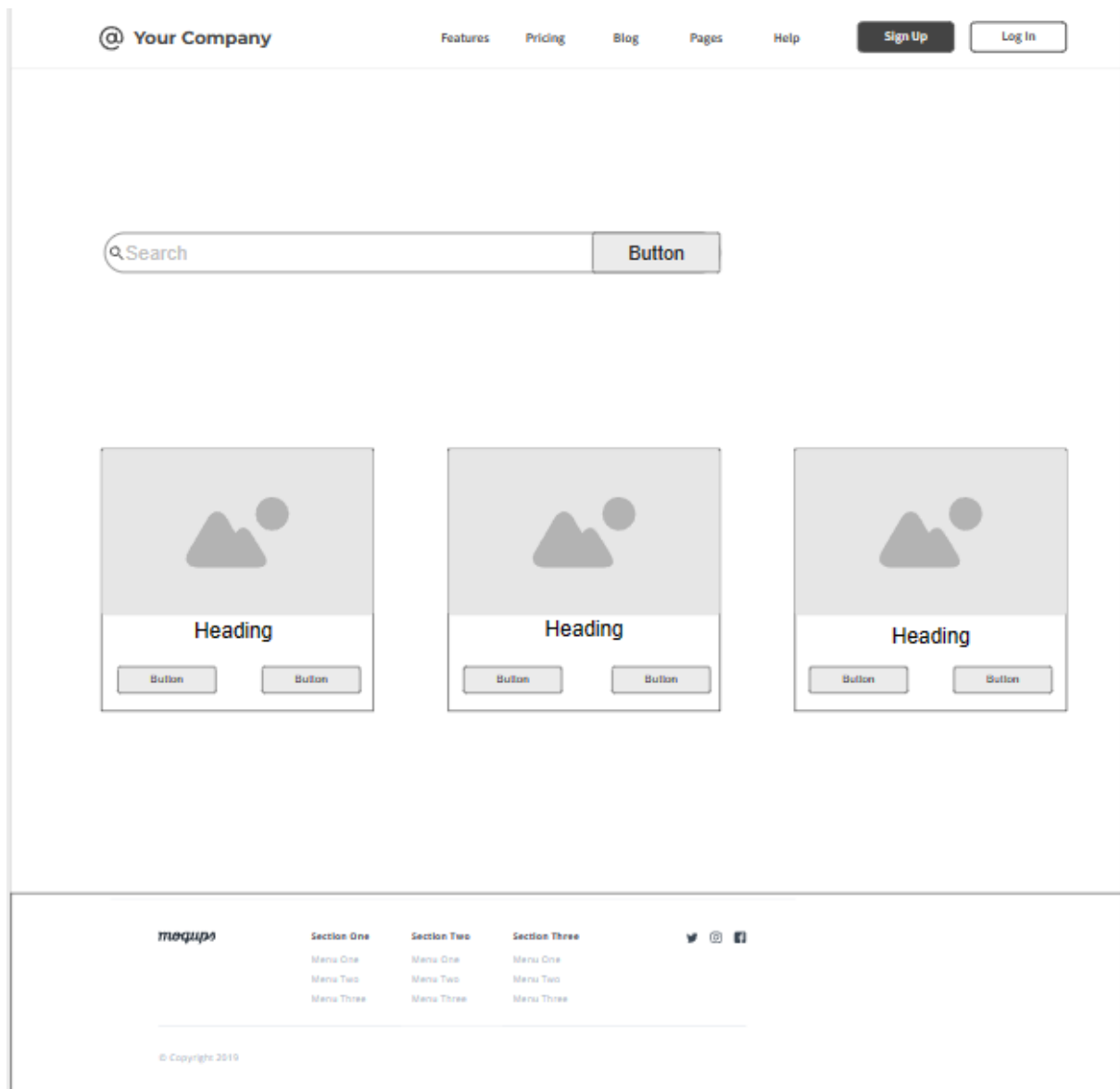


Figura 32: Diseño de pantalla de información de restaurante

- **Pantalla de información del menú (P06):** Carta o nota con los diferentes ingredientes y aportes nutricionales:

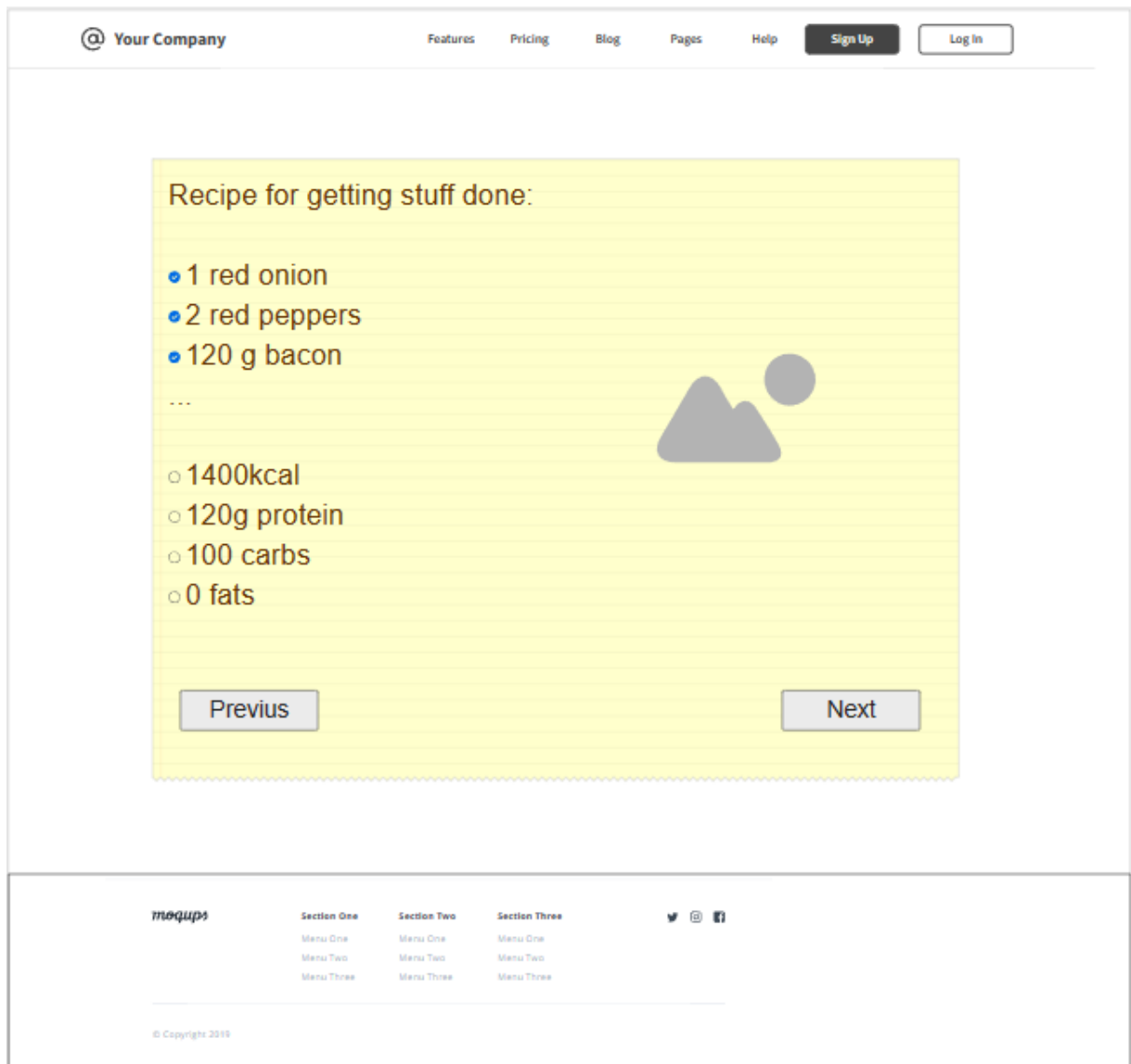


Figura 33: Diseño de pantalla de información de menú

- **Pantalla de realizar el pedido (P07):** Información general de la cesta de usuario junto con el precio a pagar y distintos campos a completar para poder completar la entrega satisfactoriamente junto con el modelo de pago a seguir y el botón de “Comprar ahora”.

Figura 34: Diseño de pantalla para realizar un pedido

- **Pantalla de registro de restaurante(P08):** Para el rol de restaurante, tenemos un formulario especial, un proceso guiado, donde está dividido en varias partes, para dar más dinamismo a nuestra página:
 - **Parte 1. Introducción del nombre del Restaurante:** Esencial en sí mismo, será como serán reconocidos los negocios y así lo reconocerán los usuarios. Está compuesta con un botón de

siguiente y en el paso posterior, en caso de cambiar de idea, el usuario podrá volver a este paso para editarlo.

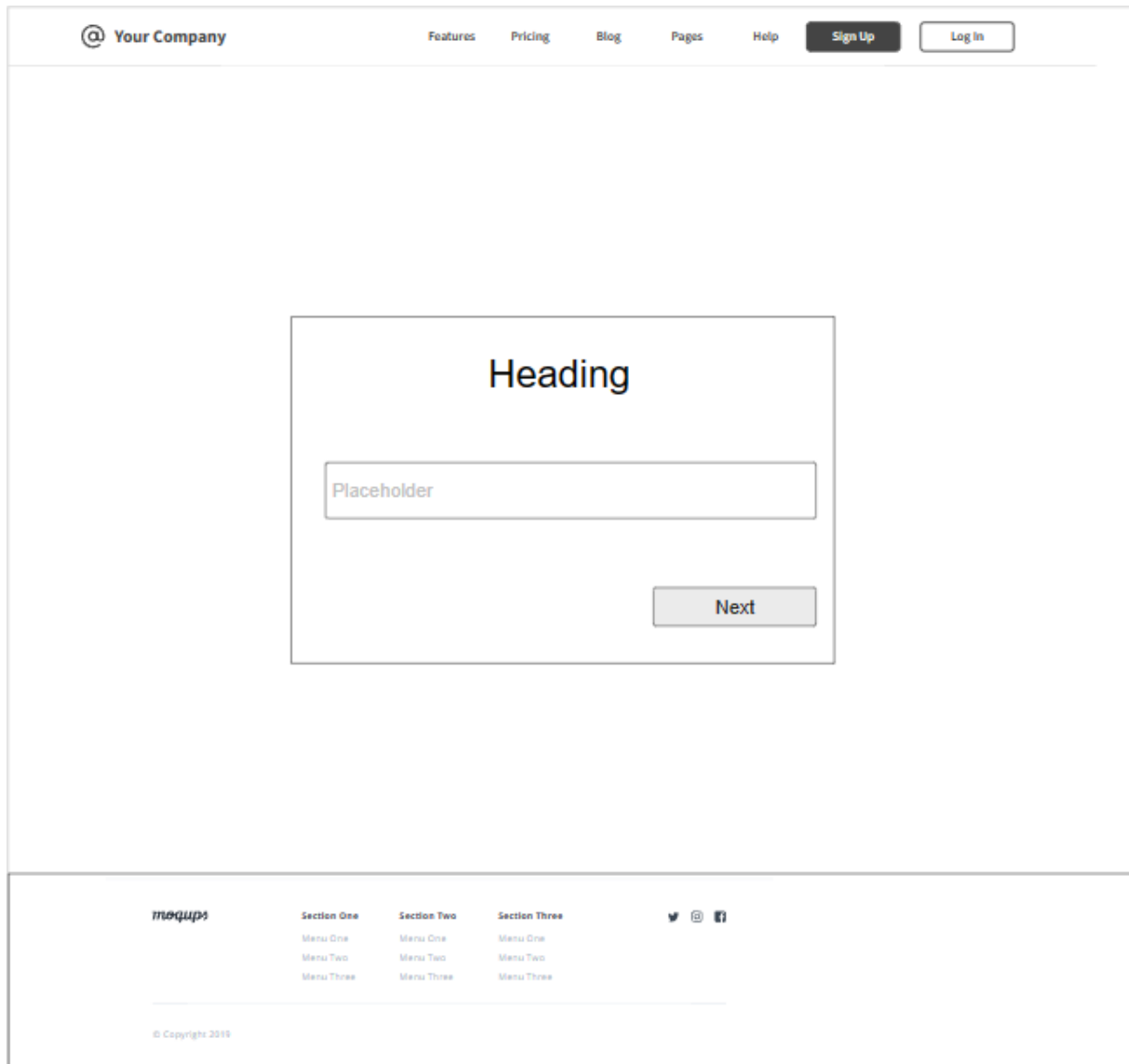


Figura 35: Diseño de pantalla 1 para la creación de un restaurante

- **Parte 2. Introducción de la imagen del restaurante:** A gusto del usuario podrá escoger una imagen que muestre su restaurante con algún tipo de eslogan o logo. Constará de 3 botones. Uno para subir la imagen (donde convertimos la imagen en cadena de

texto para poder ser guardada), otro para volver al paso anterior y otro para finalizar el proceso.

@ Your Company Features Pricing Blog Pages Help Sign Up Log In

Heading

Elegir archivo Archivo

Back Next

moqups

Section One
Menu One
Menu Two
Menu Three

Section Two
Menu One
Menu Two
Menu Three

Section Three
Menu One
Menu Two
Menu Three

Twitter Instagram Facebook

© Copyright 2019

Figura 36: Diseño de pantalla 2 para la creación de un restaurante

- **Pantalla inicio de sesión restaurante(P08):** En esta pantalla nos saldrán todos los restaurantes asociados a nuestro email, con la posibilidad de editarlos.

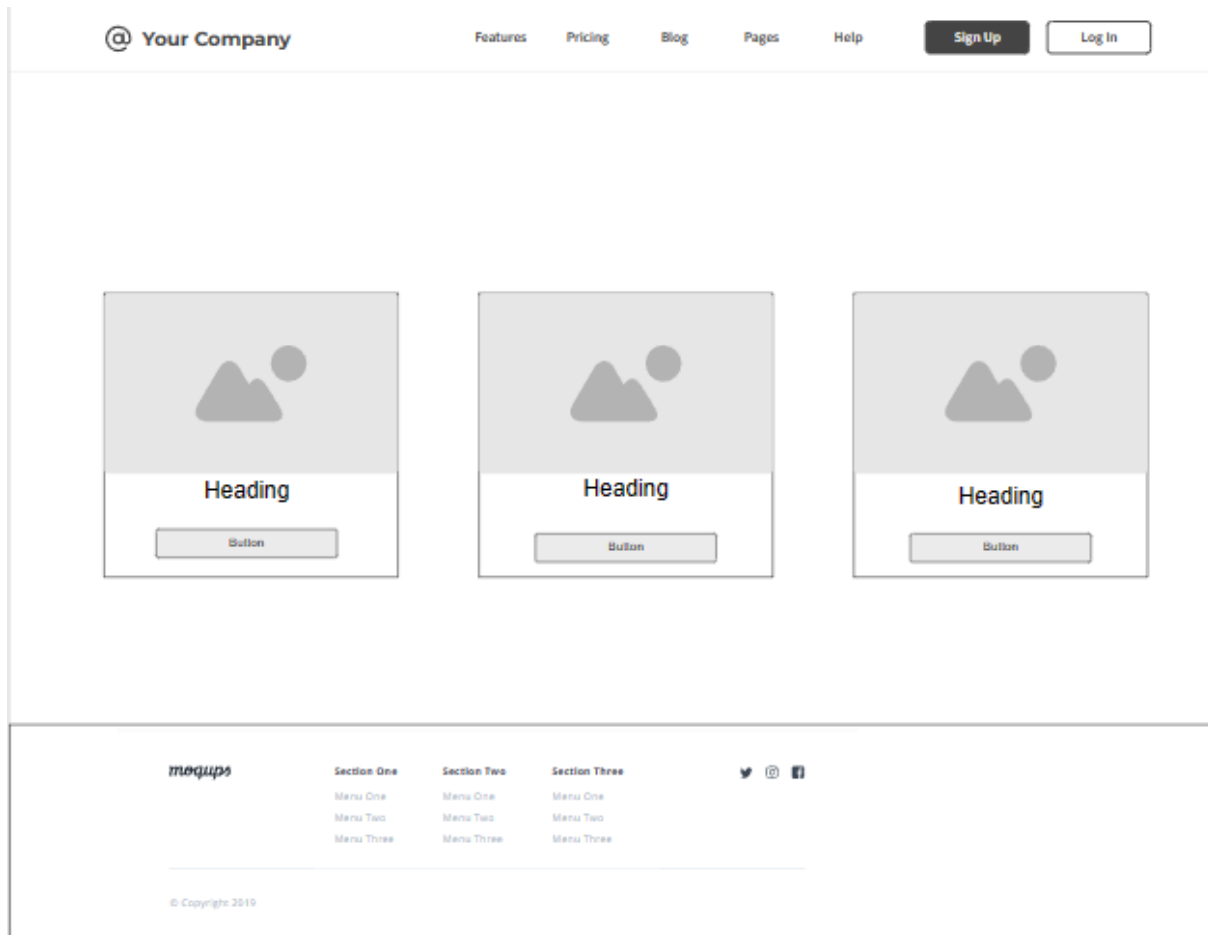


Figura 37: Diseño de pantalla de inicio de sesión de restaurante

- **Pantalla gestión restaurante(P09):** Una vez entramos en el restaurante que deseamos gestionar tendremos diferentes opciones: Tendremos diferentes tarjetas con los distintos menús que hay y con dos opciones: editarlos o eliminarlos. Para el caso del primero, hablaremos a continuación en la siguiente pantalla, y para el caso de eliminarlos saldrá un aviso de advertencia y si se confirma quedará eliminado.

También tendremos dos botones donde podremos editar la información del restaurante y donde podremos añadir un nuevo menú. Para añadir el restaurante tendremos el mismo paso que para la creación del restaurante y para añadir un menú se abrirá el mismo formulario que se usa para editarlo.

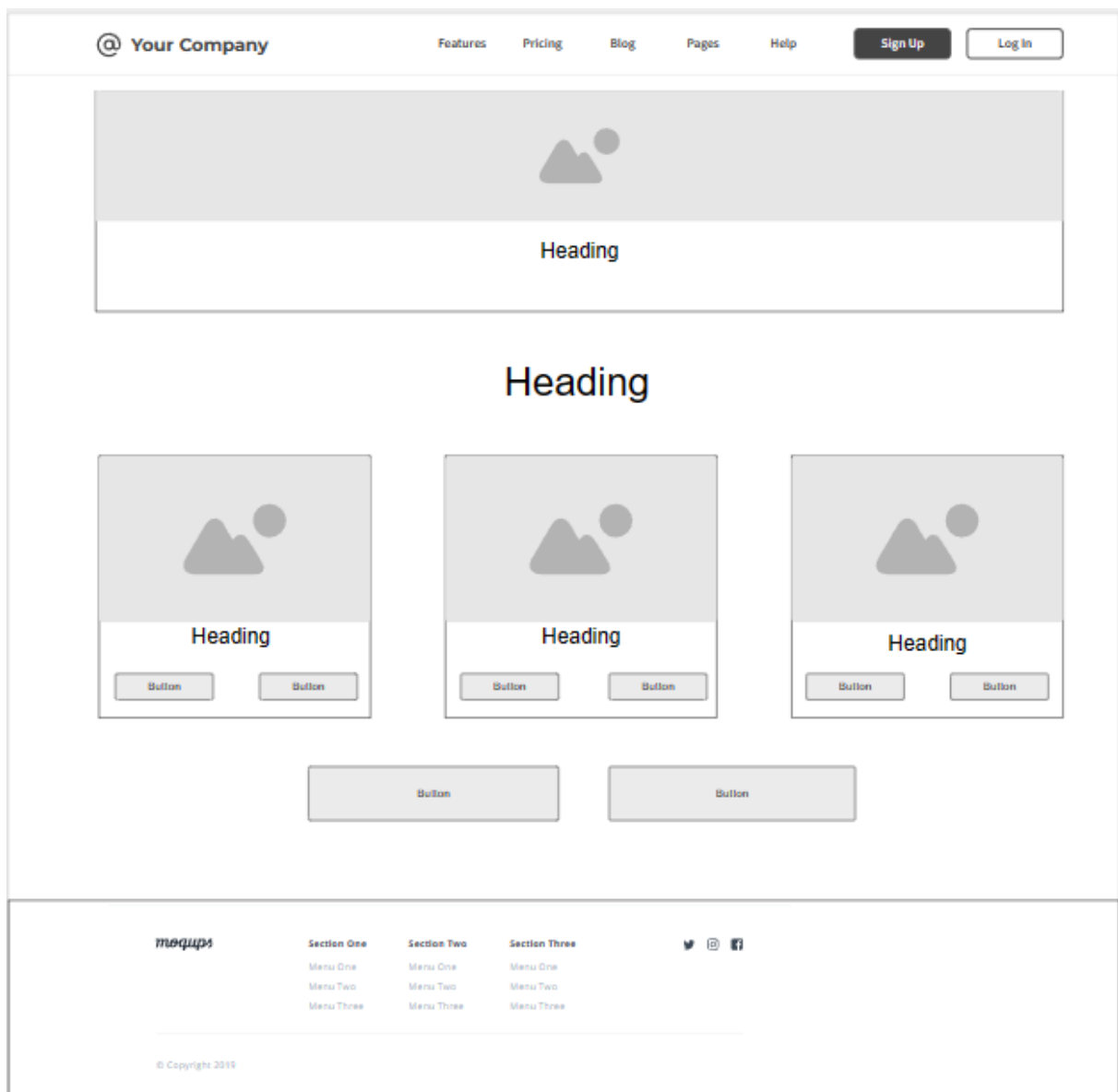


Figura 38: Diseño de pantalla de gestión de restaurante

- **Pantalla para añadir/editar y eliminar menú (P10):** Para estas dos pantallas son iguales tanto para añadir como para editar. Tendremos un formulario con los campos a rellenar y se guardarán/actualizarán en nuestro sistema, actualizando también visualmente con los datos proporcionados.

The image shows a web application mockup for 'Your Company'. The header includes navigation links: Features, Pricing, Blog, Pages, Help, Sign Up, and Log In. The main content area is a form for editing a menu item, titled 'Heading'. The form contains several input fields and buttons:

- A 'Heading' label followed by a text input field containing 'Nombre'.
- A 'Heading' label followed by a text input field containing 'Precio'.
- A 'Heading' label followed by a 'Button' and a text input field containing 'Archivo'.
- Two columns of inputs: 'Calorías' and 'Proteínas'.
- Two columns of inputs: 'Carbohidratos' and 'Azucares'.
- A 'Button' at the bottom.

The footer includes the 'maquipo' logo, three sections of menu items (Section One, Section Two, Section Three), social media icons, and a copyright notice for 2019.

Figura 39: Pantalla de edición de menú

- **Pantalla de administrador (P11):** Vista general para el rol de administrador donde tiene acceso a distintas informaciones proporcionadas por el sistema y acceso a distintas pestañas para gestionar la página, como ver usuarios restaurantes y menús globales. También se pueden ver los últimos restaurantes y usuarios registrados en la vista principal.

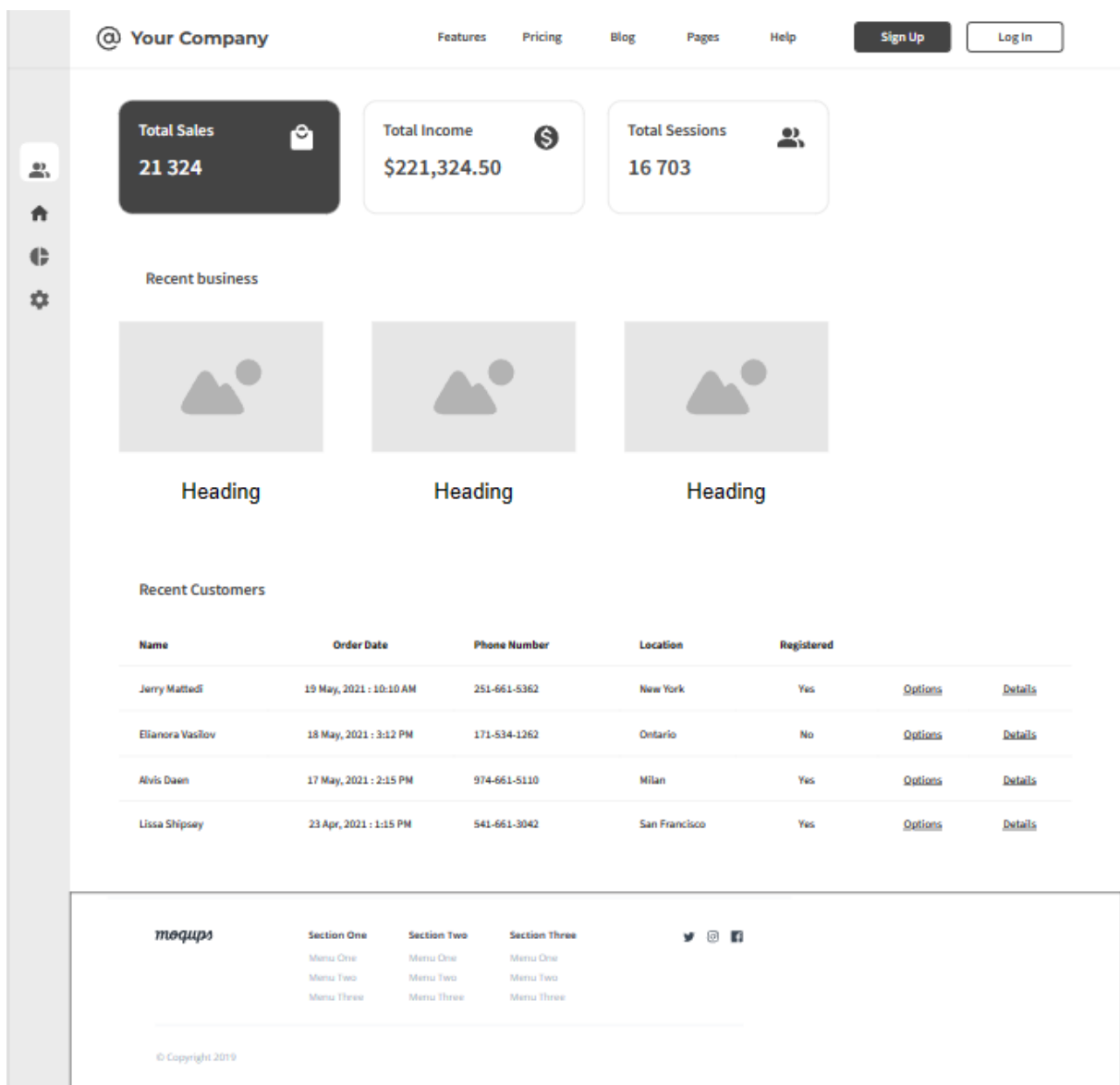
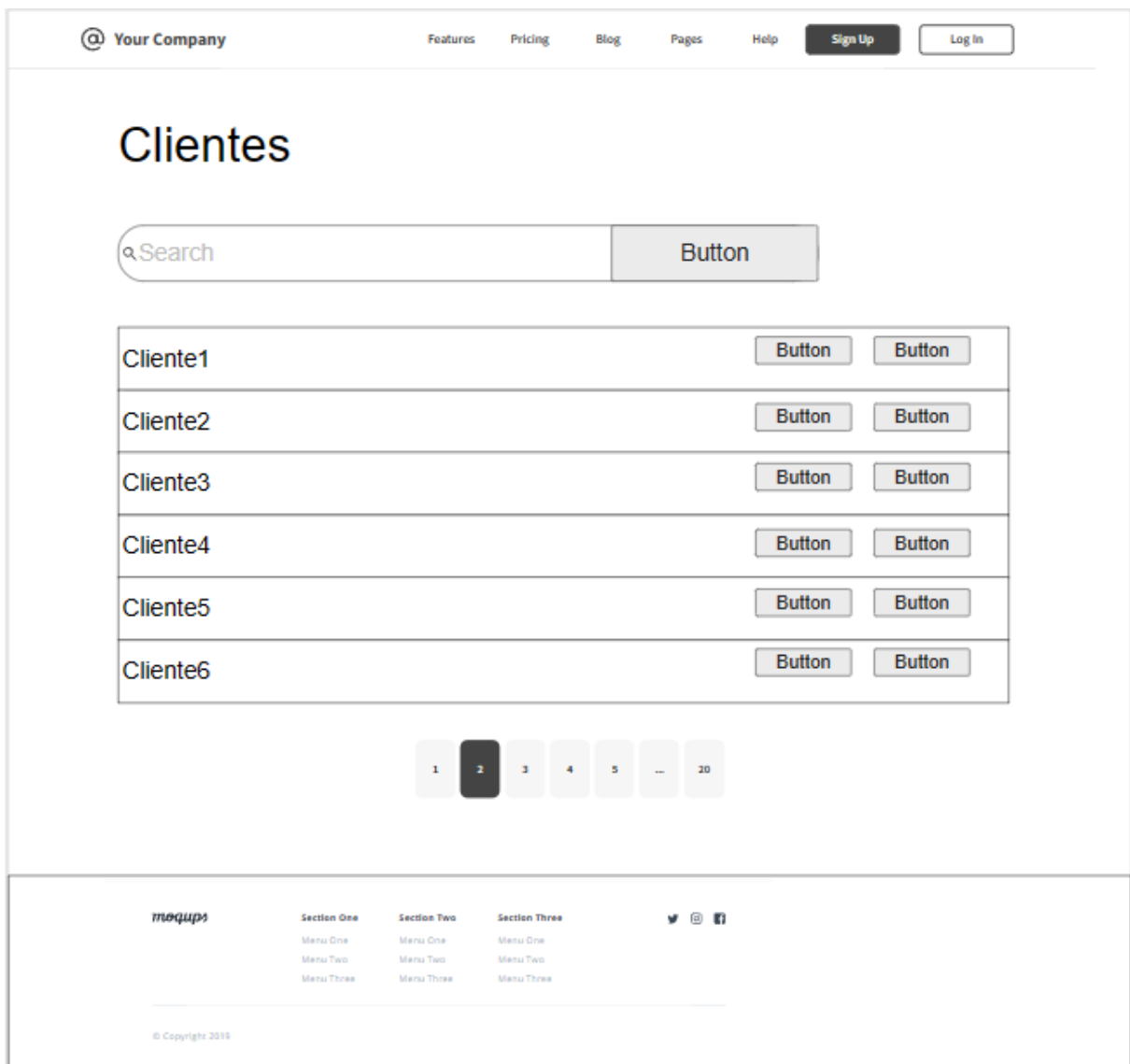


