

Doble Grado en Ingeniería Informática y Administración y
Dirección de Empresas
(2021-2022)

Trabajo Fin de Grado

“Aplicación web para