Figura 40: Diseño de pantalla de modo administrador

- **Pantalla de gestión de usuarios(P12):** Vista para administrador donde podrán editar y eliminar usuarios del sistema, así como buscar alguno en concreto:

**Figura 41: Diseño de pantalla de gestión de usuarios**

- **Pantalla de gestión de restaurantes(P13):** Vista para administrador donde se podrán editar y eliminar los restaurantes del sistema, así como buscar alguno en concreto:

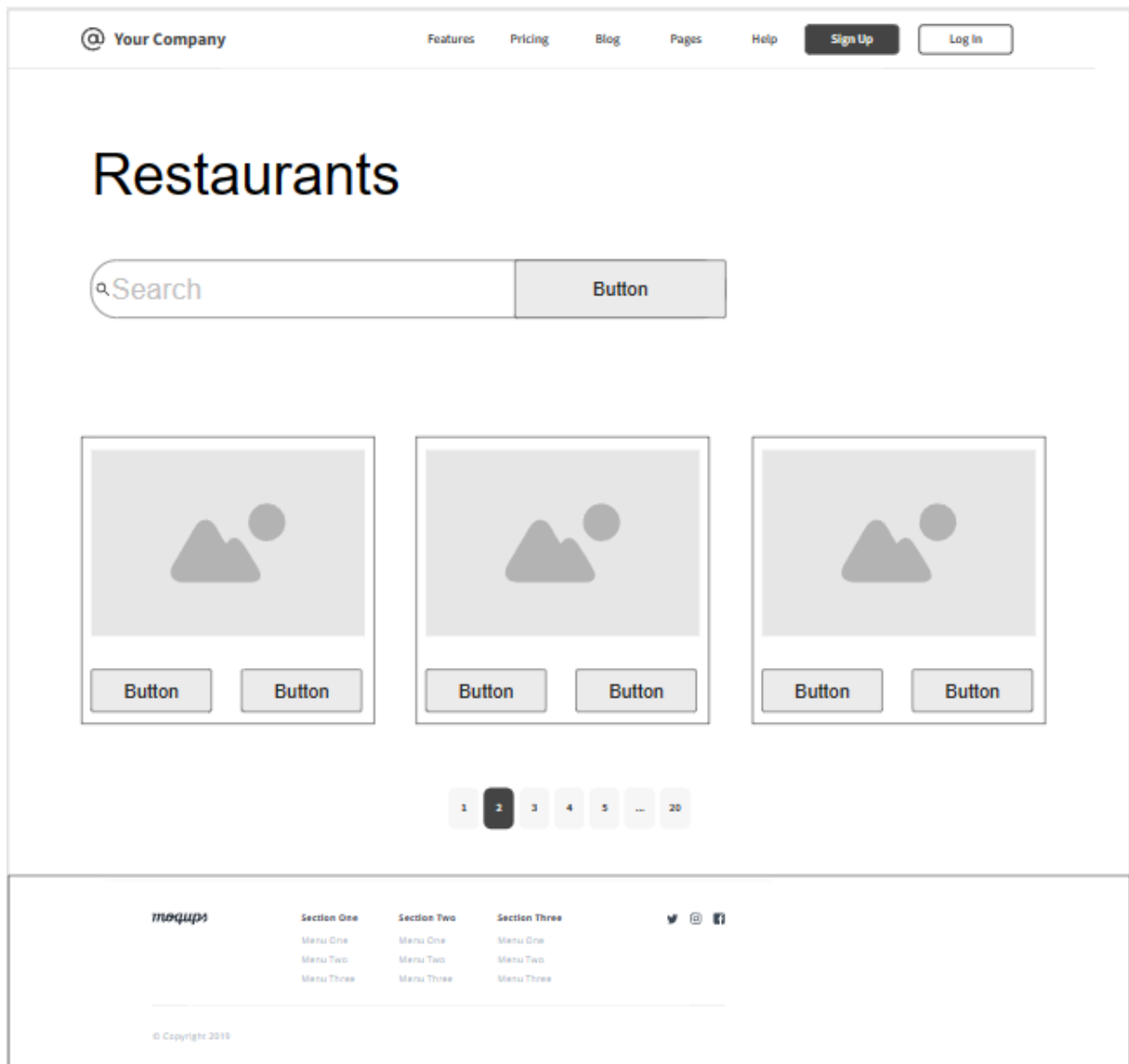


Figura 42: Pantalla de gestión de restaurantes

- **Pantalla de gestión de menús(P14):** Vista para administrador donde se podrán editar y eliminar los restaurantes del sistema, así como buscar alguno en concreto.

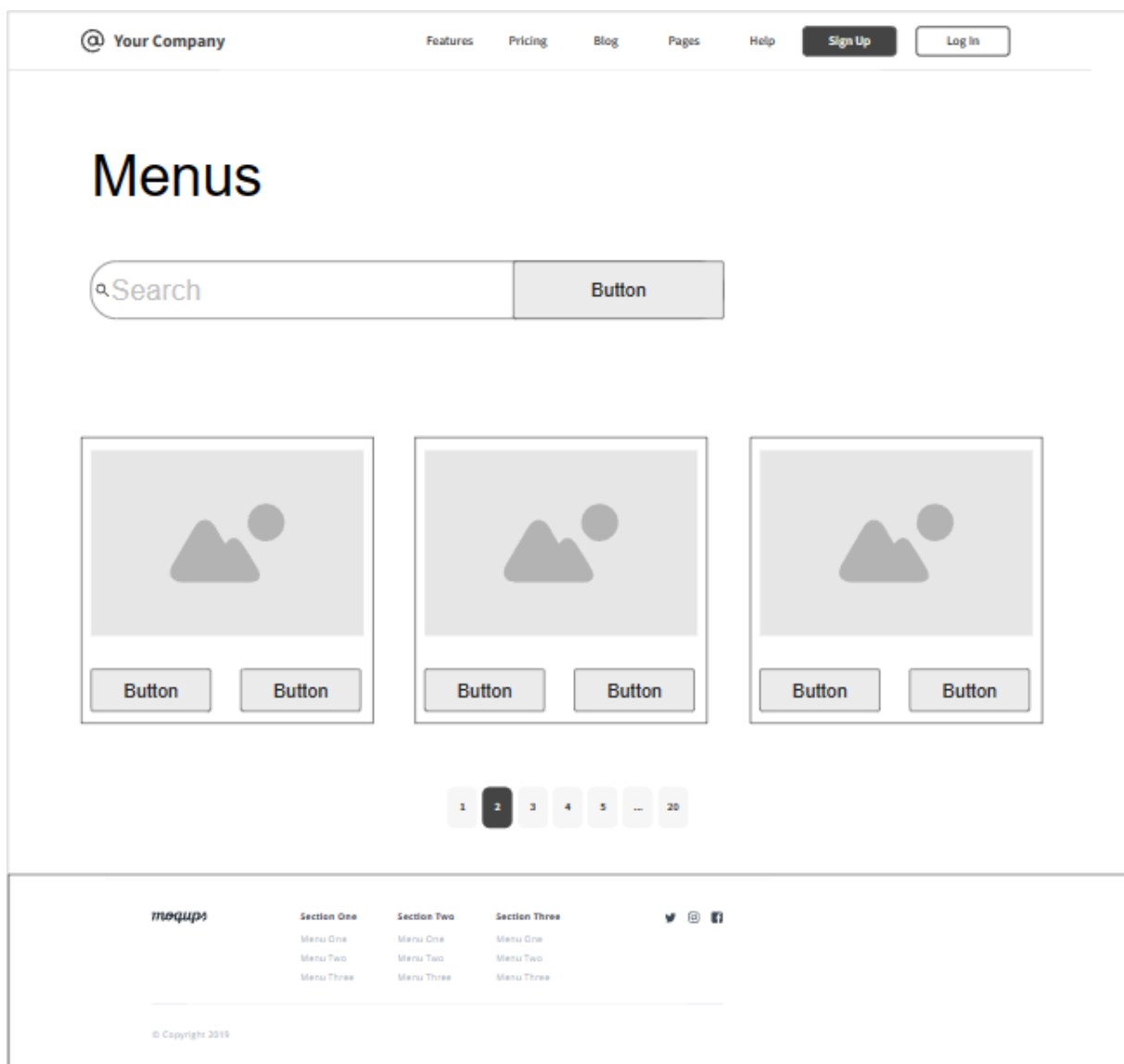


Figura 43: Diseño de pantalla de gestión de menús

Para una visualización interactiva hemos creado una página diferente, Marvel App, para que los clientes puedan navegar a través de ellos y darnos sus opiniones.

<https://marvelapp.com/prototype/c54jhb9>

2.5.3. Servidor

El servidor incluye las capas de acceso a los servicios, la lógica de negocio, persistencia junto con la base de datos. Las capas de la lógica de negocio y servicios son las implementaciones más importantes del proyecto. Dentro de esta lógica añadiremos las entidades pertinentes con cada uno de los servicios que ofrece. Cada uno de estos servicios deberán estar implementados aportando flexibilidad, concurrencia e integridad.

A continuación, un diagrama Entidad – Relación para tener una idea general de cómo sería:

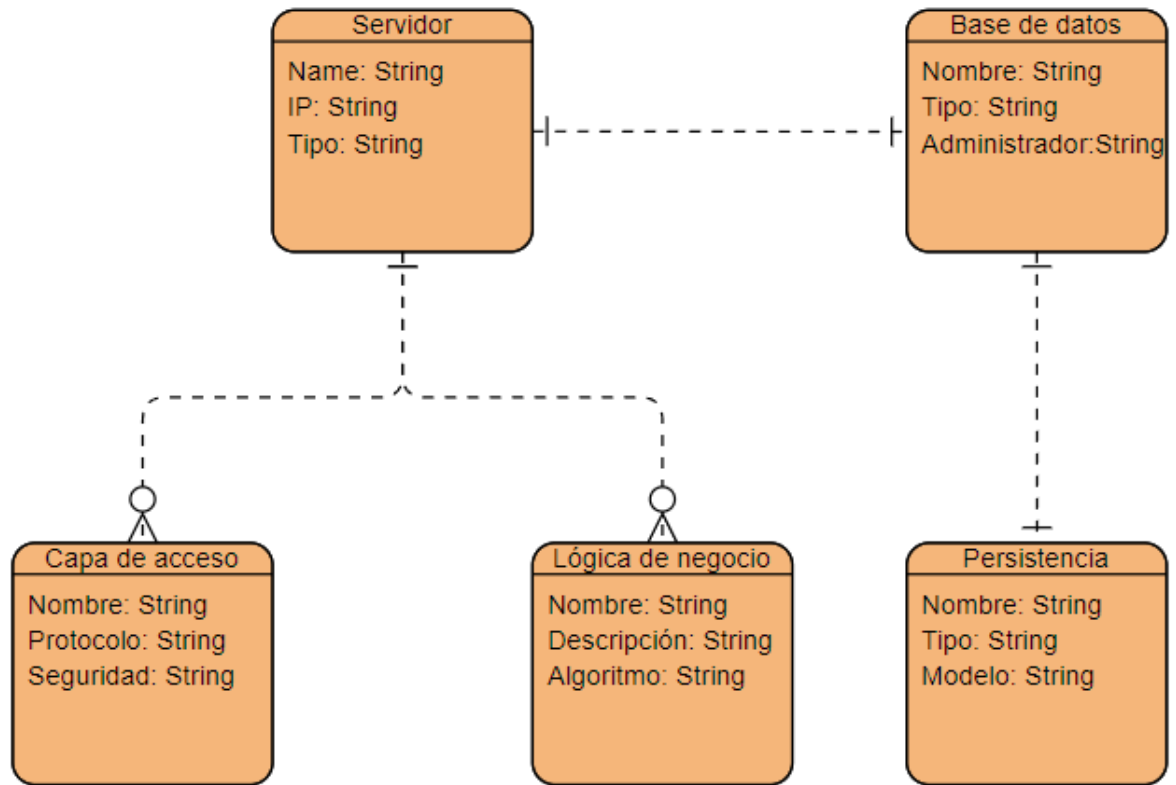


Figura 44: Diagrama ER de Lunch At Home

3. Desarrollo del proyecto

Para el desarrollo del proyecto hemos seguido una metodología ágil de 3 iteraciones, dividiendo los desarrollos en hitos y guiándonos por un usuario de prueba que puede ir dando su visto bueno respecto al desarrollo implementado y aportando nuevas ideas al proyecto.

3.1. Primera iteración

Una vez que hemos formado el diseño de nuestra aplicación, en la primera iteración, explicaremos el desarrollo de las partes principales de nuestra aplicación web, como el registro, login, edición y borrado de los usuarios que hay en nuestro sistema.

A partir de ahora incluiremos un nuevo término llamado: puntos de historia. Se asignarán a las tareas y nos servirá para estimar la carga de trabajo de ellas. Se utilizan para evaluar el esfuerzo necesario en función de varios factores:

- Dificultad en la tarea
- Competencias técnicas
- Posibles riesgos asociados

Estos puntos de historia tendrán un rango del 1 al 5, siendo 1 el mínimo valor, lo que suponen poca dificultad y poca carga de trabajo y 5 el máximo valor, una mayor carga, dificultad y riesgos.

Historias de usuario

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US001	Estimación	2	Prioridad	ALTA

Como	Usuario	Quiero	Registrar usuario	Para	Utilizar la aplicación web y realizar una tarea
Criterios de validación		Rellenar correctamente los datos del formulario, para crear correctamente un nuevo usuario en la aplicación.			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 8: Historia de usuario US002: Inicio sesión del usuario

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US001	Estimación	2	Prioridad	ALTA
Como	Usuario	Quiero	Iniciar sesión	Para	Utilizar la aplicación web y pedir comida a domicilio
Criterios de validación		El usuario introduce correctamente sus datos, el sistema los valida, y se inicia la sesión del usuario correctamente.			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 9: Historia de usuario US003: Modificación usuario

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	

ID	US001	Estimación	2	Prioridad	ALTA
Como	Usuario	Quiero	Modificar usuario	Para	Actualizar datos de su perfil
Criterios de validación		Rellenar el formulario correctamente para editar el usuario en el sistema.			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 10: Historia de usuario US004: Eliminar usuario

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US001	Estimación	3	Prioridad	ALTA
Como	Usuario	Quiero	Buscar usuario	Para	Realizar una gestión de usuarios
Criterios de validación		Introducir correctamente el nombre o correo del usuario, lo devuelve correctamente.			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 11: Historia de usuario US005: Búsqueda de usuario

3.1.1. Análisis

En este punto mostraremos los diagramas de actividad de cada historia de usuario que hemos comentado anteriormente.

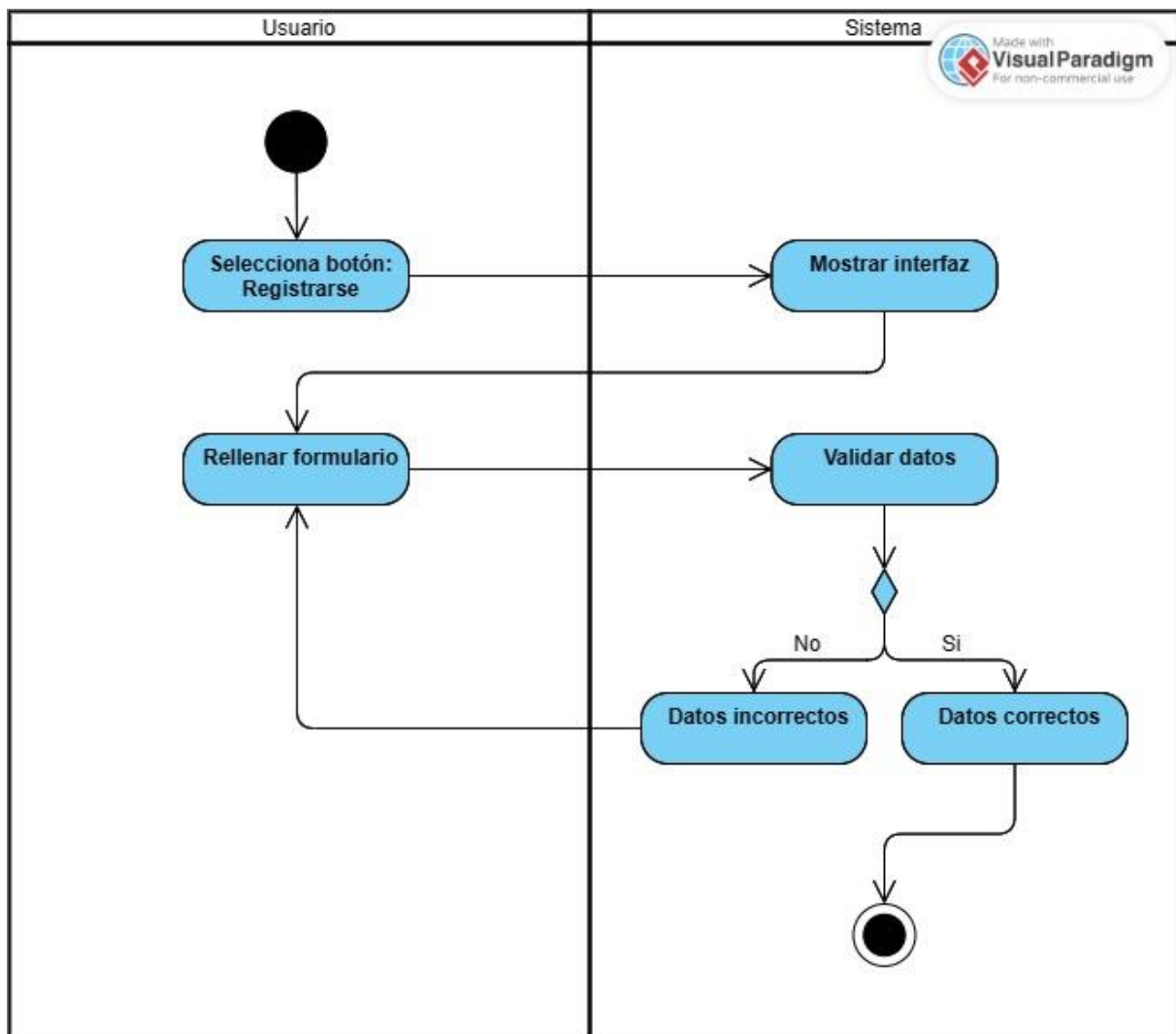


Figura 45: Diagrama de actividad – Historia de usuario US001: Registrar usuario

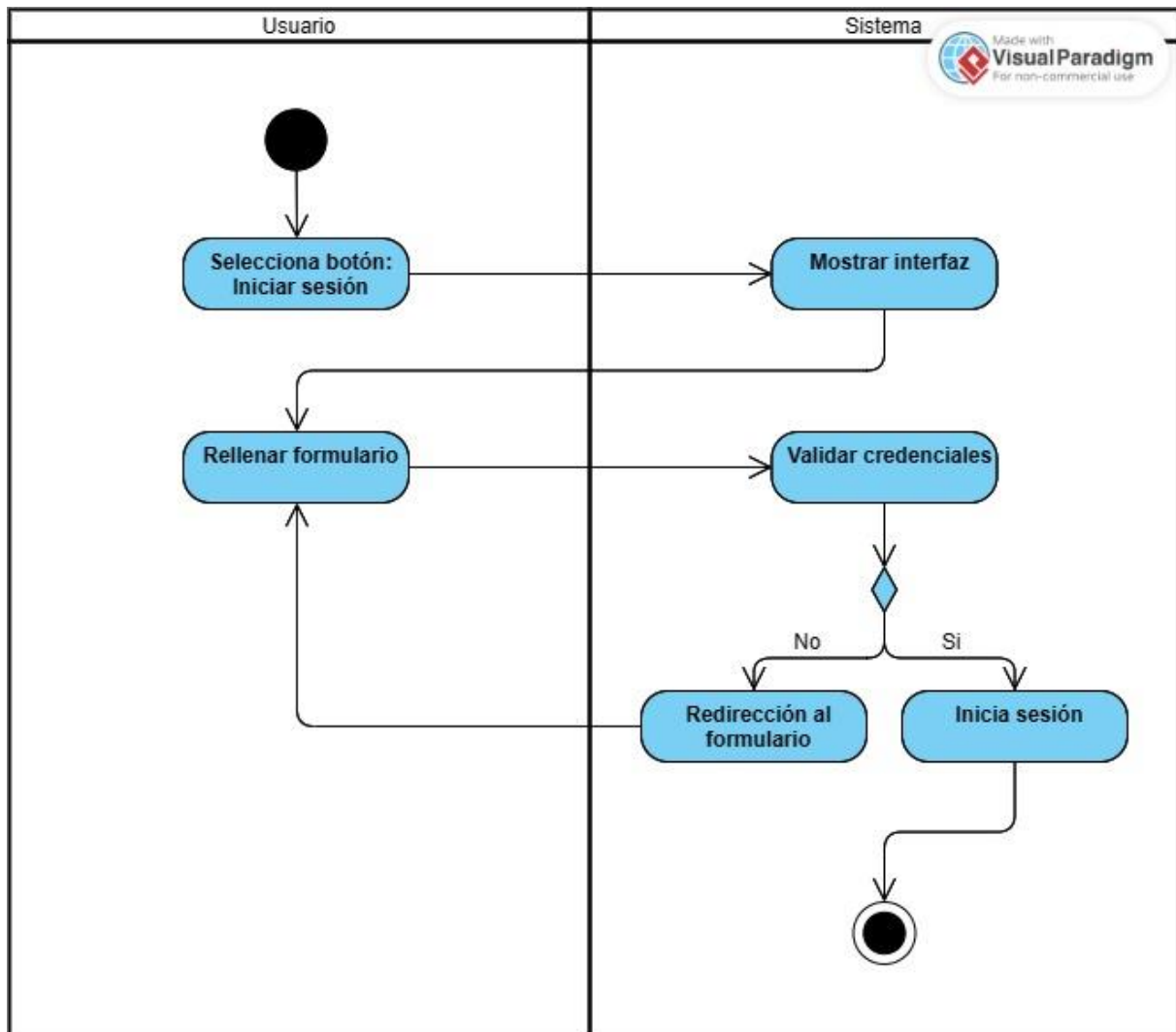


Figura 46: Diagrama de actividad – Historia de usuario US002: Iniciar sesión

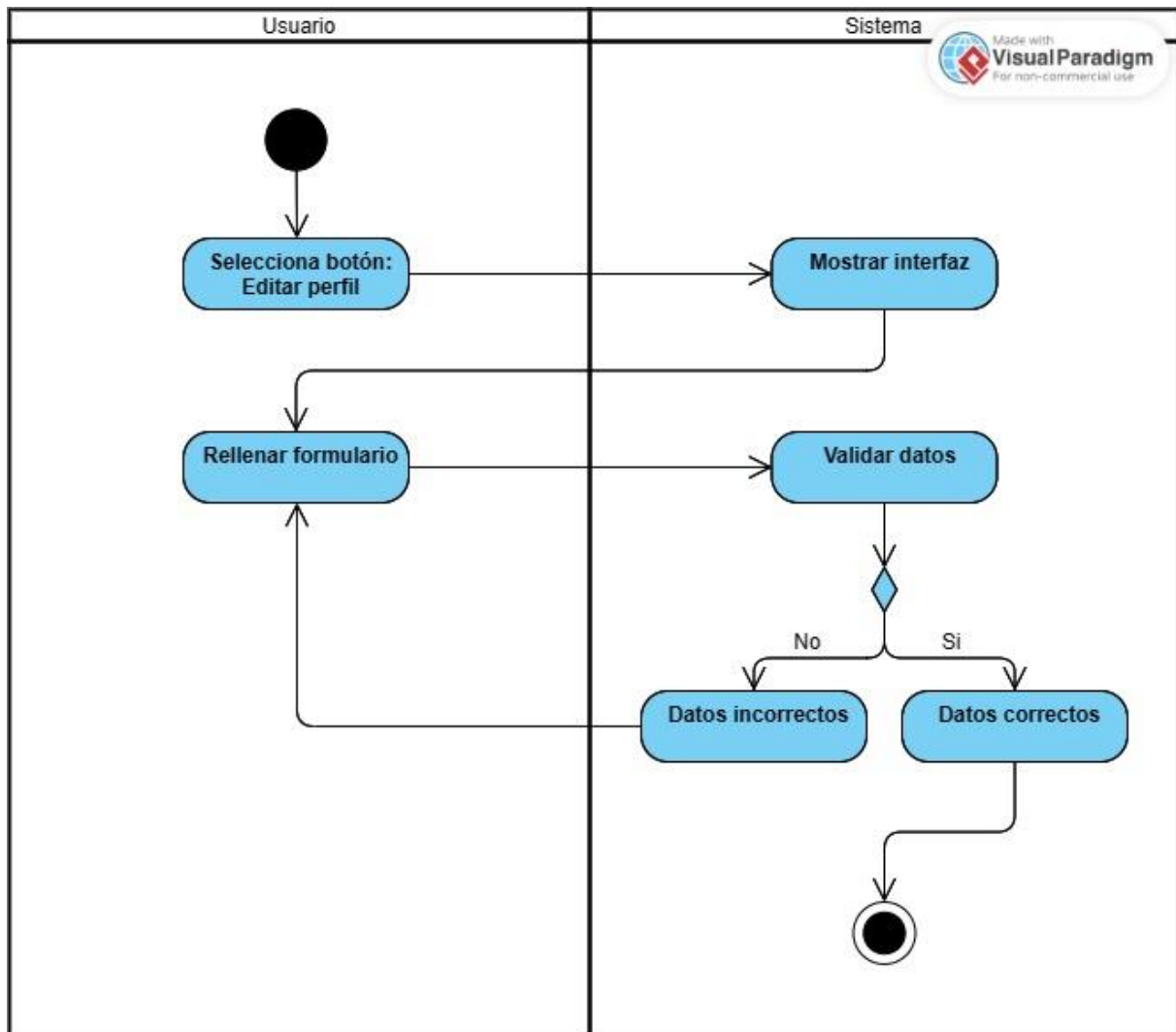


Figura 47: Diagrama de actividad – Historia de usuario US003: Modificar usuario

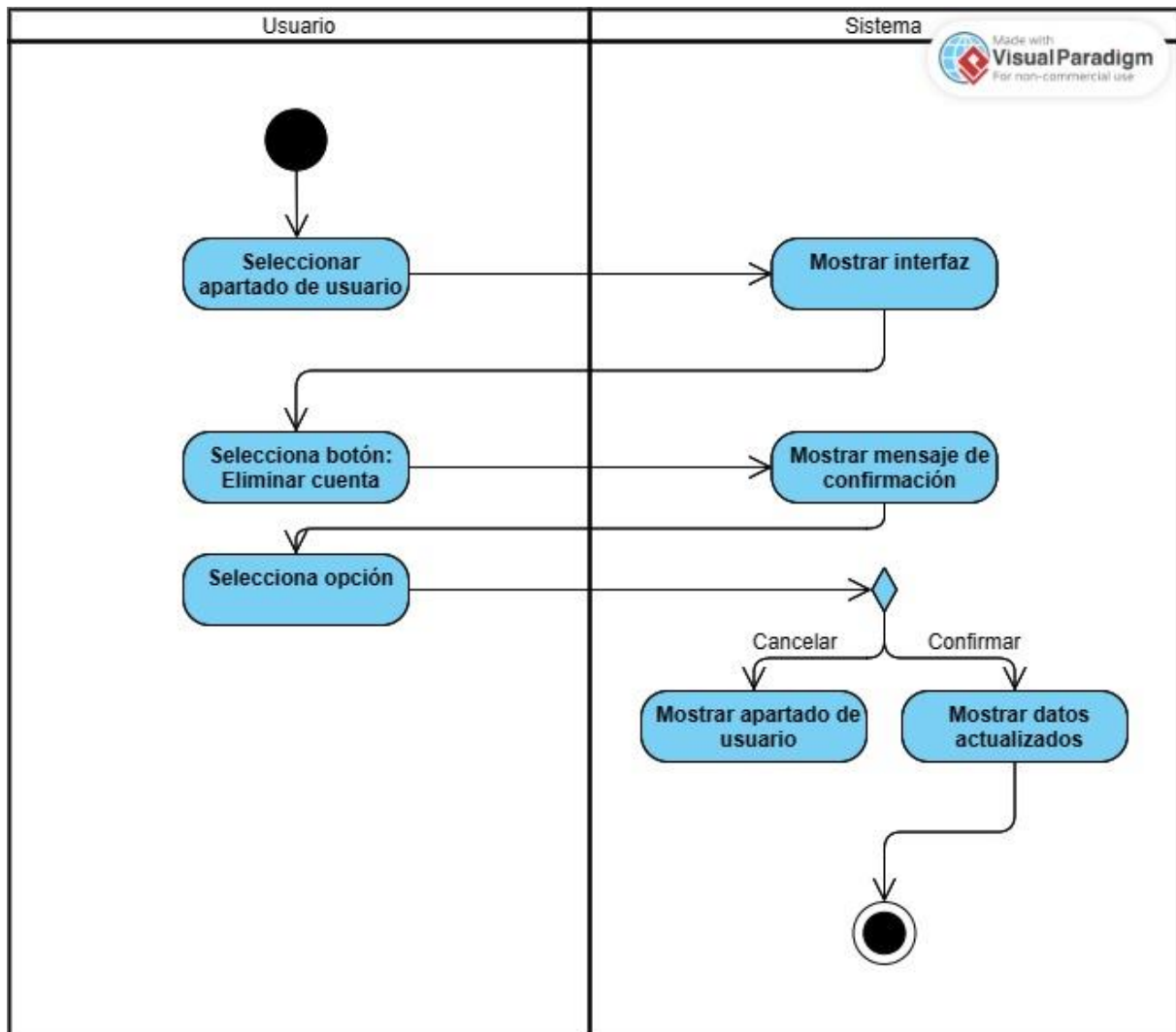


Figura 48: Diagrama de actividad – Historia de usuario US004: Eliminar usuario

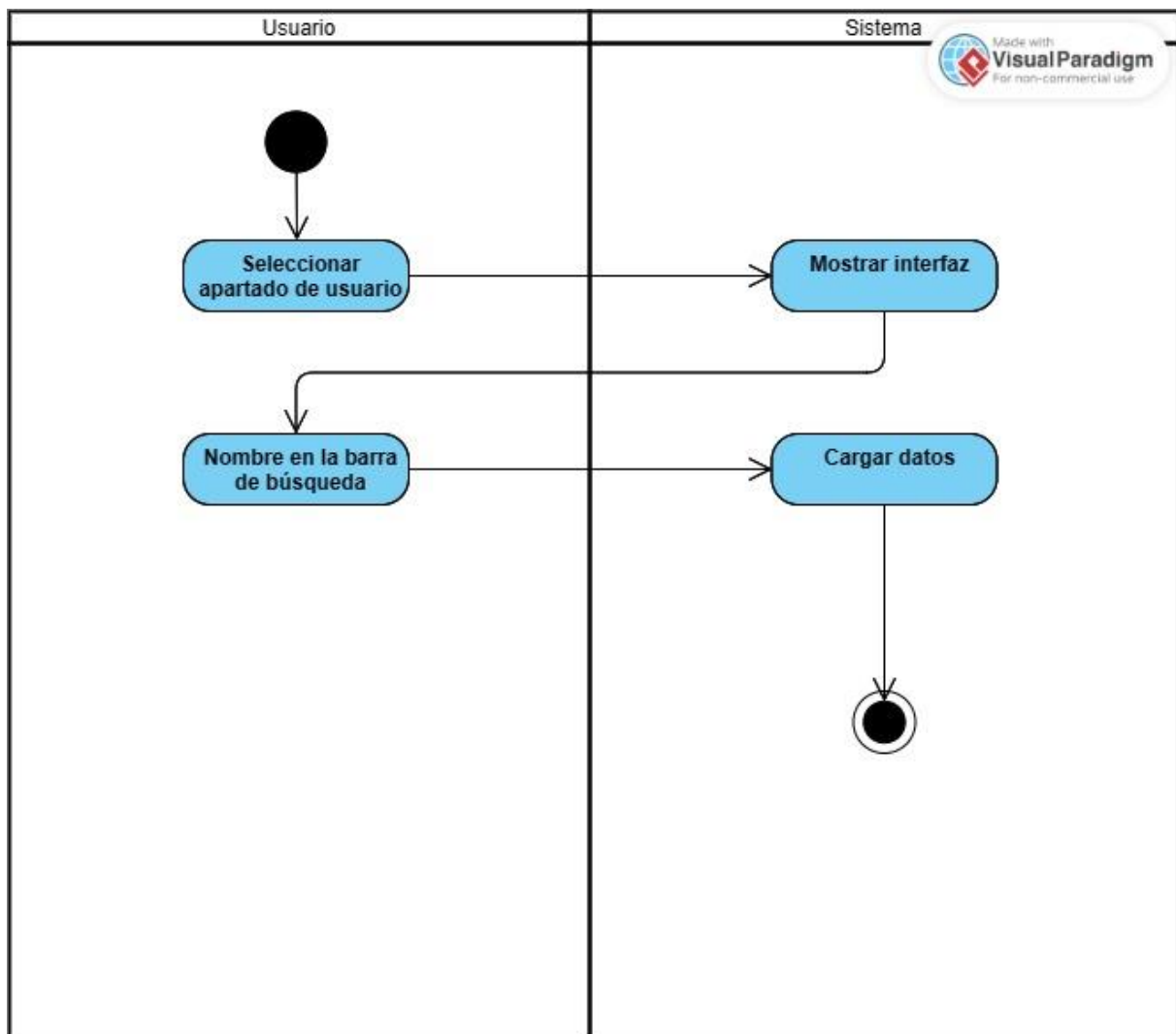


Figura 49: Diagrama de actividad – Historia de usuario US005: Buscar usuario

3.1.2. Planificación de la iteración

Al ser la primera iteración, hay tareas que no se pueden incluir en una historia, y por tanto se explicarán ahora.

1. Análisis del sistema general: Para tener información más detallada, se puede consultar el apartado **2.2.2 Análisis inicial del sistema** de esta memoria.

2. Diseño de la arquitectura: Para tener información más detallada, se puede consultar el apartado **2.3.1 Arquitectura del sistema** de esta memoria
3. Instalación y configuración de **Pycharm**: Hay que instalar y configurar correctamente nuestro entorno de trabajo para trabajar tanto con la parte cliente como con la parte servidor de nuestra aplicación.
4. Instalación y configuración de **Python + Flask**: Instalación y configuración del lenguaje con el que trabajará el servidor de nuestra aplicación.
5. Instalación y configuración del SGBD **MySQLWorkBench 8.0 CE**. Como en los apartados anteriores, la instalación y configuración de la base de datos con la que trabajaremos en nuestra aplicación.
6. Instalación y configuración de **React**: Instalación y configuración del framework utilizado para la implementación de nuestra parte cliente. Se ha realizado el desarrollo de los componentes generales usando este framework, desde los estilos generales de nuestra aplicación hasta los botones y las barras de navegación.
7. Instalación de **GitBash** y **Office** como herramientas de control de versiones y herramienta para redactar la memoria.

Proceso de documentación previa:

Aún no hemos hablado sobre cómo ha sido el proceso de aprendizaje de las tecnologías necesarias para construir el sistema. Esto se debe a que es difícil estimar cuánto tiempo y esfuerzo sería necesario para este proceso. Sin embargo, es importante destacar que se ha dedicado mucho tiempo a estudiar y leer muchos documentos sobre cómo funcionan, para encontrar la forma más simple e intuitiva para utilizarlas.

A continuación, se presenta una tabla con un Backlog de esta primera iteración:

BACKLOG DE LA PRIMERA ITERACIÓN	NOMBRE DEL PROYECTO	COORDINADOR DEL PROYECTO
	TFG – LunchAtHome	Francisco José Reyes Campos
Entidad	Cliente	
ID	Tarea	Proyecto
Inicio del proyecto		
T0001	Análisis del sistema	
T0002	Diseño de la arquitectura y desarrollo de los componentes base	
T0003	Instalación y configuración de Pycharm	
T0004	Instalación y configuración de Flask y Python y React	
T0005	Instalación de SQL WorkBench	
T0006	Instalación de otros softwares necesarios	
US001: Registrar usuario		
T0101	Diseño backend registrar usuario	Servidor
T0102	Diseño frontend registrar usuario	Cliente
T0103	Implementación backend registrar usuario	Servidor
T0104	Implementación frontend registrar usuario	Cliente
US002: Iniciar sesión		
T0201	Diseño backend iniciar sesión	Servidor
T0202	Diseño frontend iniciar sesión	Cliente
T0203	Implementación backend iniciar sesión	Servidor
T0204	Implementación frontend iniciar sesión	Cliente
US003: Editar usuario		
T0301	Diseño backend editar usuario	Servidor
T0302	Diseño frontend editar usuario	Cliente

T0303	Implementación backend editar usuario	Servidor
T0304	Implementación frontend editar usuario	Cliente
US004: Eliminar usuario		
T0401	Diseño backend eliminar usuario	Servidor
T0402	Diseño frontend eliminar usuario	Cliente
T0403	Implementación backend eliminar usuario	Servidor
T0404	Implementación frontend editar usuario	Cliente
US005: Buscar usuario		
T0501	Diseño backend buscar usuario	Servidor
T0502	Diseño frontend buscar usuario	Cliente
T0503	Implementación backend buscar usuario	Servidor
T0504	Implementación frontend buscar usuario	Cliente

Tabla 12: Backlog primera iteración**3.1.3. Diseño**

Una vez completado el diseño e implementación del backend de las historias que hemos detallado anteriormente del usuario, se procedió de la misma manera a llevar a cabo el diseño e implementación del frontend. Esto significa que, simultáneamente, se ha trabajado en ambas partes.

Este enfoque incremental y paralelo, tanto en el backend como en el frontend, facilita una integración fluida y progresiva en ambas partes de la aplicación, con el fin de asegurar un avance coherente y consistente en todas las áreas del desarrollo.

3.1.3.1. Servidor

En este apartado, se ha llevado a cabo el diseño e implementación de la entidad cliente, que es el usuario en concreto de la aplicación. Tendrá almacenada toda su información personal importante como sus credenciales para acceder a la aplicación.

3.1.3.2. Cliente

En este apartado, la parte del cliente será el encargado de mostrar las vistas de la interfaz de la pantalla de inicio, inicio de sesión, registro, entre las distintas opciones que podrá hacer el cliente

- Componentes comunes
 - Header
 - Footer
 - Barra de búsqueda
- Componentes
 - Inicio
 - Login
 - Sign Up
- Routing Module:
 - AuthContext
- Modelos
 - Cliente
- Enrutamiento (React Router)
 - Rutas públicas
 - Rutas privadas

3.1.4. Resumen del progreso

En términos generales, el progreso ha sido muy lento debido a toda la cantidad de información que había que leer e interpretar, a la vez de ir probando las distintas

cosas aprendidas para usar la más eficiente por lo que contar las horas por semanas será la manera más eficiente.

Se han realizado todas las actividades en los tiempos planificados y en la siguiente iteración se desarrollará la entidad restaurante con todo lo que conlleva, un diseño distinto y funcionalidades adicionales.

A continuación, un gráfico del progreso de esta iteración:

BACKLOG DE LA PRIMERA ITERACIÓN	NOMBRE DEL PROYECTO	COORDINADOR DEL PROYECTO					
	TFG – LunchAtHome	Francisco José Reyes Campos					
Entidad	Cliente						
ID	Tarea	Proyecto	Puntos de historia por tareas	Primera iteración (SEMANALES)			
Inicio del proyecto				25/09	02/10	09/10	Total de horas por tarea
				W39	W40	W41	
T0001	Análisis del sistema						
T0002	Diseño de la arquitectura y desarrollo de los componentes						
T0003	Instalación y configuración de Pycharm						
T0004	Instalación y configuración de Flask y Python y React						
T0005	Instalación de SQL WorkBench						
T0006	Instalación de otros softwares necesarios						
US001: Registrar usuario							

T0101	Diseño backend registrar usuario	Servidor	1	1			
T0102	Diseño frontend registrar usuario	Cliente	1	1			
T0103	Implementación backend registrar usuario	Servidor	1	1			
T0104	Implementación frontend registrar usuario	Cliente	1	1			
US002: Iniciar sesión							
T0201	Diseño backend iniciar sesión	Servidor	1	1			
T0202	Diseño frontend iniciar sesión	Cliente	1	1			
T0203	Implementación backend iniciar sesión	Servidor	1	1			
T0204	Implementación frontend iniciar sesión	Cliente	1	1			
US003: Editar usuario							
T0301	Diseño backend editar usuario	Servidor	1		1		100%
T0302	Diseño frontend editar usuario	Cliente	1		1		100%
T0303	Implementación backend editar usuario	Servidor	1		1		100%
T0304	Implementación frontend editar usuario	Cliente	1		1		100%
US004: Eliminar usuario							
T0401	Diseño backend eliminar usuario	Servidor	1		1		100%
T0402	Diseño frontend eliminar usuario	Cliente	1		1		100%
T0403	Implementación backend eliminar usuario	Servidor	1		1		100%
T0404	Implementación frontend eliminar usuario	Cliente	1		1		100%
US005: Buscar usuario							
T0501	Diseño backend buscar usuario	Servidor	1		1		100%
T0502	Diseño frontend buscar usuario	Cliente	1		1		100%

T0503	Implementación backend buscar usuario	Servidor	1		1		100%
T0504	Implementación frontend buscar usuario	Cliente	1		1		100%
Tareas restantes			20	12	0	0	
Esfuerzo ideal			20	13,66	6,68	0	
Puntos de historia restantes			9	9	0	0	
Esfuerzo ideal			9	6	3	0	

Tabla 13: Tareas y puntos de historia Primera Iteración

En las siguientes imágenes, mostramos un resumen de la iteración, un BurnDown Chart con las tareas realizadas:

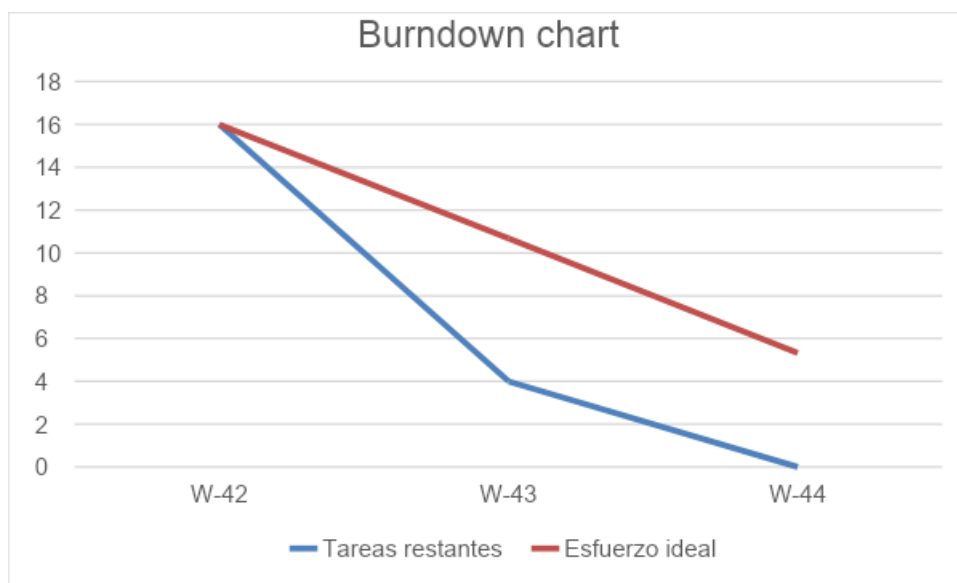


Figura 50: Burndown Chart de la iteración inicial (Gráfico de líneas)

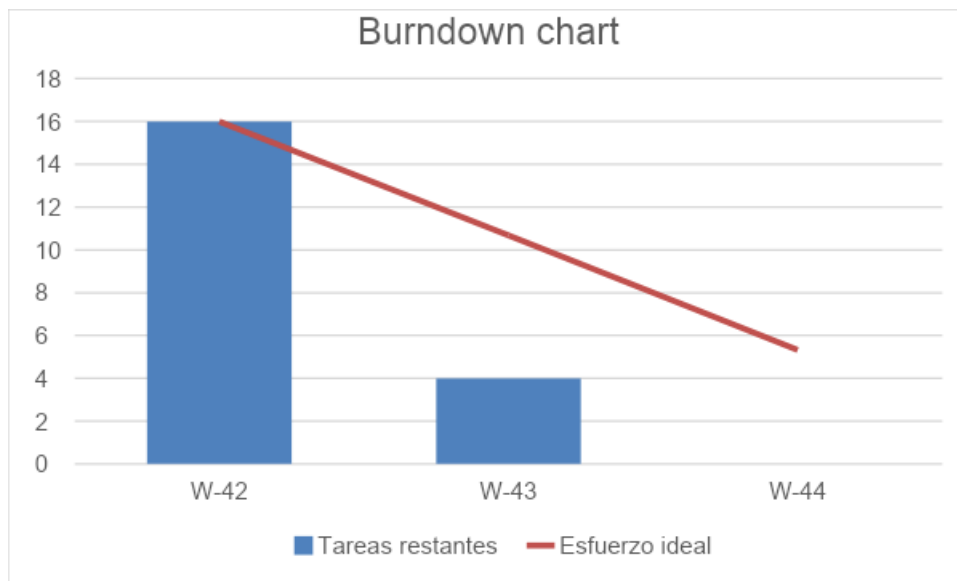


Figura 51: Burndown Chart de la iteración inicial (Gráfico de barras)

3.1.5. Entrega y reunión final de la iteración

Durante la primera iteración, se han implementado todas las funcionalidades planificadas incluidas en las historias, así como algunos retoques adicionales para la creación y gestión de usuarios de nuestra plataforma.

Hemos tenido problemas con la inclusión de la imagen del perfil, debido a que la API que íbamos a usar tenía demasiadas opciones y era demasiado complicada de integrar, por lo que, al valorar la situación y dada la escala de nuestra aplicación, hemos decidido pasar estas imágenes a cadenas de texto, pudiendo guardarlas así en nuestra base de datos y luego convertirla a imagen otra vez al enseñarla en la parte cliente. Esto se hará en la siguiente iteración.

Pese a algunos problemas encontrados, ha sido un desarrollo muy productivo y se han conseguido sentar unas bases, para que, en la siguiente iteración, sea más fructífera aún.

3.2. Segunda iteración

En la primera iteración, nos dedicamos a diseñar la base de nuestra aplicación, la implementación de la base de datos, frontend y backend e implementar la primera entidad de nuestra aplicación, los usuarios.

Una vez finalizada esta parte, comenzaremos con la siguiente entidad, el restaurante, donde debemos añadirle unas funcionalidades adicionales para que puedan ser autosuficientes y sean capaces ellos mismos de gestionar su negocio como ellos deseen. Luego este será el paso siguiente.

3.2.1. Historias de usuario

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US006	Estimación	4	Prioridad	ALTA
Como	Restaurante	Quiero	Crear restaurante	Para	Dar de alta un nuevo negocio
Criterios de validación		Realizar correctamente los pasos a seguir en la creación del restaurante			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 14: Historia de usuario US006: Crear restaurante

Historia de usuario	Nombre del proyecto	Coordinador del proyecto
---------------------	---------------------	--------------------------

		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US007	Estimación	2	Prioridad	ALTA
Como	Restaurante	Quiero	Editar restaurante	Para	Actualizar el sistema con información nueva sobre mi negocio
Criterios de validación		Rellenar correctamente el formulario con los campos necesarios			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 15: Historia de usuario US007: Editar restaurante

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US008	Estimación	2	Prioridad	ALTA
Como	Restaurante	Quiero	Eliminar restaurante	Para	Dar de baja un negocio
Criterios de validación		Realizar los pasos correctamente y confirmar la baja del negocio			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 16: Historia de usuario US008: Eliminar restaurante

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US009	Estimación	3	Prioridad	ALTA

Como	Usuario	Quiero	Buscar restaurante	Para	Encontrar el negocio deseado
Criterios de validación	Escribir correctamente el nombre y que se encuentre en nuestro sistema.				
Redactado por	Francisco Campos	José Reyes	Fecha	01/12/2023	
Aprobado por	Francisco Campos	José Reyes	Fecha	01/12/2023	

Tabla 17: Historia de usuario US009: Buscar restaurante

3.2.2. Análisis

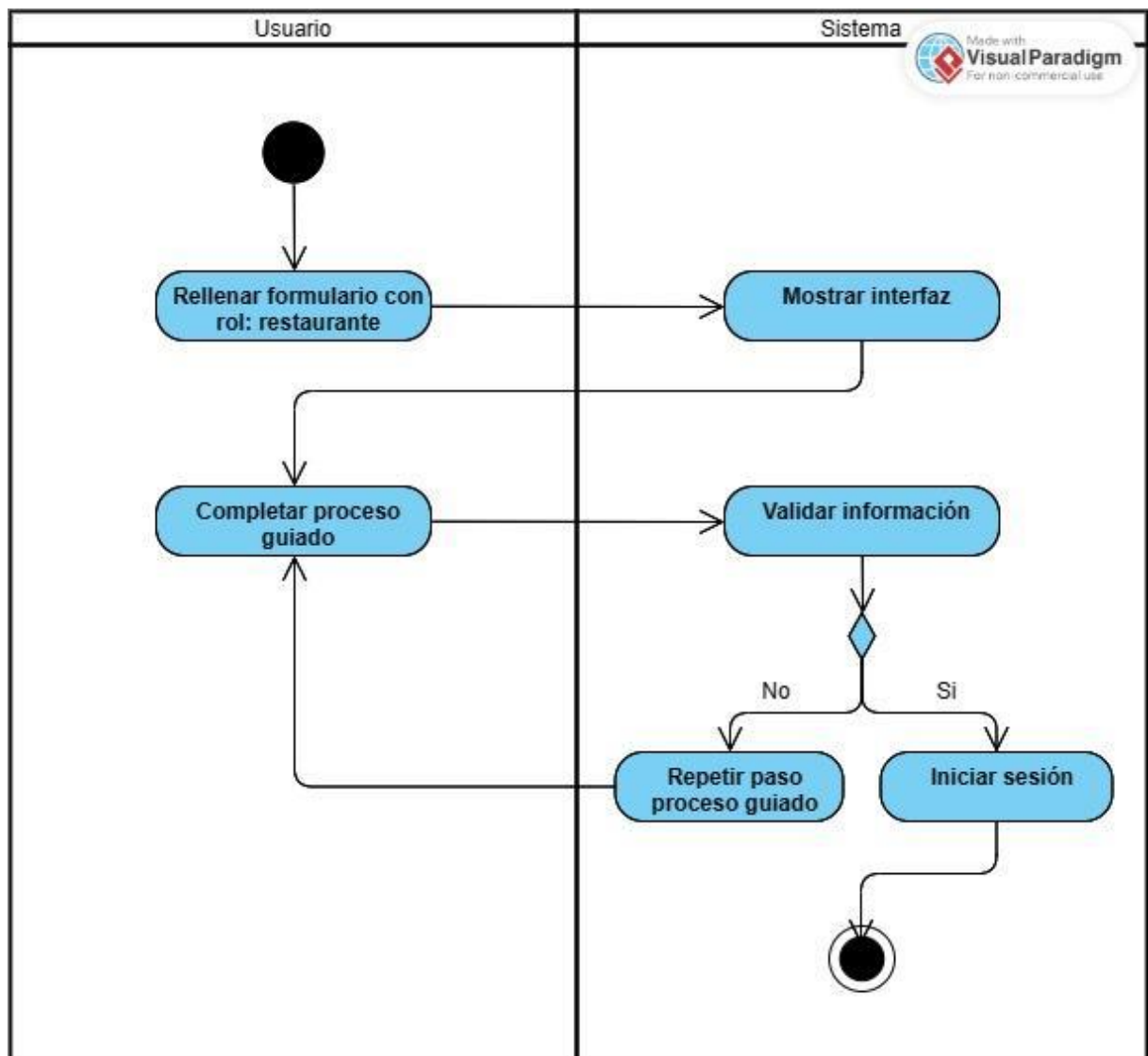


Figura 52: Diagrama de actividad – Historia de usuario US006: Crear restaurante

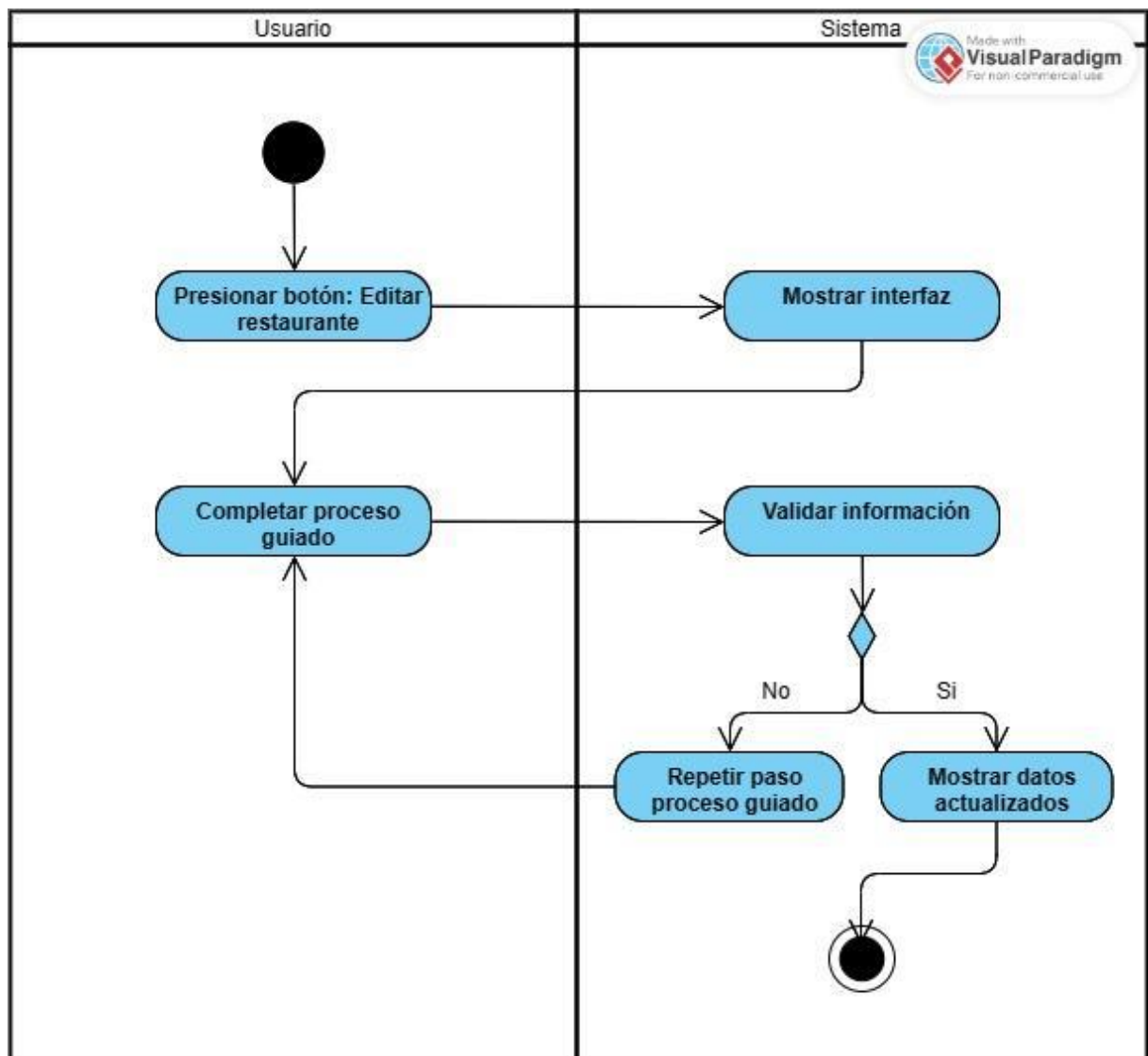


Figura 53: Diagrama de actividad – Historia de usuario US007: Editar restaurante

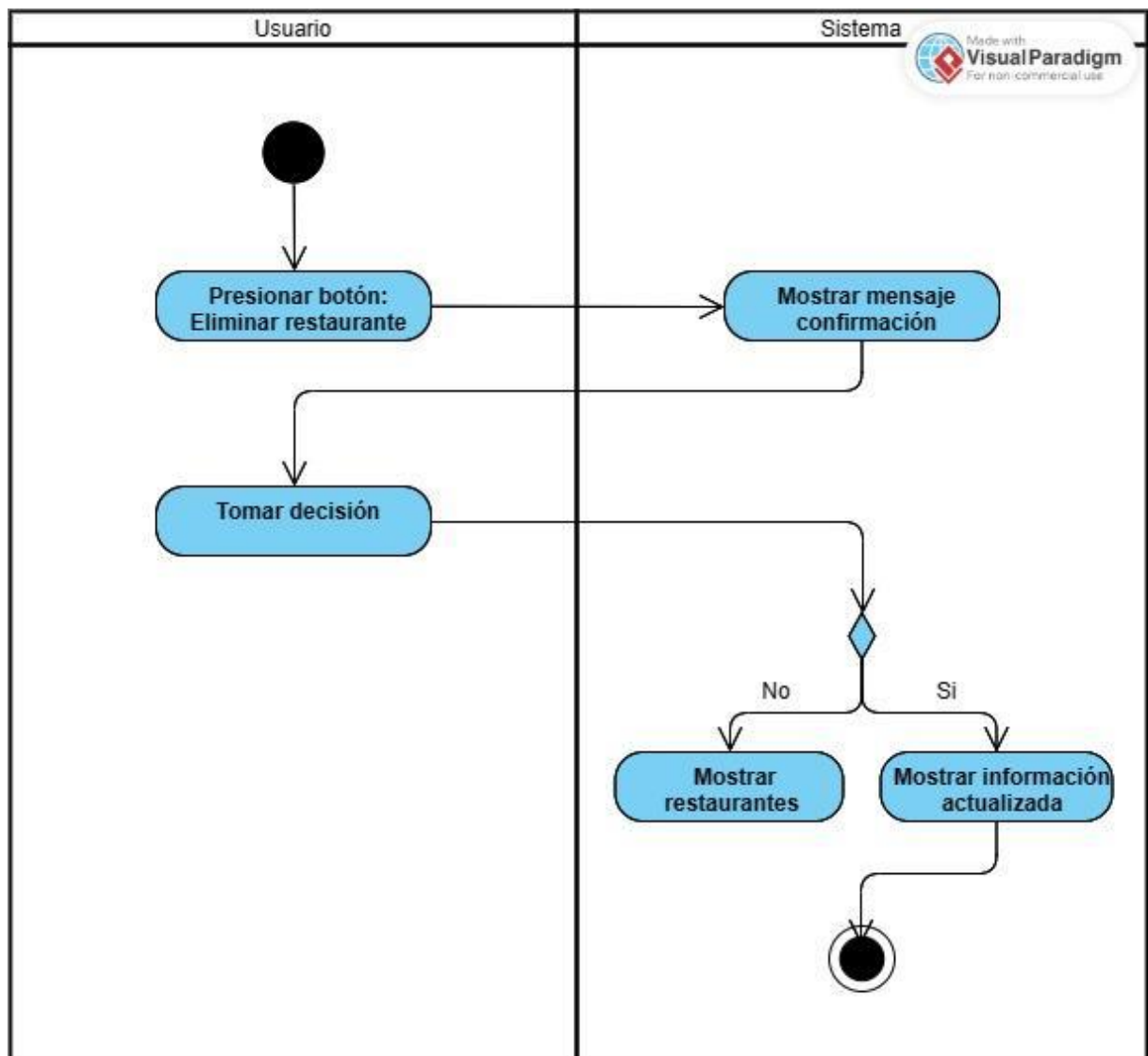


Figura 54: Diagrama de actividad – Historia de usuario US008: Eliminar restaurante

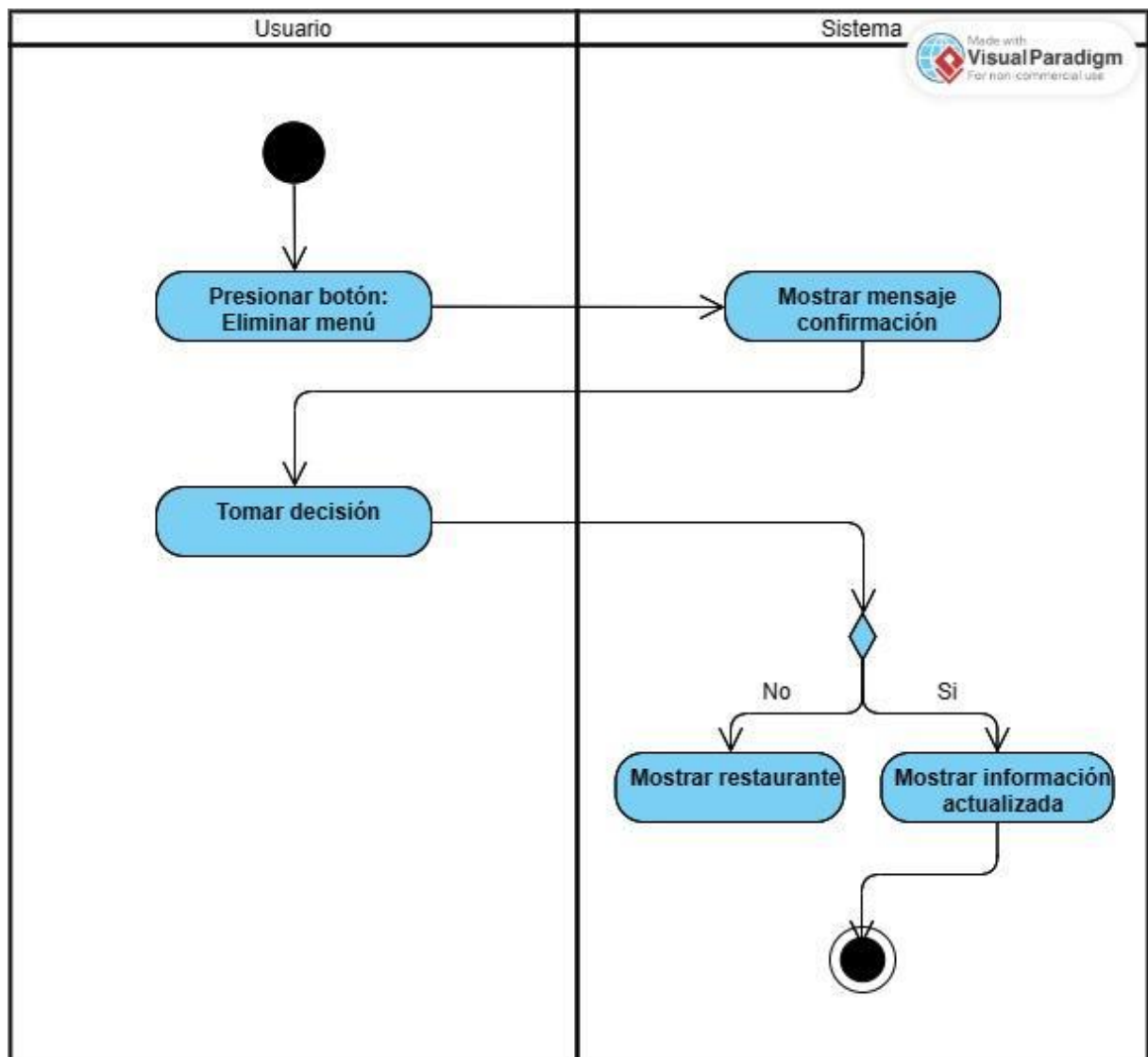


Figura 55: Diagrama de actividad – Historia de usuario US009: Buscar restaurante

3.2.3. Planificación de la iteración

Esta segunda iteración ha sido más difícil a niveles de desarrollo debido a dos factores.

1. Los restaurantes son más complicados a nivel de diseño

2. Hay que ir comprobando que todo lo que hacemos a nivel de restaurante, no tenga ningún problema con el cliente y que la comunicación entre ambos se realice de manera coherente sin pérdidas de información o fallos de seguridad.

A continuación, se muestra la siguiente tabla con el backlog de la segunda iteración:

BACKLOG DE LA SEGUNDA ITERACIÓN	NOMBRE DEL PROYECTO	COORDINADOR DEL PROYECTO
	TFG – LunchAtHome	Francisco José Reyes Campos
Entidad	Restaurante	
ID	Tarea	Proyecto
US006: Crear restaurante		
T0601	Diseño backend crear restaurante	Servidor
T0602	Diseño frontend crear restaurante	Cliente
T0603	Implementación backend crear restaurante	Servidor
T0604	Implementación frontend crear restaurante	Cliente
US007: Editar restaurante		
T0701	Diseño backend editar restaurante	Servidor
T0702	Diseño frontend editar restaurante	Cliente
T0703	Implementación backend editar restaurante	Servidor
T0704	Implementación frontend editar restaurante	Cliente
US008: Eliminar restaurante		
T0801	Diseño backend eliminar restaurante	Servidor
T0802	Diseño frontend eliminar restaurante	Cliente
T0803	Implementación backend eliminar restaurante	Servidor
T0804	Implementación frontend eliminar restaurante	Cliente
US009: Buscar restaurante		
T0901	Diseño backend buscar restaurante	Servidor

T0902	Diseño frontend buscar restaurante	Cliente
T0903	Implementación backend buscar restaurante	Servidor
T0904	Implementación frontend buscar restaurante	Cliente

Tabla 18: Backlog segunda iteración

3.2.4. Diseño

En esta sección, había que tomar muchas decisiones, como los restaurantes se mostrarán a los usuarios, como los usuarios interactuarán con los restaurantes, como mostrarán estos menús a los usuarios y como estos comprarán e interactuarán con los menús tanto para ver su contenido como para realizar un pedido.

Hemos escogido la usabilidad como concepto principal desde donde empezar a tomar decisiones. Si el conocimiento adquirido por el usuario le facilita usar nuestra aplicación, debemos conseguir que las partes de nuestra aplicación sean lo más parecidas posible entre ellas, aportando también eficiencia de código pudiendo reutilizarlo no solamente para lo que se ha pensado originalmente. Por ello, se ha decidido, mostrarle al usuario los restaurantes en tarjetas, como hacen otras páginas de Marketplace de comida a domicilio y luego los menús de la misma manera, tarjetas que muestren su contenido.

Luego el restaurante será quien ponga sus precios y podrá gestionar su negocio de acuerdo a unos términos legales, pero la idea es, que puedan ser totalmente autosuficientes y puedan gestionarse de la manera que quieran.

3.2.4.1. Servidor

En esta sección, se ha llevado a cabo el diseño e implementación de lo que es el restaurante, y como gestionar el mismo, ampliando toda la parte del usuario, ya que un usuario podrá tener a cargo uno o varios restaurantes y las opciones que tendrá

no serán las mismas que un cliente o usuario promedio. Luego en esta parte, se ha encargado de incluir toda la lógica para poder llevarlo a cabo.

3.2.4.2. Cliente

En esta sección, el frontend se encargará de mostrar las vistas del restaurante. El dueño de un negocio podrá ver todos sus negocios, la posibilidad de gestionarlo como quiera, ya sea cambiando información o dar de alta o baja otro que tenga.

- Componentes comunes
 - Header
 - Footer
 - Barra de búsqueda
- Componentes
 - Tarjeta de información del restaurante
 - Botón de editar restaurante
 - Botón de eliminar restaurante
 - Botón de gestionar restaurante
- Routing Module:
 - AuthContext
- Modelos
 - Cliente
 - Restaurante
- Enrutamiento (React Router)
 - Rutas públicas
 - Rutas privadas

3.2.5. Resumen del progreso

Durante esta segunda iteración del proyecto, se ha enfocado en mejorar lo que teníamos anteriormente, mejorar en eficiencia de código y reutilización del mismo además de añadir los cambios que teníamos pensados anteriormente con respecto a las imágenes.

En cuanto al grueso del desarrollo, no ha habido grandes problemas más allá de algunos generados por el navegador debido a lo exigentes que son actualmente, pero para las pruebas no han existido muchas complicaciones.

Se ha conseguido seguir las historias de usuario definidas anteriormente y por tanto hemos tenido un progreso fluido y continuo y esta segunda iteración ha sido exitosa y han sido establecidas las bases para el desarrollo de nuevas funcionalidades en el futuro. La estimación de tiempo se cumplió en su totalidad.

A continuación, veremos cómo ha progresado la iteración con ayuda de unos gráficos:

BACKLOG DE LA SEGUNDA ITERACIÓN	NOMBRE DEL PROYECTO	COORDINADOR DEL PROYECTO					
	TFG – LunchAtHome	Francisco José Reyes Campos					
Entidad	Restaurante						
ID	Tarea	Proyecto	Puntos de historia por tareas	Primera iteración (SEMANALES)			
Inicio del proyecto				16/10	23/10	30/10	Total de horas por tarea
				W42	W43	W44	
US006: Crear restaurante							
T0601	Diseño backend crear restaurante	Servidor	1	1			
T0602	Diseño frontend crear restaurante	Cliente	1	1			

T0603	Implementación backend crear restaurante	Servidor	1	1			
T0604	Implementación frontend crear restaurante	Cliente	1	1			
US007: Editar restaurante							
T0701	Diseño backend editar restaurante	Servidor	1		1		
T0702	Diseño frontend editar restaurante	Cliente	1		1		
T0703	Implementación backend editar restaurante	Servidor	1		1		
T0704	Implementación frontend editar restaurante	Cliente	1		1		
US008: Eliminar restaurante							
T0801	Diseño backend eliminar restaurante	Servidor	1		1		100%
T0802	Diseño frontend eliminar restaurante	Cliente	1		1		100%
T0803	Implementación backend eliminar restaurante	Servidor	1		1		100%
T0804	Implementación frontend eliminar restaurante	Cliente	1		1		100%
US009: Buscar restaurante							
T0901	Diseño backend buscar restaurante	Servidor	1			1	100%
T0902	Diseño frontend buscar restaurante	Cliente	1			1	100%
T0903	Implementación backend buscar restaurante	Servidor	1			1	100%
T0904	Implementación frontend buscar restaurante	Cliente	1			1	100%
Tareas restantes			16	12	4	0	
Esfuerzo ideal			16	10,67	5,33	0	
Puntos de historia restantes			11	8	5	0	
Esfuerzo ideal			11	7,34	3,68	0	

Tabla 19: Tareas y puntos de historia Segunda Iteración

En las siguientes imágenes, mostramos un resumen de la iteración, un BurnDown Chart con las tareas realizadas:

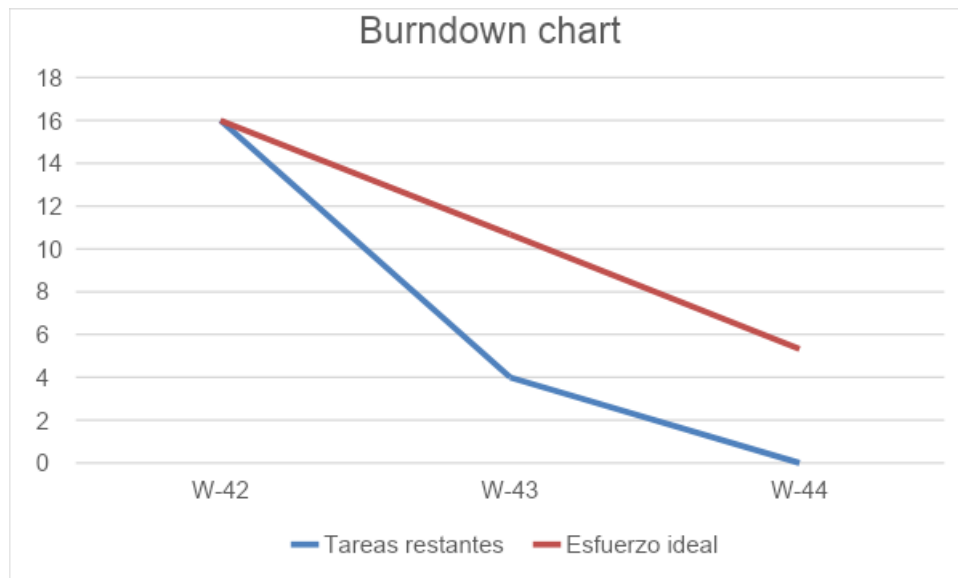


Figura 56: Burndown Chart de la segunda iteración (Gráfico de líneas)

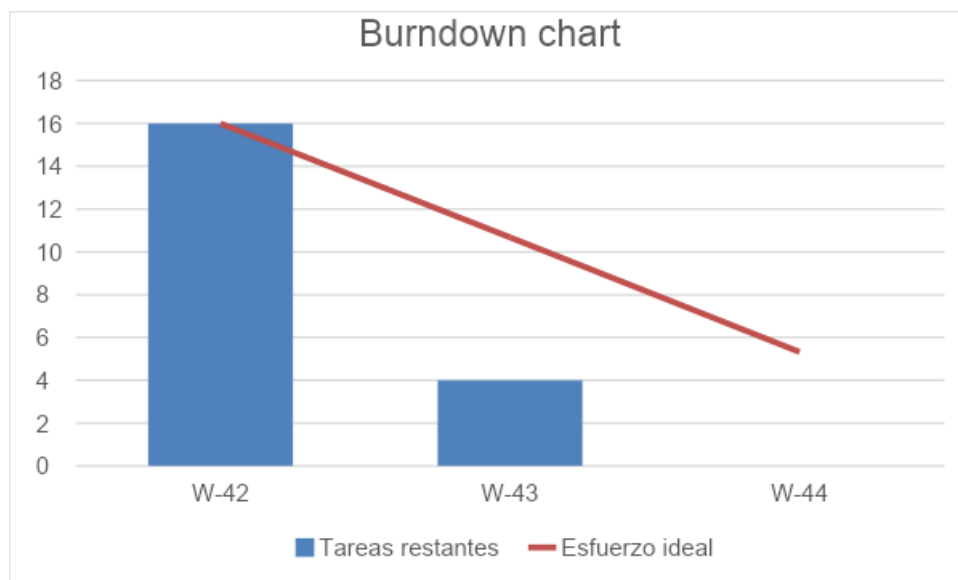


Figura 57: Burndown Chart de la segunda iteración (Gráfico de barras)

3.2.6. Entrega y reunión de la iteración.

Para esta segunda entrega, se ha implementado en su totalidad todas las funcionalidades pensadas al principio, así como algunas mejoras de lo que ya teníamos. Se ha mejorado la experiencia visual del usuario y ha sido integrada totalmente la interfaz del restaurante.

A pesar de los problemas técnicos que mencioné anteriormente respecto a los navegadores, no se ha sufrido ningún retraso y se ha podido avanzar correctamente.

Para la tercera iteración, al igual que ocurrió con las imágenes, hemos tenido la idea de que los usuarios puedan buscar restaurantes según la demanda nutricional que necesitan, por tanto, en la última etapa también intentaremos mejorar la eficiencia de búsqueda tanto de restaurantes como de menús.

3.3. Tercera iteración

En esta última iteración, implementaremos la última entidad, los menús de los restaurantes. Esta parte será más llevadera que las dos anteriores, debido a que todas las comunicaciones entre el cliente y el restaurante están ya acabadas, ya tenemos experiencia con nuestras tecnologías y la relación entre el restaurante y el menú no son tan complicadas a nivel de diseño como las anteriores.

En esta parte también, el enfoque se dirigirá hacia la implementación de medidas de seguridad para fortalecer el sistema. Se dará especial atención a

garantizar la interacción segura entre el cliente y el restaurante, optimizando la eficiencia en la búsqueda por parte de los usuarios, asegurando a su vez nuestro sistema. Además, se abordarán algunos requisitos extra para los usuarios basados en su ideas y consejos, cómo la integración de un sistema de recomendación en base de las preferencias del usuario. La mayor carga de trabajo la tendrá esta segunda parte de esta iteración.

3.3.1. Historias de usuario

4. Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US010	Estimación	3	Prioridad	ALTA
Como	Restaurante	Quiero	Añadir menú	Para	Añadir nuevas comidas a mi restaurante
Criterios de validación		Rellenar correctamente el formulario y completar todos los campos			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 20: Historia de usuario US009: Añadir menú

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US011	Estimación	3	Prioridad	ALTA
Como	Restaurante	Quiero	Editar menú	Para	Actualizar el sistema con información nueva de mi menú
Criterios de validación		Rellenar correctamente el formulario y completar todos los campos			

Redactado por	Francisco José Reyes Campos	Fecha	01/12/2023
Aprobado por	Francisco José Reyes Campos	Fecha	01/12/2023

Tabla 21: Historia de usuario US010: Editar menú

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US012	Estimación	2	Prioridad	ALTA
Como	Restaurante	Quiero	Eliminar menú	Para	Quitar del sistema un menú que ya no está en la carta
Criterios de validación		Realizar los pasos correctamente y confirmar la eliminación del menú			
Redactado por		Francisco José Reyes Campos		Fecha	01/12/2023
Aprobado por		Francisco José Reyes Campos		Fecha	01/12/2023

Tabla 22: Historia de usuario US011: Eliminar menú

Historia de usuario		Nombre del proyecto		Coordinador del proyecto	
		TFG – LunchAtHome		Francisco José Reyes Campos	
ID	US013	Estimación	3	Prioridad	ALTA
Como	Usuario	Quiero	Buscar menú	Para	Realizar una gestión de usuarios
Criterios de validación		Introducir correctamente el nombre del menú si este se encuentra en el restaurante			

Redactado por	Francisco José Reyes Campos	Fecha	01/12/2023
Aprobado por	Francisco José Reyes Campos	Fecha	01/12/2023

Tabla 23: Historia de usuario US011: Buscar menú

3.3.2. Análisis

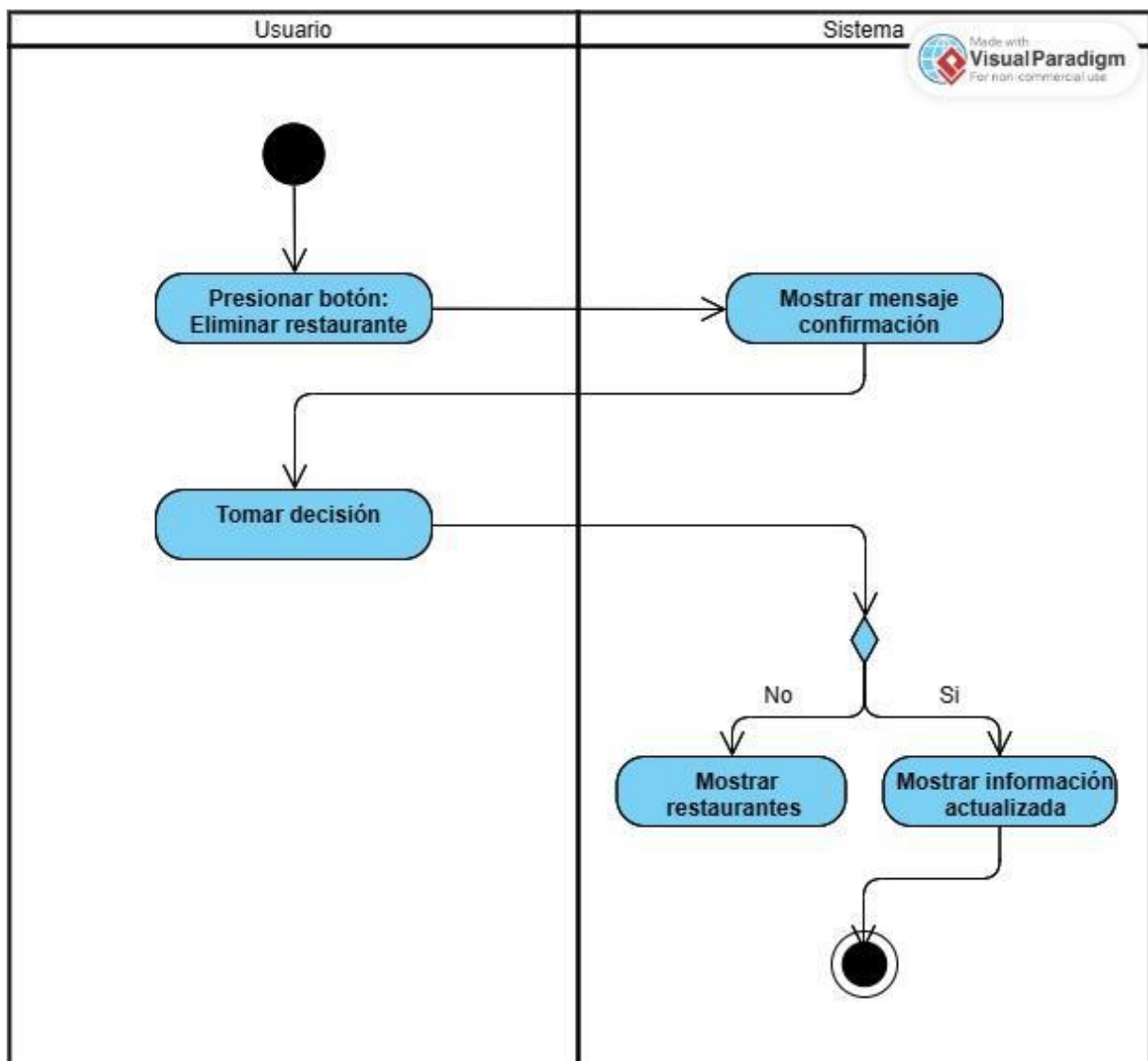
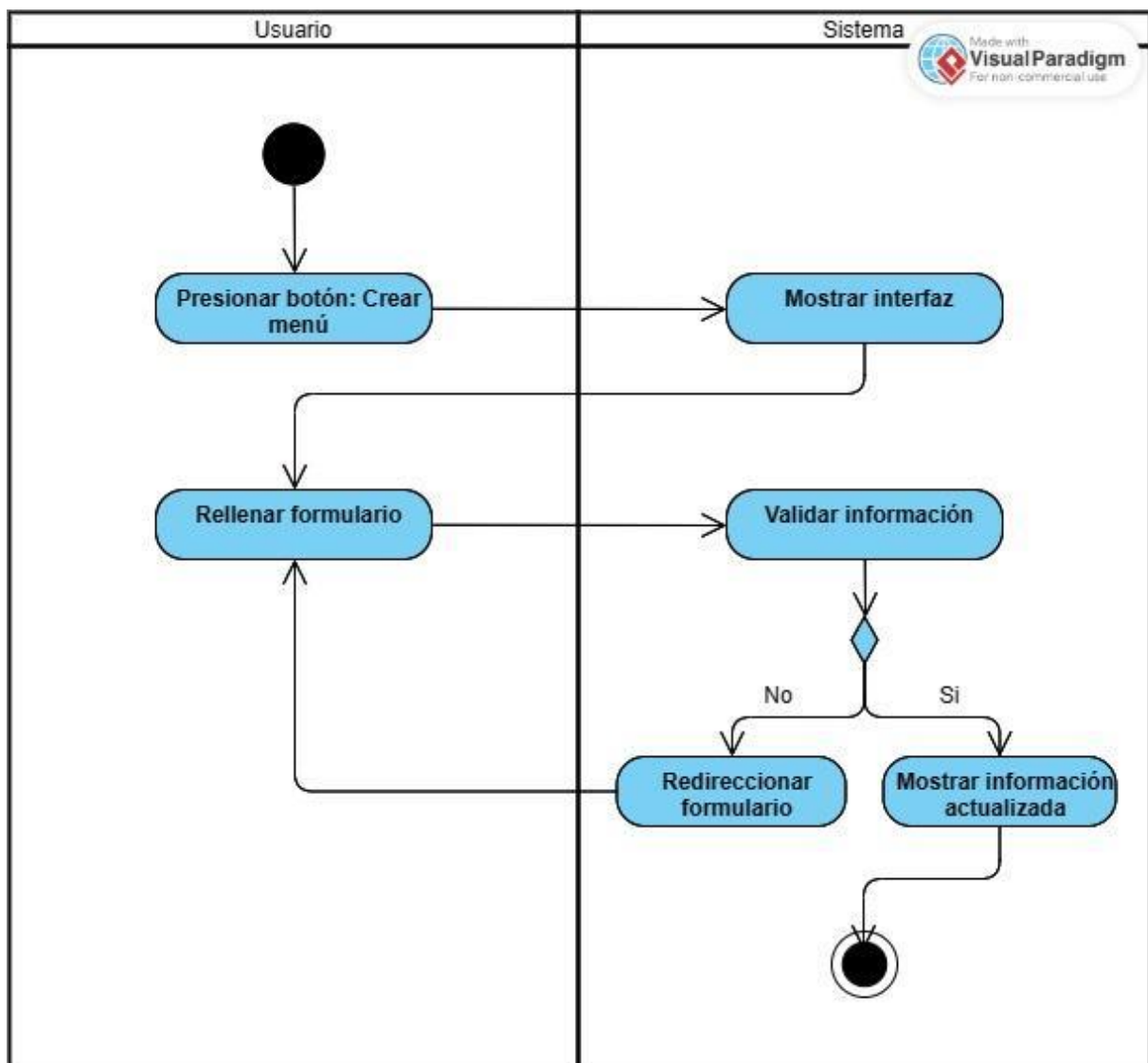


Figura 58: Diagrama de actividad – Historia de usuario US010: Crear menú**Figura 59: Diagrama de actividad – Historia de usuario US011: Editar menú**

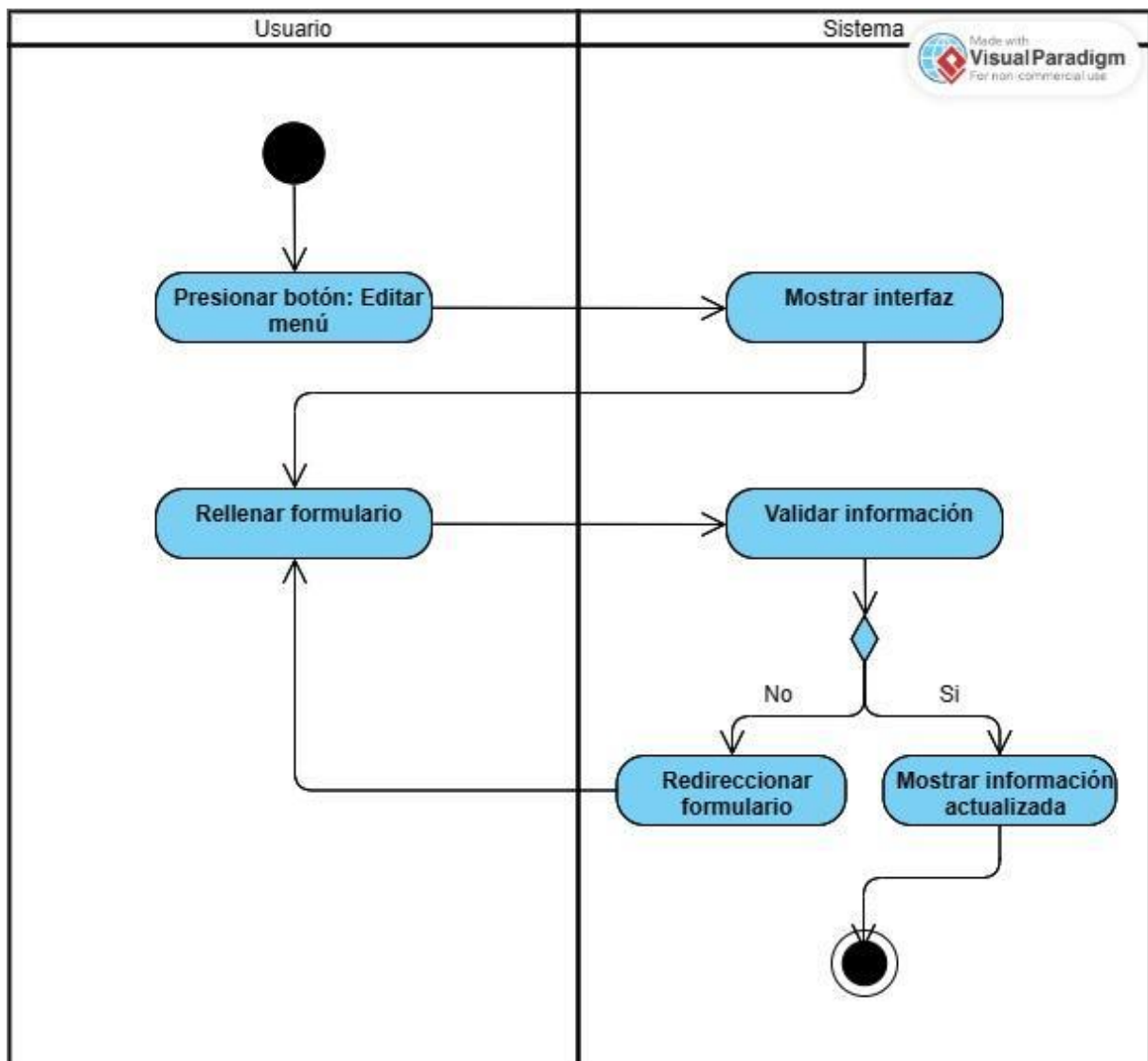


Figura 60: Diagrama de actividad – Historia de usuario US012: Eliminar menú

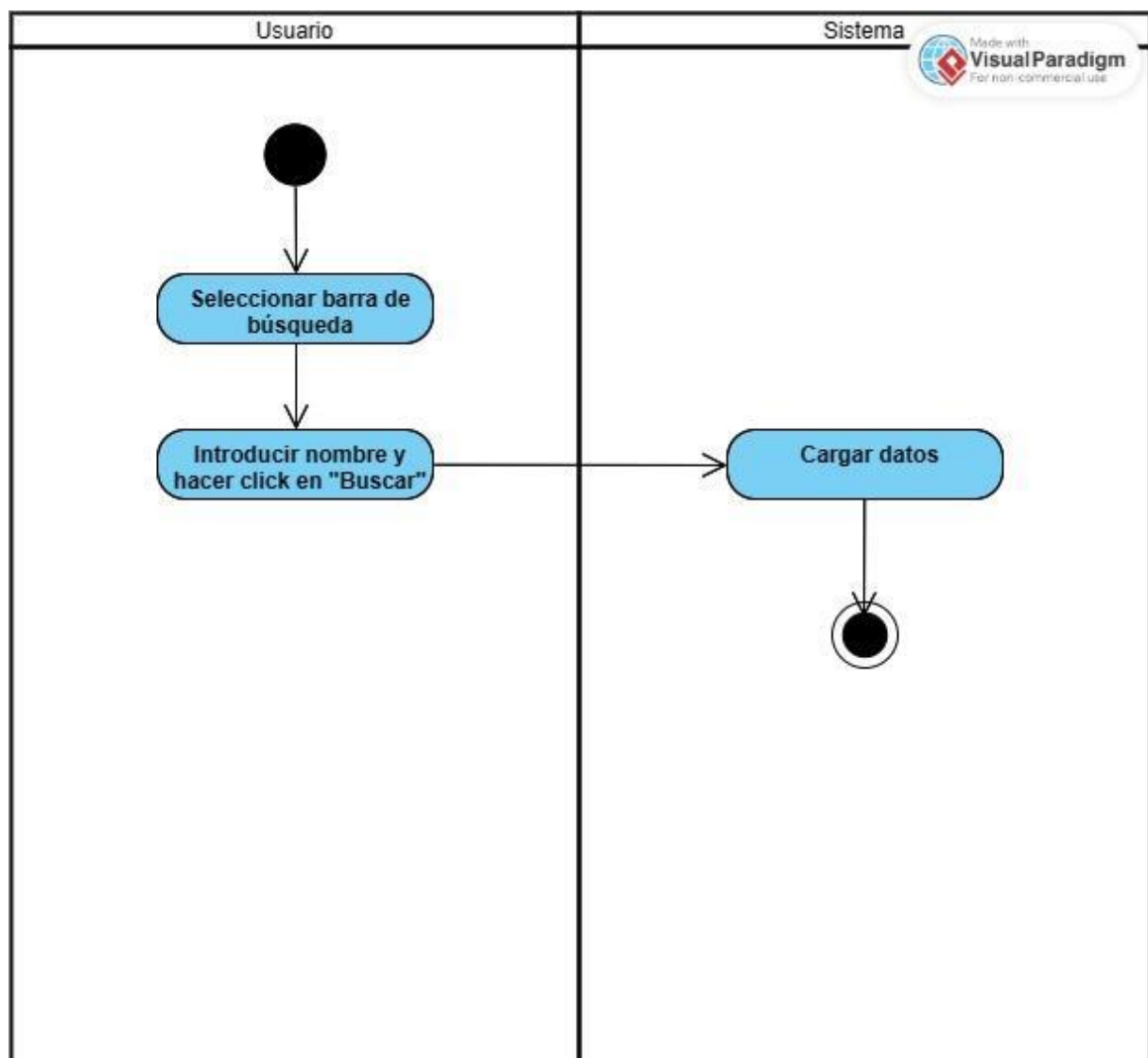


Figura 61: Diagrama de actividad – Historia de usuario US013: Buscar menú

3.3.3. Planificación de la iteración.

BACKLOG DE LA TERCERA ITERACIÓN	NOMBRE DEL PROYECTO	COORDINADOR DEL PROYECTO
	TFG – LunchAtHome	Francisco José Reyes Campos
Entidad	Menú	

ID	Tarea	Proyecto
US010: Crear menú		
T1001	Diseño backend crear menú	Servidor
T1002	Diseño frontend crear menú	Cliente
T1003	Implementación backend crear menú	Servidor
T1004	Implementación frontend crear menú	Cliente
US011: Editar menú		
T1101	Diseño backend editar menú	Servidor
T1102	Diseño frontend editar menú	Cliente
T1103	Implementación backend editar menú	Servidor
T1104	Implementación frontend editar menú	Cliente
US012: Eliminar menú		
T1201	Diseño backend eliminar menú	Servidor
T1202	Diseño frontend eliminar menú	Cliente
T1203	Implementación backend eliminar menú	Servidor
T1204	Implementación frontend eliminar menú	Cliente
US013: Buscar menú		
T1301	Diseño backend buscar menú	Servidor
T1302	Diseño frontend buscar menú	Cliente
T1303	Implementación backend buscar menú	Servidor
T1304	Implementación frontend buscar menú	Cliente
Final del proyecto		
T0007	Medidas de seguridad datos del usuario	
T0009	Medidas de seguridad sesión del usuario	
T0010	Medidas de seguridad mediante reglas ACL	
T0011	Normas de estilo	

Tabla 24: Backlog tercera iteración

3.3.4. Diseño

Esta última iteración, ha sido la más fácil de manejar tanto por la parte frontend como backend, debido a la familiaridad que hay con respecto al software y la tecnología que estamos usando. Esta última hemos implementado el menú como última entidad de nuestro proyecto.

No existen restricciones adicionales salvo que los menús se puedan ver sus aportes nutricionales, luego solamente habrá que construir una vista adicional para ello.

3.3.4.1. Servidor

En esta sección se ha desarrollado toda la lógica que tiene la entidad menú en nuestro proyecto además de todas las historias de usuario detalladas anteriormente. Esta parte tiene menos carga de trabajo, pero es igual de importante ya que serán los objetos que compren los usuarios y por tanto no puede tener ningún error.

3.3.4.2. Cliente

En cuanto a la visualización de las vistas los menús, como hemos explicado anteriormente serán parecidas a los restaurantes salvo elementos cruciales de los mismos. En este caso podremos ver la información nutricional en una nueva vista y añadir al carro el menú que queramos pedir. Por tanto, hemos maximizado la usabilidad y la hemos adaptado a los nuevos requisitos que tiene nuestra nueva entidad de una manera sencilla y elegante.

- Componentes comunes

- Header
- Footer
- Barra de búsqueda
- Componentes
 - Tarjeta de información del menú
 - Botón de información menú
 - Botón de añadir al carrito
 - Botón de editar menú (Restaurante)
 - Botón de añadir/eliminar menú (Restaurante)
- Routing Module:
 - AuthContext
- Modelos
 - Cliente
 - Restaurante
 - Menú
- Enrutamiento (React Router)
 - Rutas públicas
 - Rutas privadas

3.3.5. Resumen del progreso

Como último resumen de nuestro desarrollo, se han incluido todas las historias de usuario, así como las adicionales con respecto a la seguridad de nuestra aplicación. Cabe mencionar que se han intentado implementar algunas medidas de seguridad más, como el certificado HTTPS, pero debido a que te lo tiene que conceder una entidad externa o concederla tú mismo, hemos considerado que no procedía y lo hemos tenido desistir con la idea.

Por otro lado, la idea que teníamos en la segunda iteración de incluir en las búsquedas por aportes nutricionales se ha incluido correctamente y los

usuarios tienen una manera distinta de buscar los menús o restaurantes que más aporte nutricional en específico tengan.

No se han detectado muchos problemas en el desarrollo más allá de los que suelen surgir cuando se está programando. Además, se han llevado a cabo pruebas exhaustivas en cuanto a la seguridad y no ha surgido ningún inconveniente. También hemos solucionado los errores que hemos encontrado antes de la entrega.

Se considerarán nuevas funcionalidades en el futuro, como la opción de recuperar la contraseña o asignar identificadores únicos a los usuarios en lugar de sus emails, aunque por ahora, debido a la escala de nuestro proyecto, no es necesario.

A continuación, se mostrará una tabla con el progreso de esta iteración:

BACKLOG DE LA TERCERA ITERACIÓN	NOMBRE DEL PROYECTO	COORDINADOR DEL PROYECTO					
	TFG – LunchAtHome	Francisco José Reyes Campos					
Entidad	Menú						
ID	Tarea	Proyecto	Puntos de historia por tareas	Tercera iteración (SEMANALES)			
Inicio del proyecto				06/11	13/11	20/11	Total de horas por tarea
				W45	W46	W47	
US006: Crear menú							
T1001	Diseño backend crear menú	Servidor	1	1			
T1002	Diseño frontend crear menú	Cliente	1	1			

T1003	Implementación backend crear menú	Servidor	1	1			
T1004	Implementación frontend crear menú	Cliente	1	1			
US007: Editar menú							
T1101	Diseño backend editar menú	Servidor	1	1			
T1102	Diseño frontend editar menú	Cliente	1	1			
T1103	Implementación backend editar menú	Servidor	1	1			
T1104	Implementación frontend editar menú	Cliente	1	1			
US008: Eliminar menú							
T1201	Diseño backend eliminar menú	Servidor	1	1			100%
T1202	Diseño frontend eliminar menú	Cliente	1	1			100%
T1203	Implementación backend eliminar menú	Servidor	1	1			100%
T1204	Implementación frontend eliminar menú	Cliente	1	1			100%
US009: Buscar menú							
T1301	Diseño backend buscar menú	Servidor	1		1	1	100%
T1302	Diseño frontend buscar menú	Cliente	1		1	1	100%
T1303	Implementación backend buscar menú	Servidor	1		1	1	100%
T1304	Implementación frontend buscar menú	Cliente	1		1	1	100%
Tareas restantes			16	4	0	0	
Esfuerzo ideal			16	10,67	5,33	0	
Puntos de historia restantes			9	9	0	0	
Esfuerzo ideal			9	6	3	0	

Tabla 25: Tareas y puntos de historia tercera iteración

En las siguientes imágenes, mostramos un resumen de la iteración, un BurnDown Chart con las tareas realizadas:

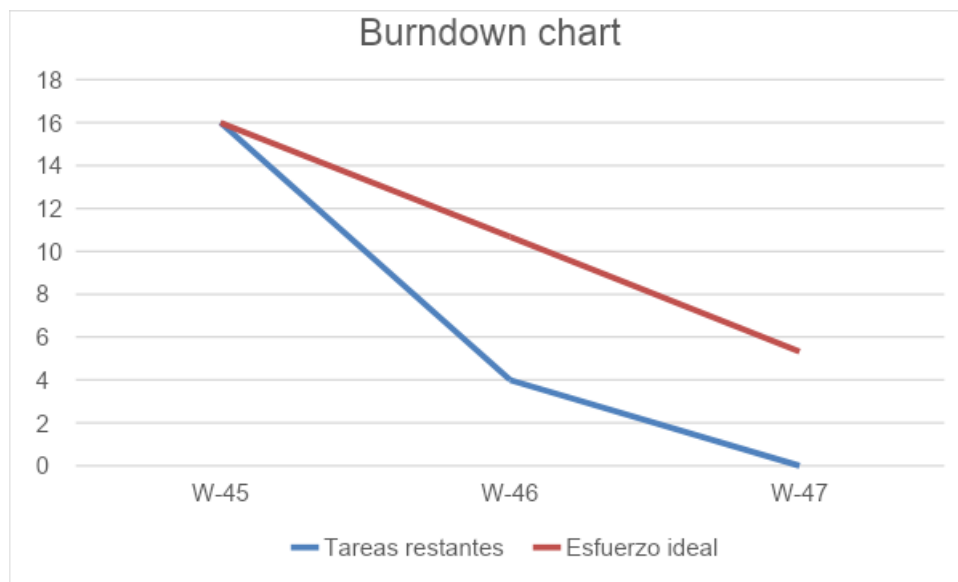


Figura 62: Burndown Chart de la tercera iteración (Gráfico de líneas)

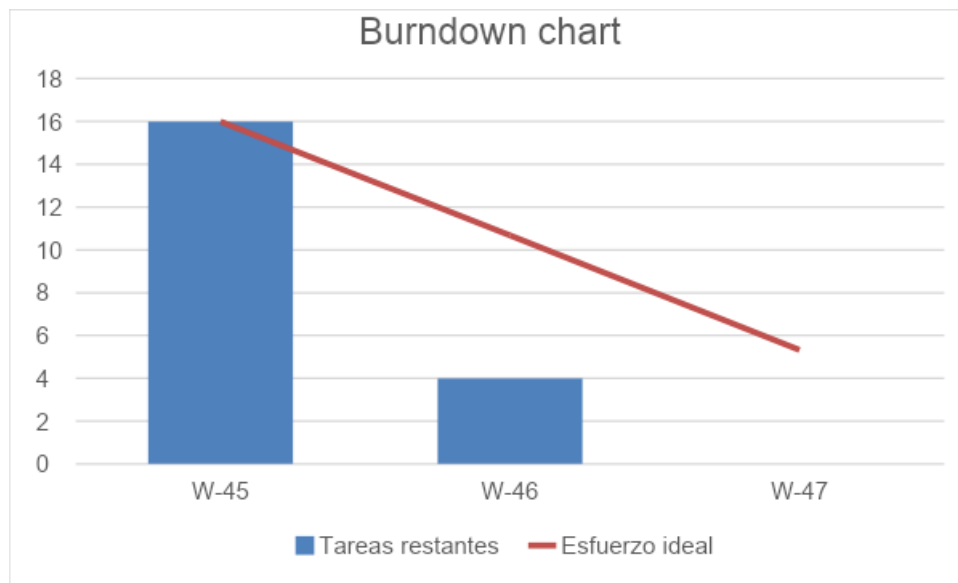


Figura 63: Burndown Chart de la tercera iteración (Gráfico de barras)

3.3.6. Entrega final de la iteración.

Hemos realizado la entrega al usuario de la aplicación funcional con el requisito fundamental de la aplicación: capacidad de pedir comida a domicilio con toda la información importante a mano.

Nosotros como desarrolladores hemos quedado muy contentos y el usuario como no podía ser menos, también, y muy agradecido nos comentó algunas cosas que podría mejorar la aplicación:

- Mejorar de alguna manera el sistema de recomendación. Aquí hemos pensado en usar un algoritmo de recomendación o ingeniería de características.

- Introducir eventos en las páginas de ofertas con tiempo limitado que cambien en función del usuario
- Algún modo para recordar las credenciales de usuario para no tener que estar introduciéndose en todo momento.

En resumen, nuestro proyecto ha sido un éxito y hemos conseguido cumplir con los objetivos planteados en la memoria del proyecto. Adoptamos una metodología ágil, realizando entregas periódicas para garantizar un desarrollo continuo para asegurar que se cumplieran con las metas propuestas.

3.4. Pruebas de usabilidad

Para este apartado hemos realizado encuestas, contestadas por 20 personas, para evaluar la usabilidad de la aplicación. Las preguntas abordan los aspectos que hemos considerado más importantes desde la facilidad de uso hasta la experiencia de usuario.

Los resultados de estas encuestas nos han dejado satisfechos y nos han permitido encontrar áreas donde mejorar nuestra aplicación, información obtenida por parte de los usuarios. Las áreas y preguntas en las que nos hemos enfocado han sido las siguientes:

- **Facilidad de uso**

¿Consideras que la página es fácil de usar?

20 respuestas

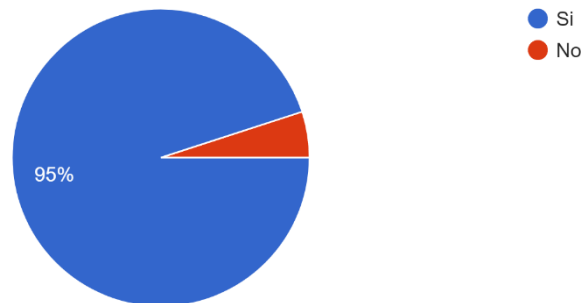


Figura 64: Pregunta 1 encuesta usabilidad

¿Te resultó sencillo navegar por la página web?

20 respuestas

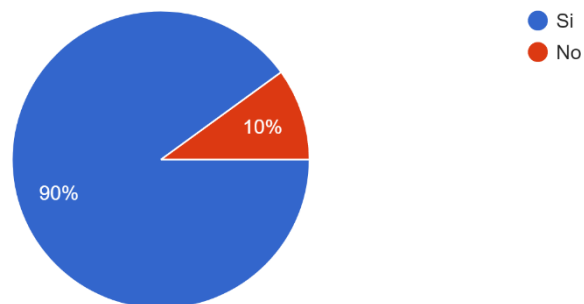


Figura 65: Pregunta 2 encuesta usabilidad

- **Funcionalidad**

¿Todas las funciones de la aplicación son útiles para ti?

20 respuestas

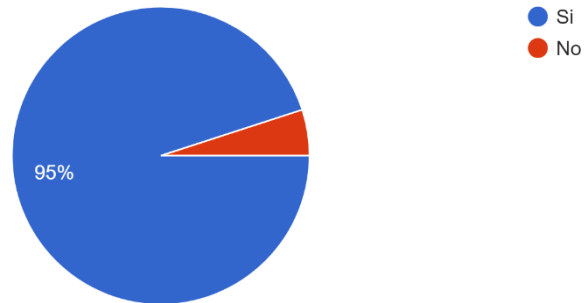


Figura 66: Pregunta 3 encuesta usabilidad

¿Experimentaste algún problema técnico o error mientras usabas la aplicación?

20 respuestas

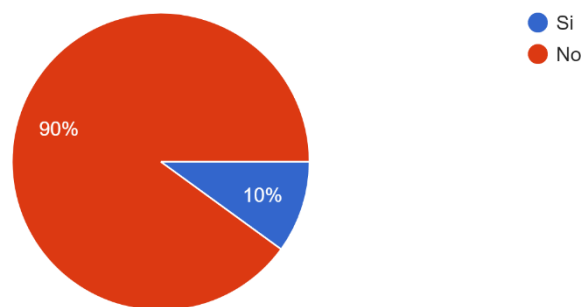


Figura 67: Pregunta 4 encuesta usabilidad

- **Experiencia de usuario**

¿Cómo calificarías la aplicación?

20 respuestas

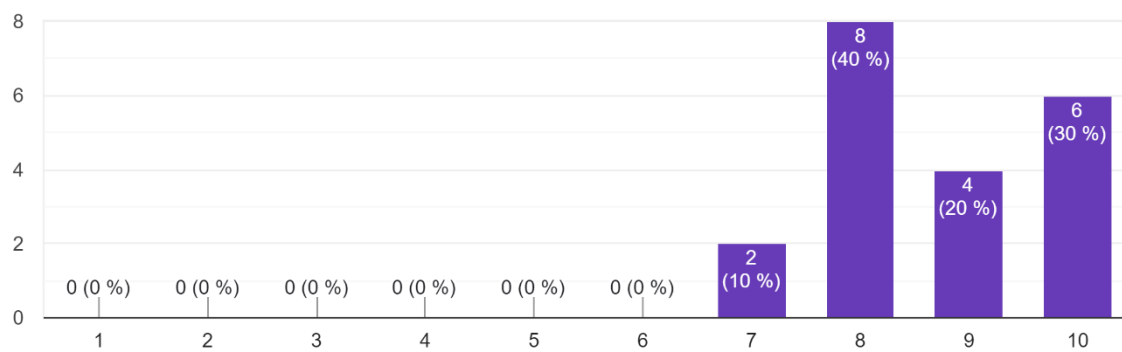


Figura 68: Pregunta 5 encuesta de usabilidad

4. Resultados y discusión

En este apartado, vamos a hablar sobre lo que hemos logrado durante la fase del desarrollo del proyecto y las cosas que hemos aprendido, con intención de poder usarlas en futuros proyectos. También vamos a analizar los problemas que hemos encontrado y daremos posibles soluciones, por si nos volvemos a encontrar otra vez con ellos.

4.1. Resultado y resumen final del progreso de aprendizaje

De los problemas que hemos ido encontrando, se han hablado de ellos en los apartados correspondientes a las iteraciones donde fueron encontrados, luego aquí, haremos un breve repaso de cómo fueron encontrados, que hicimos para solucionarlos y por último como fueron adaptados a nuestro proyecto.

Para la primera iteración, nos centramos en construir la base de nuestro proyecto y sus funcionalidades básicas, como son el inicio de sesión, registro de usuarios, así como todo lo relacionado con la entidad Usuario. Para la segunda iteración, se añadieron todas las funcionalidades de la entidad Restaurante, así como todas las relaciones que tiene con la entidad Usuario. Para la tercera iteración, se incluyó la última entidad Menú, así como algunas mejoras de seguridad y calidad de nuestro proyecto.

Dicho esto, durante el proceso se encontraron algunos conflictos en cuanto al diseño y desarrollo de ciertas funcionalidades de nuestras entidades, por lo que, para abordarlas, hicimos lo siguiente: Inicialmente llevamos a cabo una sesión de lluvia de ideas en cuanto a posibles soluciones de nuestro problema, posteriormente empezamos a descartar en función a unos criterios específicos, tales como eficiencia,

costos de desarrollo o tiempos necesarios para implementarlos, con el propósito de filtrar y seleccionar las soluciones más idóneas.

Pongamos el ejemplo de la última iteración de nuestro proyecto y uno de funcionalidades que quisimos implementar, un sistema de recomendación para un usuario. Para ellos tuvimos una lluvia de ideas entre las que se dieron, nuestra solución dada, recopilación de datos del usuario y se dieron algunas más como fueron un algoritmo de recomendación o ingeniería de características. Después de investigar a fondo cada una de las soluciones, las dos últimas eran un problema en cuanto a costes y tiempo de desarrollo de las mismas, por lo que, nos decidimos por la primera, lo cual es menos eficiente, pero muchísimo más asequible para nosotros, y en caso de que en un futuro nuestra aplicación tenga éxito, podríamos implementarlas, que una idea sea desechada en las primeras fases de un proyecto no tiene por qué ser descartada completamente.

Aun así, podemos concluir que hemos logrado alcanzar todos los requisitos del proyecto y satisfacer las necesidades de nuestro cliente. Además, debido a lo fácilmente escalable que es nuestra aplicación, podremos mejorarla y dar actualizaciones a futuro, dando a los clientes una experiencia de usuario satisfactoria.

Elegí participar en este proyecto con el objetivo de profundizar mis conocimientos sobre las tecnologías empleadas, trabajo y experiencia que conlleva realizar una aplicación web. Con seguridad, puedo afirmar que esta experiencia ha enriquecido mis habilidades y conocimientos, dándome capacidad para tomar futuros trabajos de desarrollo de aplicaciones web.

En conclusión, el desarrollo de este proyecto no solo me ha dado una valiosa comprensión de la práctica de las tecnologías usadas y desafíos que puedo encontrar desarrollando una aplicación web, sino que también ha fortalecido mi confianza de manera positiva para futuros proyectos similares.

5. Conclusiones y trabajo futuro

En resumen, este proyecto ha representado un desafío personal en mi búsqueda por ampliar mis conocimientos en el desarrollo de aplicaciones web y el manejo de las tecnologías involucradas. Aunque la disponibilidad de tiempo no ha sido la ideal, la experiencia ha sido enriquecedora y ha contribuido significativamente en el aprendizaje en este ámbito.

En términos de habilidades técnicas, he adquirido experiencia significativa utilizando tecnologías web modernas. Estas son, por un lado, en el lado cliente, se ha llevado a cabo utilizando React.js un potente marco de trabajo JavaScript. Por el otro, del lado servidor, utilicé Flask, un marco de desarrollo web ligero de Python, para gestionar las solicitudes HTTP y proporcionar una estructura eficaz a la aplicación.

En cuanto a habilidades de gestión, ha mejorado mi capacidad para administrar mi tiempo de manera eficiente. Establecí metas claras, definiendo plazos tanto para el desarrollo global del proyecto como para las tareas específicas, lo que me permitió mantener un progreso constante. Además, en base a la experiencia, he aprendido cómo gestionar problemas o sorpresas inesperadas, buscando soluciones pragmáticas que garantizaran el avance continuo del proyecto.

El aprendizaje obtenido no solo ha sido por lo técnico y la gestión que hemos hablado anteriormente, sino que también he aprendido a ser más creativo, crítico, expresarme mejor y ser más disciplinado conmigo mismo. En general, el proyecto ha sido una gran experiencia de aprendizaje que ha permitido adquirir nuevas habilidades y conocimientos que pueden ser aplicados en futuros proyectos.

A modo de conclusión, no estoy seguro si el proyecto continuará, pero me siento orgullo de todo lo que he aprendido a lo largo de estos meses, y si en un futuro trabajo de esto, siento que puedo mejorar mucho y que me queda mucho por

aprender. Echando un vistazo a nuestra aplicación web, se nota que hay mucho por mejorar y se pueden sacar muchos puntos fuertes y débiles.

BIBLIOGRAFÍA Y REFERENCIAS WEB

[1] *¿Qué es React?*

- Recuperado de [https://www.hostinger.es/tutoriales/que-es-reactt: definición, características y funcionamiento \(hostinger.es\)](https://www.hostinger.es/tutoriales/que-es-reactt: definición, características y funcionamiento (hostinger.es)) el día 21/09/2023

[2] *¿Qué es Node.js?*

- Recuperado de <https://openwebinars.net/blog/que-es-nodejs/> el día 21/09/2023

[3] *¿Qué es NPM?*

- Recuperado de <https://www.freecodecamp.org/espanol/news/que-es-npm/> el día 21/09/2023

[4] *Como usar Tailwind*

- Recuperado de <https://tailwindcss.com/> el día 21/09/2023

[5] *¿Qué es JavaScript?*

- Recuperado de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript el día 21/09/2023

[6] *¿Qué es Python?*

- Recuperado <https://aws.amazon.com/es/what-is/python/> el día 21/09/2023

[7] *¿Qué es Flask?*

- Recuperado de <https://openwebinars.net/blog/que-es-flask/> el día 21/09/2023

[8] *Arquitectura de SPA (Single Page Application)*

- Recuperado de https://es.wikipedia.org/wiki/Single-page_application el día 21/09/2023

[9] *Desarrollo Backend*

- Recuperado de <https://flask-restful.readthedocs.io/en/latest/> el día 21/09/2023

[10] *Que es MySQL*

- Recuperado de <https://openwebinars.net/blog/que-es-mysql/> el día 21/09/2023

[11] *¿Qué es una metodología ágil?*

- Recuperado <https://www.redhat.com/es/topics/devops/what-is-agile-methodology> el día 21/09/2023

[12] *Que es Scrum*

- Recuperado de <https://proyectosagiles.org/que-es-scrum/> el día 21/09/2023

[13] *Glovo acapara ya el 25% del mercado de los agregadores de comida a domicilio.*

- Recuperado de <https://www.eleconomista.es/retail-consumo/noticias/11995155/10/22/Glovo-acapara-ya-el-25-del-mercado-de-los-agregadores-de-comida-a-domicilio.html> [lara ya el 25% del mercado de los agregadores de comida a domicilio \(eleconomista.es\)](https://www.eleconomista.es) el día 28/09/2023

[14] *Salario medio para un programador Full Stack en España:*

- Recuperado de <https://es.talent.com/salary?job=full+stack> el día 28/09/2023

[15] *Guía de redacción de casos de uso MADEJA*

- Recuperado de <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416> el día 10/10/2023

[16] *Elaboración de historias de usuario.*

- Recuperado de https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf el día 10/10/2023

Índice de Figuras

<i>Figura 1: Gráfico de consumidores que usan agregadores de comida</i>	8
<i>Figura 2: Gráfico del mercado de agregadores</i>	8
<i>Figura 3: Vista principal de la plataforma web Glovo.</i>	10
<i>Figura 4: Vista personal de la plataforma web Glovo</i>	11
<i>Figura 5: Vista de negocio de la plataforma web Glovo</i>	11
<i>Figura 6: Vista principal de la plataforma web Uber Eats</i>	13
<i>Figura 7: Vista personalizada de la plataforma web Uber Eats</i>	13
<i>Figura 8: Vista de negocio de la plataforma web Uber Eats</i>	14
<i>Figura 9: Vista principal de la plataforma web Just Eats</i>	15
<i>Figura 10: Vista principal de la plataforma web Just Eats</i>	16
<i>Figura 11: Vista de negocio de la plataforma web Just Eats</i>	17
<i>Figura 12: Logo React</i>	20
<i>Figura 13: Logo Node.js</i>	20
<i>Figura 14: Logo Vue.js</i>	21
<i>Figura 15: Logo Angular</i>	22
<i>Figura 16: Logo Python</i>	23
<i>Figura 17: Logo Flask</i>	23
<i>Figura 18: Logo Django</i>	24
<i>Figura 19: Logo Express.js</i>	24
<i>Figura 20: Logo MySQL Workbench</i>	25
<i>Figura 21: Salario programador full stack con un perfil junior</i>	31
<i>Figura 22: Diagrama de Gantt de Lunch At Home</i>	33
<i>Figura 23: Diagrama de clases de Lunch At Home</i>	45
<i>Figura 24: Arquitectura de Lunch At Home</i>	48
<i>Figura 25: Diagrama de flujo del rol Restaurante</i>	50
<i>Figura 26: Diagrama de flujo del rol administrador</i>	51
<i>Figura 27: Diagrama de flujo del rol cliente</i>	51
<i>Figura 28: Diagrama ER de Lunch At Home</i>	71
<i>Figura 29: Diagrama de actividad – Historia de usuario US001: Registrar usuario</i>	75
<i>Figura 30: Diagrama de actividad – Historia de usuario US002: Iniciar sesión</i>	76
<i>Figura 31: Diagrama de actividad – Historia de usuario US003: Modificar usuario</i>	77
<i>Figura 32: Diagrama de actividad – Historia de usuario US004: Eliminar usuario</i>	78
<i>Figura 33: Diagrama de actividad – Historia de usuario US005: Buscar usuario</i>	79
<i>Figura 34: Burndown Chart de la iteración inicial (Gráfico de líneas)</i>	86
<i>Figura 35: Burndown Chart de la iteración inicial (Gráfico de barras)</i>	87
<i>Figura 36: Diagrama de actividad – Historia de usuario US006: Crear restaurante</i>	91
<i>Figura 37: Diagrama de actividad – Historia de usuario US007: Editar restaurante</i>	92
<i>Figura 38: Diagrama de actividad – Historia de usuario US008: Eliminar restaurante</i>	93

<i>Figura 39: Diagrama de actividad – Historia de usuario US009: Buscar restaurante</i>	<i>94</i>
<i>Figura 40: Burndown Chart de la segunda iteración (Gráfico de líneas).....</i>	<i>100</i>
<i>Figura 41: Burndown Chart de la segunda iteración (Gráfico de barras).....</i>	<i>100</i>
<i>Figura 42: Diagrama de actividad – Historia de usuario US010: Crear menú.....</i>	<i>105</i>
<i>Figura 43: Diagrama de actividad – Historia de usuario US011: Editar menú.....</i>	<i>105</i>
<i>Figura 44: Diagrama de actividad – Historia de usuario US012: Eliminar menú</i>	<i>106</i>
<i>Figura 45: Diagrama de actividad – Historia de usuario US013: Buscar menú.....</i>	<i>107</i>
<i>Figura 46: Burndown Chart de la tercera iteración (Gráfico de líneas).....</i>	<i>113</i>
<i>Figura 47: Burndown Chart de la tercera iteración (Gráfico de barras).....</i>	<i>114</i>
<i>Figura 48: Pregunta 1 encuesta usabilidad</i>	<i>116</i>
<i>Figura 49: Pregunta 2 encuesta usabilidad</i>	<i>116</i>
<i>Figura 50: Pregunta 3 encuesta usabilidad</i>	<i>117</i>
<i>Figura 51: Pregunta 4 encuesta usabilidad.....</i>	<i>117</i>
<i>Figura 52: Pregunta 5 encuesta de usabilidad</i>	<i>118</i>
<i>Figura 53: Vista principal de Lunch At Home</i>	<i>131</i>
<i>Figura 54: Vista de registro de Lunch At Home</i>	<i>132</i>
<i>Figura 55: Vista de acceso de Lunch At Home.....</i>	<i>133</i>
<i>Figura 56: Vista principal con acceso correcto de Lunch At Home</i>	<i>134</i>
<i>Figura 57: Buscador de Lunch At Home</i>	<i>135</i>
<i>Figura 58: Vista de cerrar sesión de Lunch At Home</i>	<i>136</i>
<i>Figura 59: Vista de menús de un restaurante de Lunch At Home</i>	<i>137</i>
<i>Figura 60: Vista de la carta de un menú de Lunch At Home</i>	<i>138</i>
<i>Figura 61: Añadir al carrito de Lunch At Home</i>	<i>139</i>
<i>Figura 62: Pasarela de pago de Lunch At Home</i>	<i>140</i>
<i>Figura 63: Confirmación pasarela de pago de Lunch At Home</i>	<i>141</i>
<i>Figura 64: Paso 1 en la creación de restaurante de Lunch At Home</i>	<i>142</i>
<i>Figura 65: Paso 2 en la creación de restaurante de Lunch At Home</i>	<i>142</i>
<i>Figura 66: Edición de restaurante de Lunch At Home</i>	<i>143</i>
<i>Figura 67: Edición de menús de Lunch at Home</i>	<i>144</i>
<i>Figura 68: Formulario de menú de Lunch At Home</i>	<i>145</i>
<i>Figura 69: Eliminar menú en Lunch At Home</i>	<i>146</i>
<i>Figura 70: Vista principal de administrador de Lunch at Home</i>	<i>147</i>
<i>Figura 71: Vista principal de administrador de Lunch at Home</i>	<i>147</i>
<i>Figura 72: Edición de usuario de administrador de Lunch at Home.....</i>	<i>148</i>
<i>Figura 73: Edición de restaurantes de administrador de Lunch at Home</i>	<i>149</i>
<i>Figura 74: Edición de menús de administrador de Lunch at Home</i>	<i>150</i>

Índice de Tablas

<i>Tabla 1: Presupuesto de Lunch At Home</i>	30
<i>Tabla 2: Planificación temporal Lunch At Home</i>	33
<i>Tabla 3: Caso de uso – Registrar usuario</i>	37
<i>Tabla 4: Caso de uso – Iniciar sesión</i>	38
<i>Tabla 5: Caso de uso – Buscar un producto</i>	39
<i>Tabla 6: Caso de uso – Gestionar un restaurante</i>	40
<i>Tabla 7: Caso de uso – Realizar compra</i>	41
<i>Tabla 8: Historia de usuario US002: Inicio sesión del usuario</i>	73
<i>Tabla 9: Historia de usuario US003: Modificación usuario</i>	73
<i>Tabla 10: Historia de usuario US004: Eliminar usuario</i>	74
<i>Tabla 11: Historia de usuario US005: Búsqueda de usuario</i>	74
<i>Tabla 12: Backlog primera iteración</i>	82
<i>Tabla 13: Tareas y puntos de historia Primera Iteración</i>	86
<i>Tabla 14: Historia de usuario US006: Crear restaurante</i>	88
<i>Tabla 15: Historia de usuario US007: Editar restaurante</i>	89
<i>Tabla 16: Historia de usuario US008: Eliminar restaurante</i>	89
<i>Tabla 17: Historia de usuario US009: Buscar restaurante</i>	90
<i>Tabla 18: Backlog segunda iteración</i>	96
<i>Tabla 19: Tareas y puntos de historia Segunda Iteración</i>	99
<i>Tabla 20: Historia de usuario US009: Añadir menú</i>	102
<i>Tabla 21: Historia de usuario US010: Editar menú</i>	103
<i>Tabla 22: Historia de usuario US011: Eliminar menú</i>	103
<i>Tabla 23: Historia de usuario US011: Buscar menú</i>	104
<i>Tabla 24: Backlog tercera iteración</i>	108
<i>Tabla 25: Tareas y puntos de historia tercera iteración</i>	112

ANEXOS

Anexo 1: Manual de instalación y ejecución de la aplicación

Este anexo de instalación está pensado para la instalación y desarrollo del código en un sistema operativo Windows. Será necesaria conexión a internet para su puesta en marcha.

1. Instalación de cliente

Necesitaremos instalar NodeJS para poder utilizar y ejecutar nuestra aplicación en local.

1.1. Instalación de NodeJS

2. Lo primero que debemos hacer es instalar NodeJS desde su página oficial, su versión recomendada y actualizada de la misma [Node.js \(https://nodejs.org/en/nodejs.org\)](https://nodejs.org/en/nodejs.org). Seguimos los pasos que el instalador nos indique por defecto y pasamos al siguiente paso.
3. Nos dirigiremos a una línea de comandos del sistema y escribiremos lo siguiente: “npm install -g yarn”.

2. Instalación del servidor

Necesitaremos la instalación de Python y Flask, que es la librería en el desarrollo del backend de nuestra aplicación.

2.1. Instalación de Python

1. En la página oficial de Python podremos descargar la última versión [Downlo https://www.python.org/downloads/ad_Python | Python.org](https://www.python.org/downloads/ad_Python | Python.org) y seleccionamos la

versión de nuestro sistema operativo si tenemos 32 o 64 bits. Seguimos los pasos que marca el instalador por defecto.

2. Para comprobar que se ha instalado correctamente, nos iremos a una línea de comandos y escribiremos en la misma: “Python” y nos aparecerá para poder introducir comandos de Python

En caso de no funcionar, recomiendo ir a la Microsoft Store e instalar [Pyth](https://apps.microsoft.com/detail/9PJPW5LDXLZ5?hl=en-US&gl=USon)
<https://apps.microsoft.com/detail/9PJPW5LDXLZ5?hl=en-US&gl=USon> 3.10 -
[Microsoft Apps](#). Directamente configurará las variables de entorno en nuestro sistema.

2.2. Instalación de Flask

1. Nos dirigiremos a una línea de comandos del sistema y escribiremos lo siguiente: “pip install Flask”. En el caso que necesitemos ser administradores, ejecutaremos “sudo pip install Flask”.
2. A continuación, instalaremos las dependencias necesarias. Recomiendo usar un fichero de texto con todas ellas y a continuación usar “pip install -r nombreFichero.txt”. En el caso que necesitemos ser administradores, ejecutaremos “sudo pip install -r nombreFichero.txt”.

Las dependencias son las siguientes:

- Flask==2.0.1
- Flask-Cors==3.0.10
- mysql-connector-python==8.0.26
- PyJWT==2.1.0
- bcrypt==3.2.0

El fichero quedaría de la siguiente manera:

3. Instalación de la base de Datos

1. Nos dirigiremos a la página oficial de [MySQL :: MySQL](https://www.mysql.com/products/workbench/) <https://www.mysql.com/products/workbench/> y nos descargaremos el programa.
2. Seguimos los pasos de instalación que nos diga por defecto.
3. Configuramos nuestra base de datos. Creamos una nueva conexión, con valores por defecto. Importante poner una contraseña segura y de la cual no nos olvidemos. Será con la que entremos a ver los datos que hay en nuestra base de datos.

Ejecución de la aplicación

En la carpeta del proyecto nos encontraremos los siguientes archivos:

Para la puesta en marcha basta con abrir dos consolas de comandos.

Abrimos la primera consola de comandos:

1. cd ruta/Proyecto\react-flask-app
2. yarn start

Abrimos la segunda consola de comandos:

1. cd ruta/Proyecto\react-flask-app/app
2. python app.py

Como nota importante, no se debe cambiar el nombre de las carpetas dentro del proyecto ni cambiar de ubicación ningún archivo, pues se podría perder el funcionamiento de ejecución de la aplicación.

Si todo ha salido bien, en el navegador (si no se ha abierto automáticamente) mediante <http://localhost:3000> deberíamos poder acceder a la aplicación web de manera correcta.

Anexo 2: Manual de usuario

Este anexo presenta el manual del usuario para hacer uso de las distintas funcionalidades de la aplicación.

Instrucciones: Registrarse

Una vez tenemos nuestra aplicación en marcha nos debe aparecer la página principal de la aplicación.

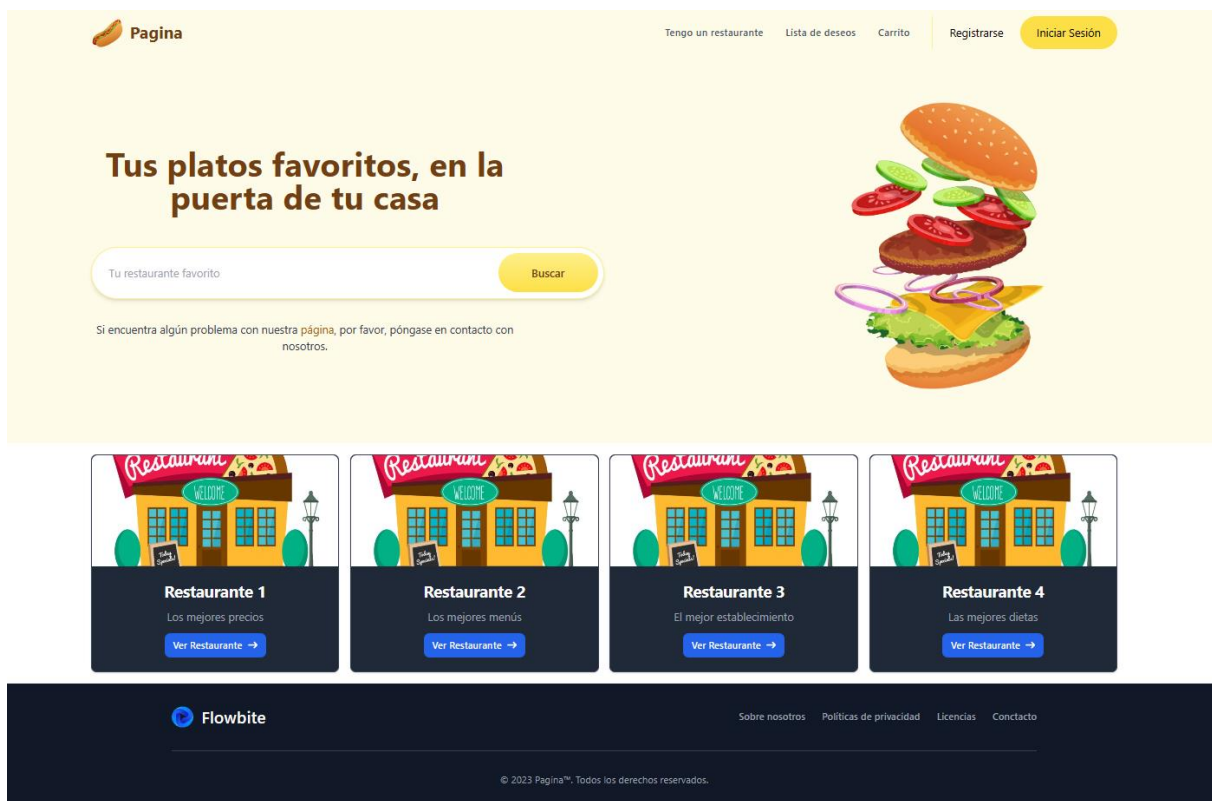


Figura 69: Vista principal de Lunch At Home

Debemos registrarnos para poder acceder a ella. Por ello nos situaremos en el botón de “Registrarse”.



Nos aparecerá el siguiente formulario:

The image shows a registration form titled 'Create and account' from Flowbite. The form is set against a dark background. It includes the following elements: a 'Your email' input field with the placeholder 'name@company.com'; a dropdown menu for '¿Eres cliente o restaurante?' with 'Cliente' selected; 'Password' and 'Confirm password' input fields, both masked with dots; another dropdown menu for '¿Que aporte nutricional buscas?' with 'Muchas/pocas calorías' selected; a checkbox for 'I accept the Terms and Conditions'; a 'Create an account' button; and a link at the bottom that says 'Already have an account? Login here'.

Figura 70: Vista de registro de Lunch At Home

Todos los campos que vemos serán obligatorios. Es importante escribir bien el email pues no se podrá modificar salvo que en un futuro cambies los identificadores únicos de los usuarios. Por último, volveremos a la pantalla principal, que será igual que la que vemos en primera instancia cuando entramos a nuestra aplicación web, con la diferencia de que ahora en una esquina aparecerá que tenemos iniciada sesión.

Instrucciones: Iniciar Sesión

Igual que el apartado anterior, los pasos serán casi similares. En primer lugar, iremos a la pestaña “Iniciar sesión”.



Lo que nos llevará al siguiente formulario:

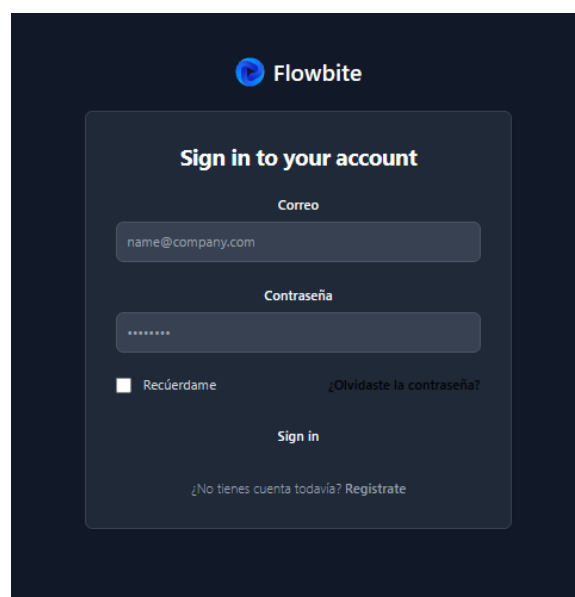
The image shows a login form from Flowbite. At the top is the Flowbite logo. Below it is the title 'Sign in to your account'. There are two input fields: 'Correo' (Email) with the placeholder 'name@company.com' and 'Contraseña' (Password) with a masked input '*****'. Below the password field are two links: 'Recuérdame' (Remember me) with a checkbox and '¿Olvidaste la contraseña?' (Forgot your password?). At the bottom is a 'Sign in' button and a link '¿No tienes cuenta todavía? Regístrate' (Don't have an account yet? Register).

Figura 71: Vista de acceso de Lunch At Home

Deberemos completar los campos correctamente, y si el usuario ya se encuentra en el sistema podrá le saldrá la pantalla que comentamos anteriormente, con los botones cambiados pudiendo cerrar sesión cuando queramos.

Instrucciones: Sesión iniciada

Una vez tenemos la sesión iniciada se verá de la siguiente manera:

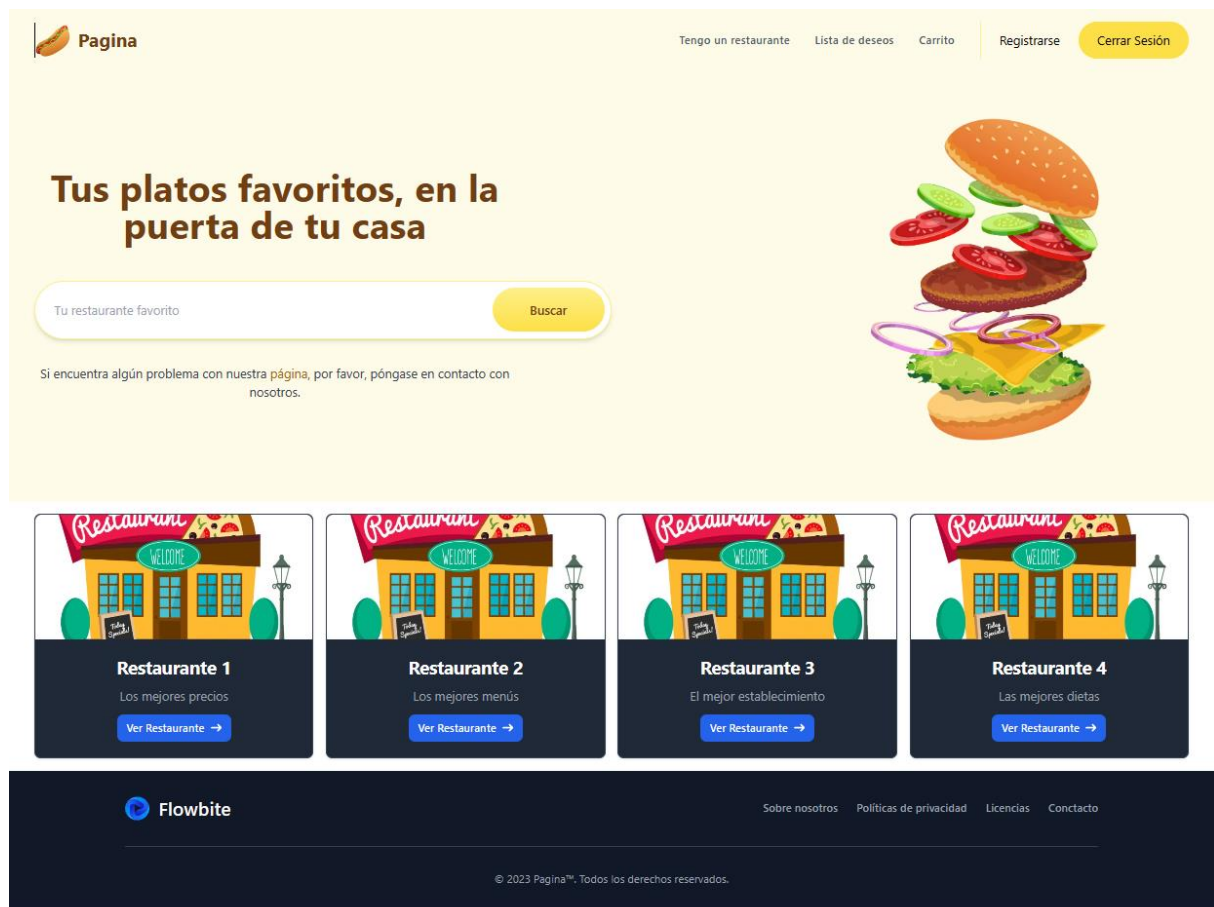


Figura 72: Vista principal con acceso correcto de Lunch At Home

Aquí podremos hacer lo siguiente:

- Ver restaurantes: Tanto si hemos buscado como si no lo hemos hecho, aparecerán restaurantes en base a nuestras preferencias. Podremos ver la información que ofrece si presionamos “Ver Restaurante”. En la imagen anterior podemos ver como se ve inicialmente.
- Barra buscadora: Si conocemos el nombre del restaurante que buscamos, podremos directamente teclear el nombre y presionar sobre el botón “Buscar”, lo que nos devolverá los nombres que estén compuestos por esa cadena.

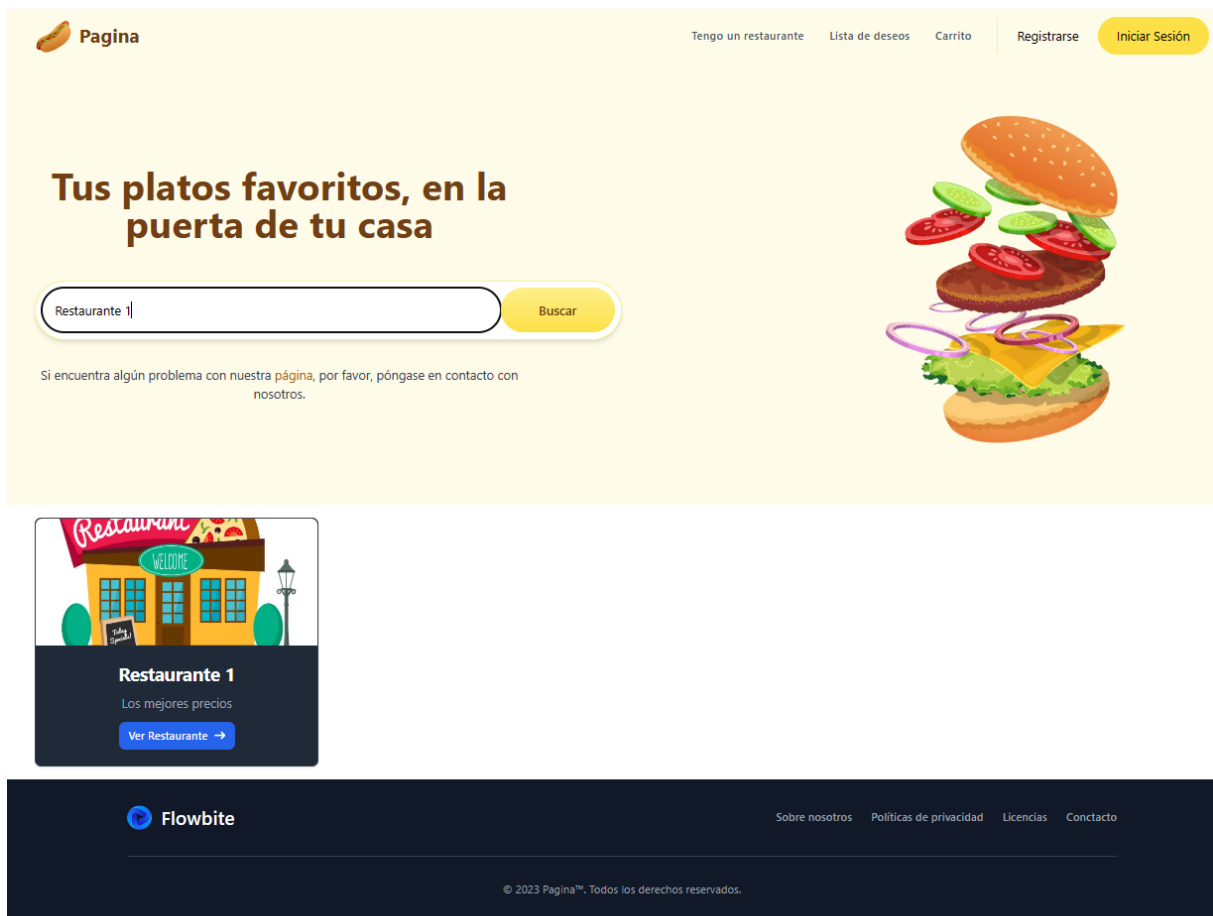


Figura 73: Buscador de Lunch At Home

- Cerrar sesión: Si decidimos cerrar nuestra sesión presionando el botón “Cerrar sesión” aparecerá un mensaje de confirmación.

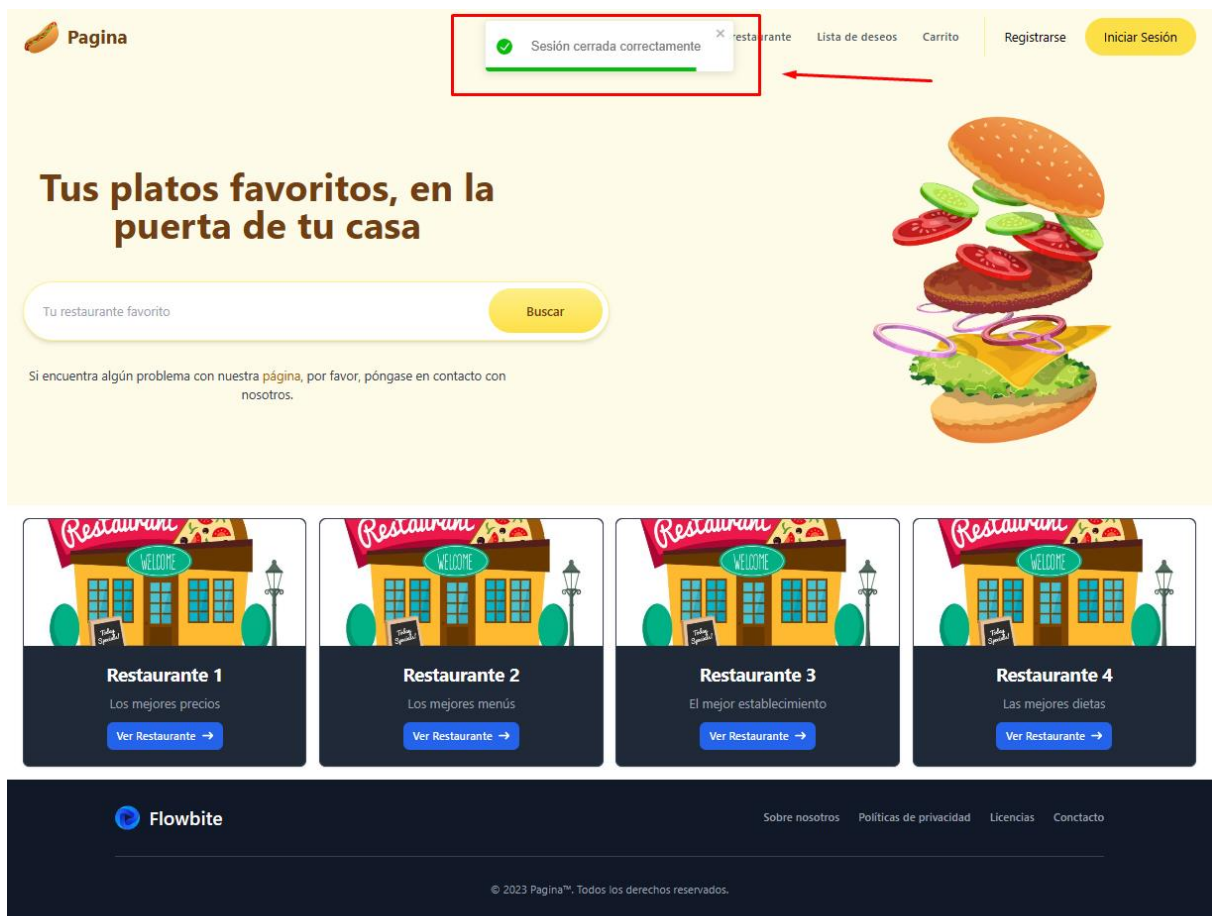


Figura 74: Vista de cerrar sesión de Lunch At Home

Instrucciones: Ver información del menú

Una vez hemos entrado con éxito a nuestra plataforma, si queremos ver un menú para ello debemos entrar al restaurante que lo contiene. Para ello nos vamos al restaurante y presionamos en “Ver Restaurante”. Esto nos dirigirá a la siguiente vista.

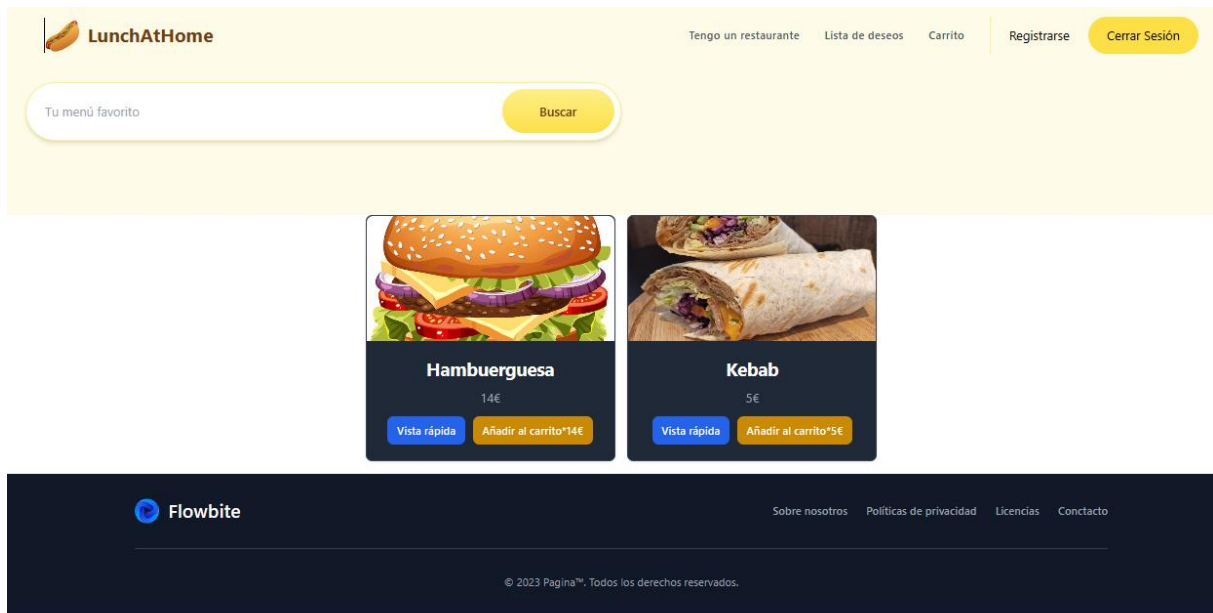


Figura 75: Vista de menús de un restaurante de Lunch At Home

Ahora veremos todos los menús que contiene el restaurante. Cada menú tiene dos botones asociados. El primero de ellos y en el que tenemos interés, tiene puesto “Vista rápida” donde podremos ver toda la información asociada al menú.

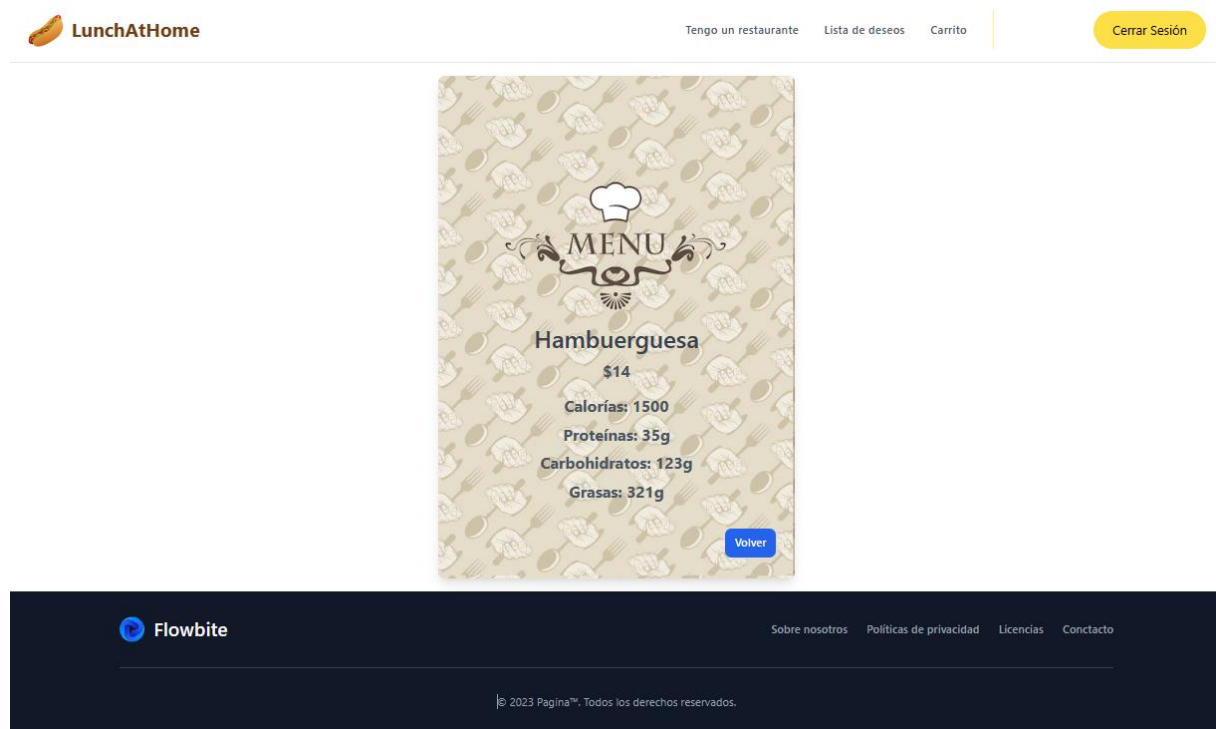


Figura 76: Vista de la carta de un menú de Lunch At Home

A continuación, podremos volver a la página anterior si presionamos en “Volver”.

Instrucciones: Comprar menú

Este paso contiene los mismos pasos iniciales que las instrucciones anteriores. Entramos a nuestra página ya continuación presionamos el botón “Ver Restaurante” del que deseamos ver sus menús. A continuación, si deseamos pedir algún menú presionaremos en el botón “Añadir al carrito”, dando como resultado una actualización del sistema del que el usuario será informado.

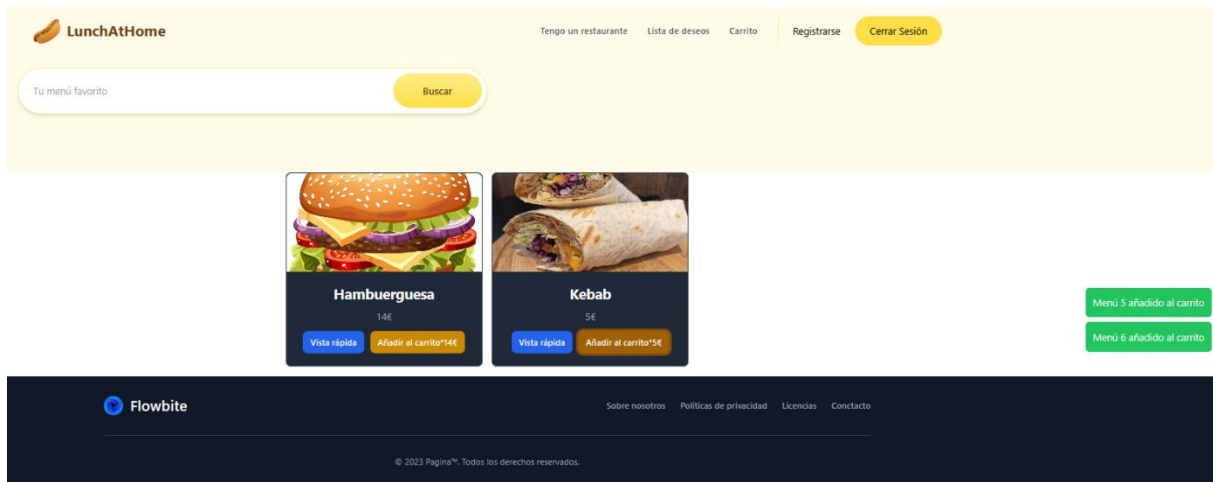


Figura 77: Añadir al carrito de Lunch At Home

Una vez hemos añadido todos los menús que deseamos pedir, nos dirigiremos al botón “Carrito” donde tendremos un resumen de nuestro pedido y unos datos a introducir para terminar de realizar el pedido.

 **LunchAtHome**

Tengo un restauranteLista de deseosCarrito

Cerrar Sesión

Tramitar pedido

1. Dirección de envío

Calle Fuente Nueva N°14, Canena Jaén 23420

Cambiar

2. Método de pago

Visa Electrón XXXX-XXXX-XXXX-XXXX

Cambiar

3. Código de Descuento

Cambiar

4. Revisión del Pedido

Resumen del pedido...

Cambiar

Ver Carrito

 **Flowbite**

Sobre nosotrosPolíticas de privacidadLicenciasContacto

© 2023 Pagina™. Todos los derechos reservados

Figura 78: Pasarela de pago de Lunch At Home

Aquí introduciremos todos los datos necesarios. Si presionamos en ver carrito podremos ver un resumen de lo pedido:

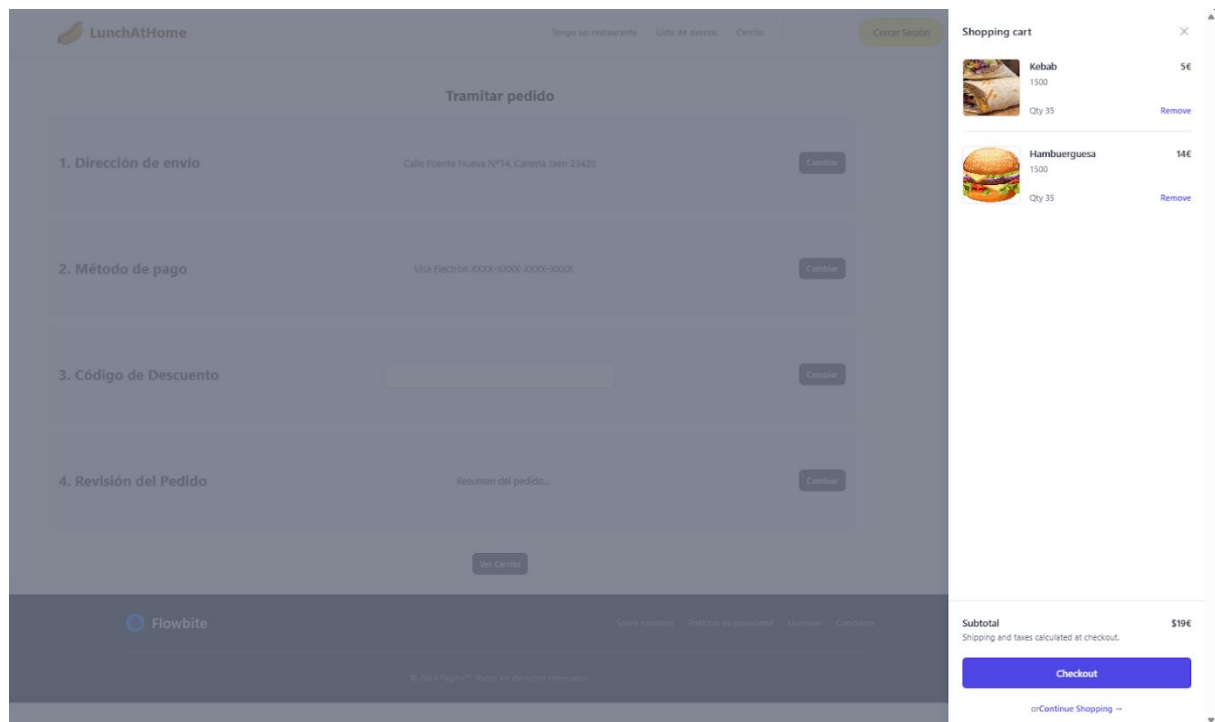


Figura 79: Confirmación pasarela de pago de Lunch At Home

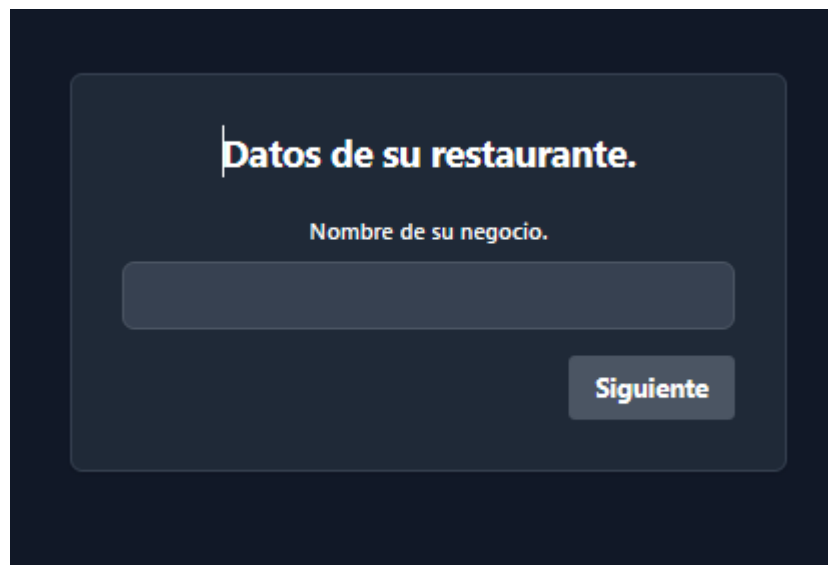
A continuación, se accedería a la pasarela de pago.

Instrucciones: Crear restaurante

Si en el proceso de creación del usuario, definimos nuestro rol como “Restaurante” tendremos una serie de pasos adicionales a seguir.

A continuación, tendremos un proceso guiado en el que podremos introducir toda la información relativa al restaurante. Es un paso sencillo en el que solamente introduciremos una imagen del restaurante y un nombre.

Paso 1 del proceso guiado:



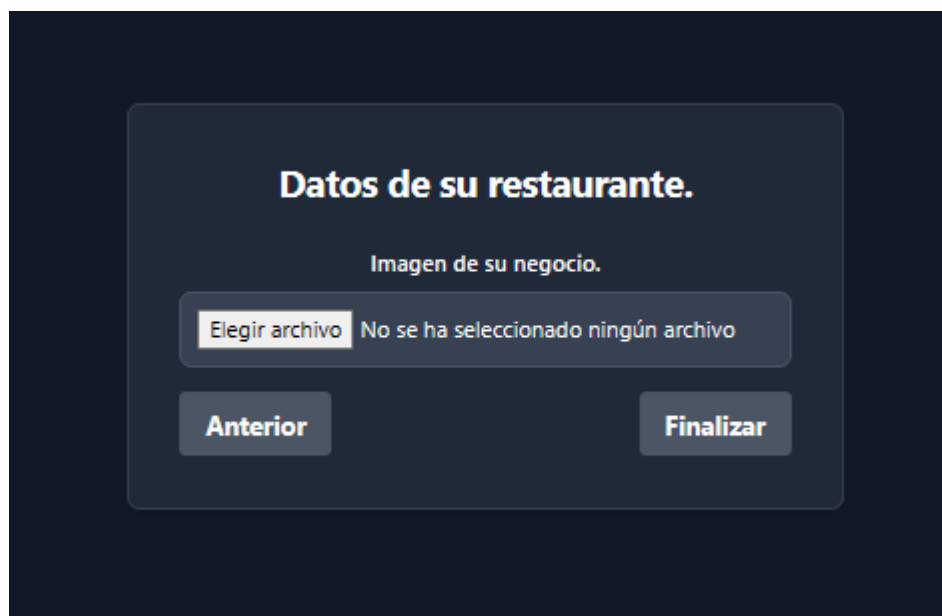
Datos de su restaurante.

Nombre de su negocio.

Siguiete

Figura 80: Paso 1 en la creación de restaurante de Lunch At Home

Paso 2 del proceso guiado:



Datos de su restaurante.

Imagen de su negocio.

Elegir archivo No se ha seleccionado ningún archivo

Anterior **Finalizar**

Figura 81: Paso 2 en la creación de restaurante de Lunch At Home

Una vez terminado, iremos a la vista del usuario en el rol de “Restaurante”, donde podremos ver los distintos restaurantes asociados a un email, ya que puede tener varios negocios.

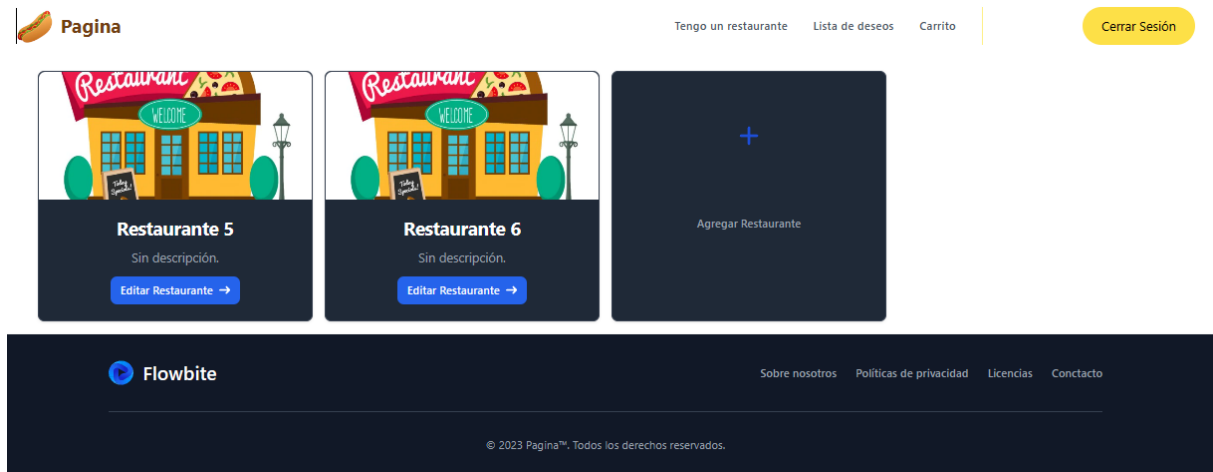


Figura 82: Edición de restaurante de Lunch At Home

Aquí podremos añadir un nuevo restaurante o editar uno que ya tenemos.

Instrucciones: Editar restaurante

Una vez hayamos completado el paso “Crear Restaurante” ahora nuestra vista será la de los restaurantes asociados a nuestro email y tendremos que presionar en el botón “Editar restaurante” donde nos aparecerá la siguiente vista:

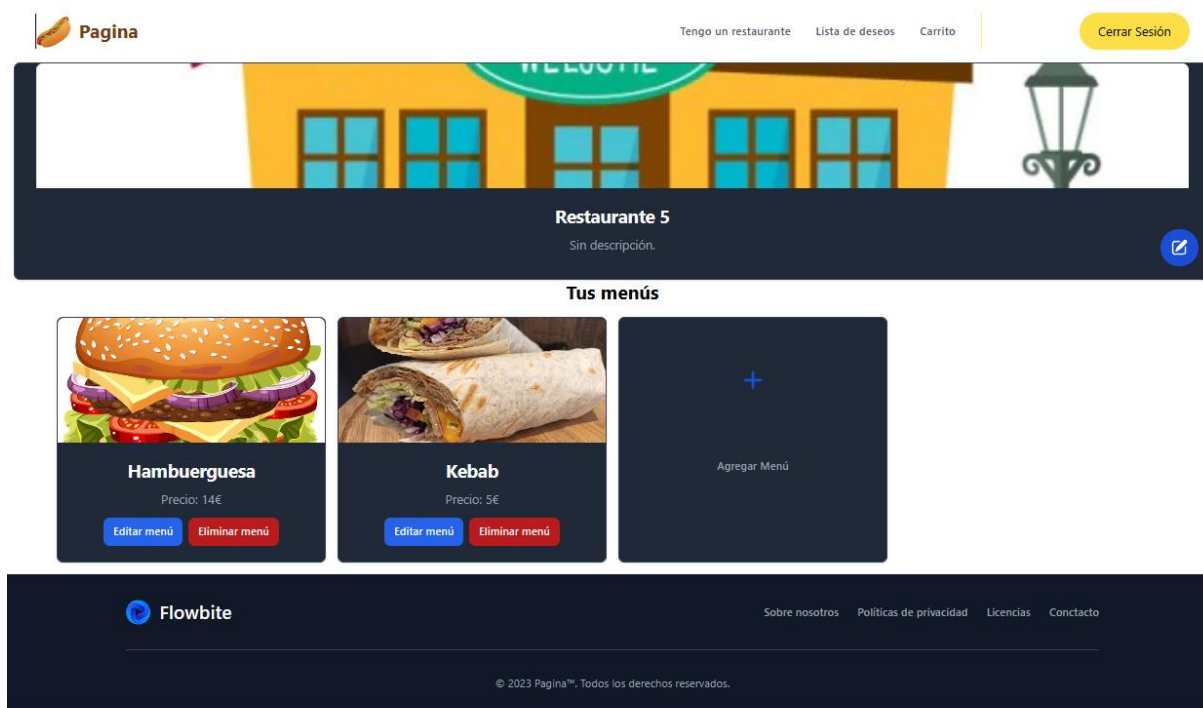


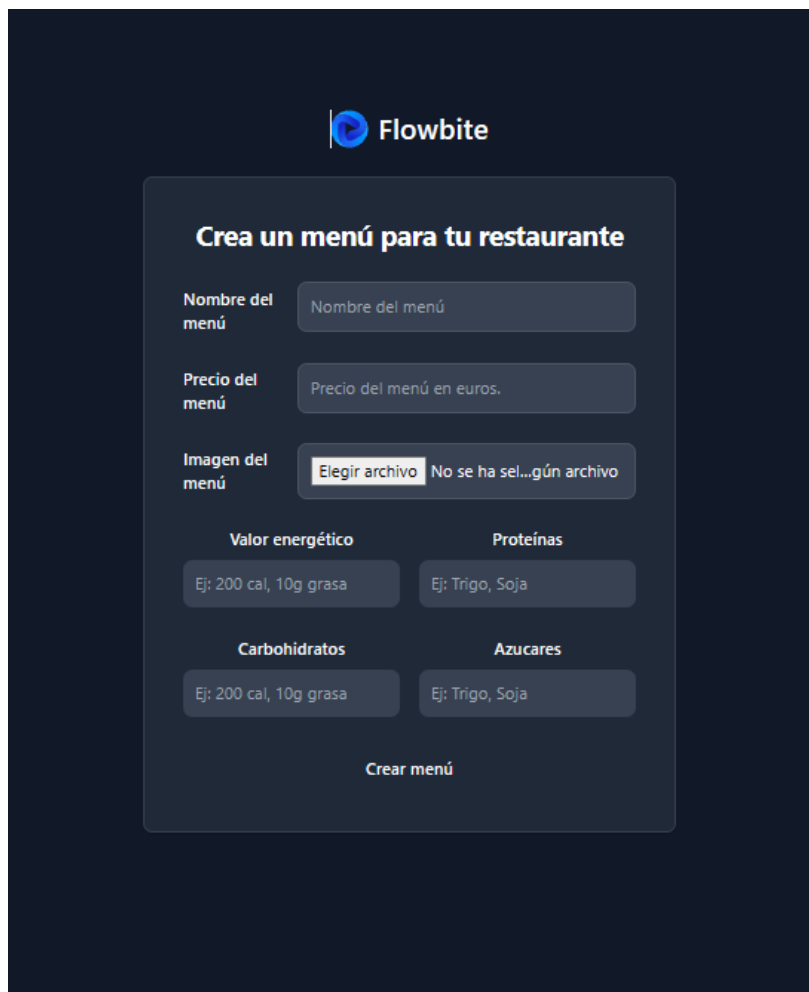
Figura 83: Edición de menús de Lunch at Home

En la tarjeta de nuestro restaurante tendremos un botón flotante, con el icono de edición, donde se nos presentará el proceso guiado para editar el restaurante.

Los campos que no se rellenen permanecerán igual en nuestra base de datos por lo que al usar el botón de “Confirmar” se guardarán los cambios y se actualizará el sistema con la nueva información.

Instrucciones: Añadir menú

Una vez hemos iniciado sesión en nuestra cuenta en el rol de restaurante, editaremos el restaurante en el que queremos añadir el menú y a continuación presionamos el botón: Añadir menú que nos llevará al siguiente formulario:



The image shows a dark-themed web interface for creating a restaurant menu. At the top, the 'Flowbite' logo is visible. Below it, a central form titled 'Crea un menú para tu restaurante' contains several input fields. The first field is 'Nombre del menú' with a placeholder 'Nombre del menú'. The second is 'Precio del menú' with a placeholder 'Precio del menú en euros.'. The third is 'Imagen del menú' with a file selection button 'Elegir archivo' and a placeholder 'No se ha seleccionado ningún archivo'. Below these are four nutritional information fields: 'Valor energético' (placeholder: 'Ej: 200 cal, 10g grasa'), 'Proteínas' (placeholder: 'Ej: Trigo, Soja'), 'Carbohidratos' (placeholder: 'Ej: 200 cal, 10g grasa'), and 'Azúcares' (placeholder: 'Ej: Trigo, Soja'). At the bottom of the form is a 'Crear menú' button.

Figura 84: Formulario de menú de Lunch At Home

Estos con los campos que aparecerán para rellenar, obligatorios y serán los que luego se muestren al usuario cuando vaya a comprar.

Instrucciones: Editar menú

Los primeros pasos son iguales que las instrucciones para añadir un menú. Iniciamos sesión, editamos el restaurante y a continuación le daremos al botón de editar menú y nos aparecerá el mismo formulario que para añadir el menú con la diferencia de que no hará falta rellenar los datos que sean iguales. Guardamos los cambios y la información será actualizada en el sistema.

Instrucciones: Eliminar menú

Si deseamos eliminar un menú, tenemos que ir al botón “Eliminar un menú” y nos saldrá el siguiente botón de confirmación:

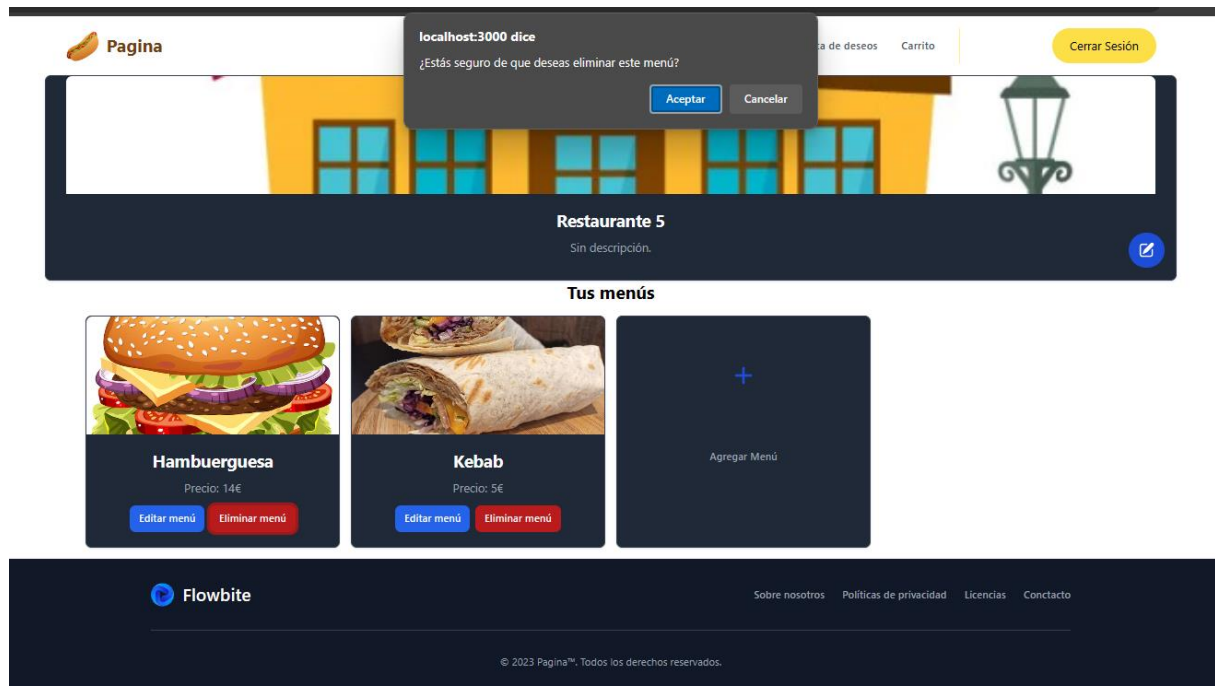


Figura 85: Eliminar menú en Lunch At Home

Instrucciones: Vista administrador

Para entrar al modo administrador, el rol debe ser proporcionado por el ingeniero jefe de la aplicación. Una vez tenemos una cuenta de administrador, metemos las credenciales y entraremos en la siguiente vista:

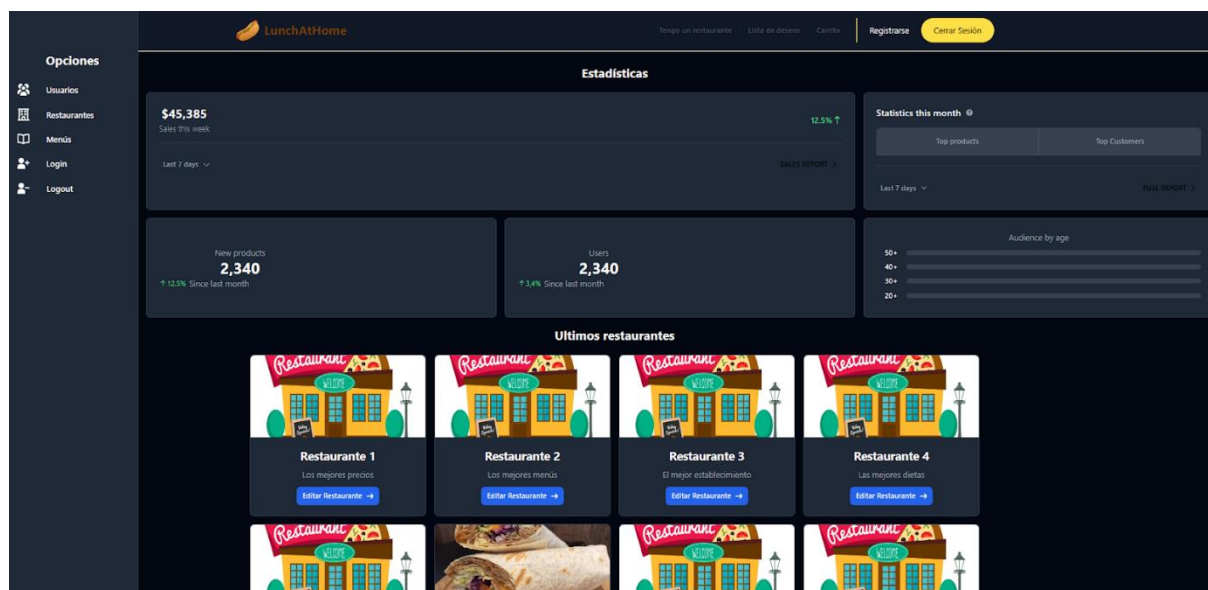


Figura 86: Vista principal de administrador de Lunch at Home

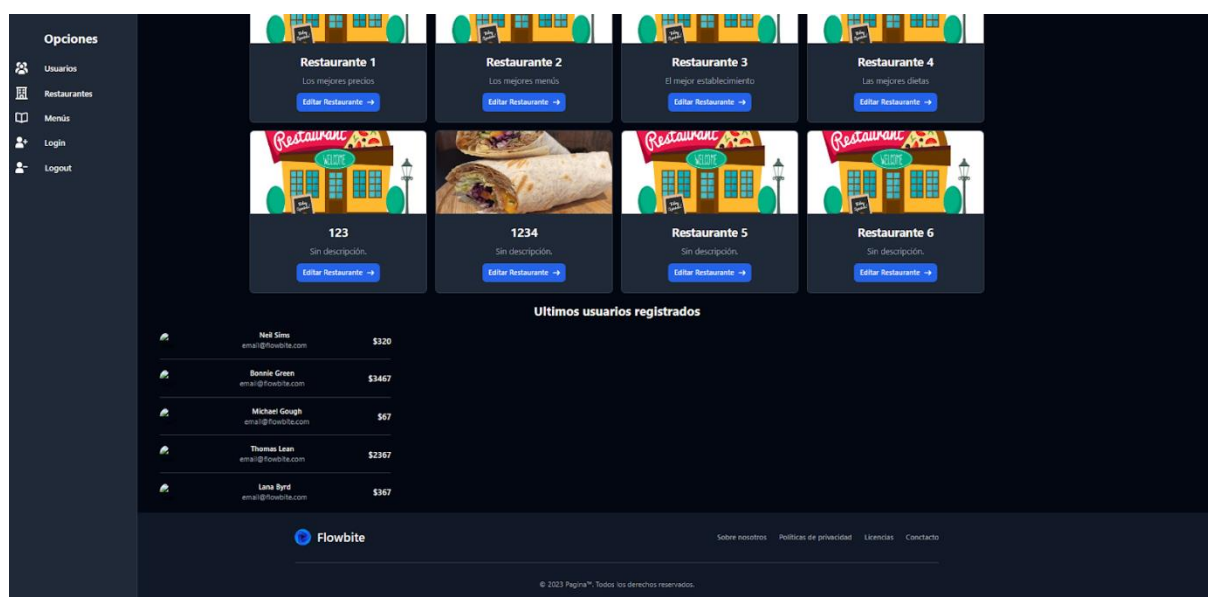


Figura 87: Vista principal de administrador de Lunch at Home

Aquí nos saldrá información importante sobre la página, cuando tengamos usuarios y restaurantes que quieran estar en nuestra aplicación y podremos realizar distintas cosas.

1. Administrar los usuarios

Si presionamos en el botón “Usuarios” se nos redirigirá a la siguiente vista donde podremos gestionar los usuarios.



Tengo un restaurante

Lista de deseos

Carrito

Cerrar Sesión

	admin@admin.com Administrador	Editar Eliminar
	fjrc0010@red.ujaen.es11 Restaurante	Editar Eliminar
	fjrc0010@red.ujaen.es123 Restaurante	Editar Eliminar
	fjrc0010@red.ujaen.es1234 Restaurante	Editar Eliminar
	fjrc0010@red.ujaen.es12345 Restaurante	Editar Eliminar
	fjrc0010@red.ujaen.es123456 Restaurante	Editar Eliminar
	fjrc0010@red.ujaen.es1234567 Restaurante	Editar Eliminar
	francisco_kkena@hotmail.com Cliente	Editar Eliminar
	francisco_kkena@hotmail.com1 Restaurante	Editar Eliminar
	francisco_kkena@hotmail.com2 Restaurante	Editar Eliminar
	francisco_kkena@hotmail.com3 Restaurante	Editar Eliminar

Figura 88: Edición de usuario de administrador de Lunch at Home

2. Administrar los restaurantes

Igual que la opción anterior, podremos gestionar los restaurantes.

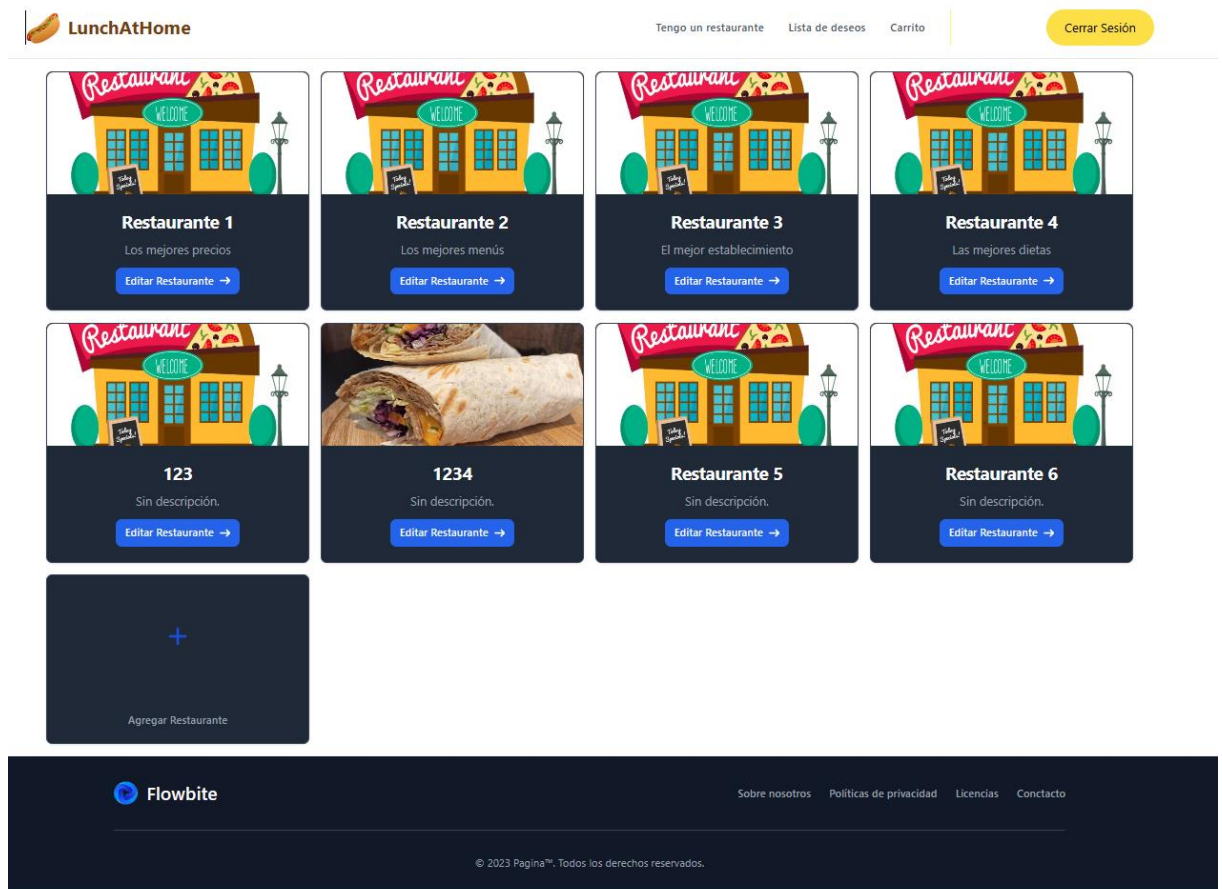


Figura 89: Edición de restaurantes de administrador de Lunch at Home

3. Administrar los menús.

Por último, podemos gestionar los menús de los restaurantes, como, por ejemplo, en caso de que incumplan alguna norma.

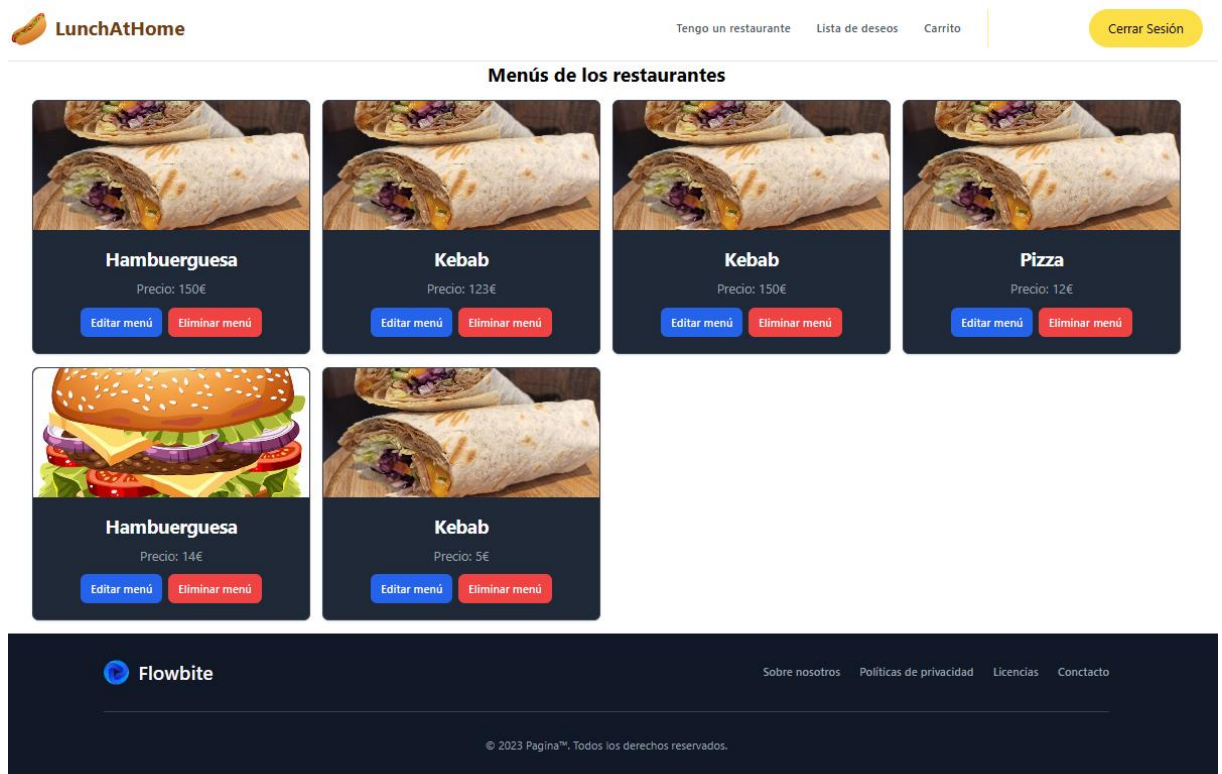


Figura 90: Edición de menús de administrador de Lunch at Home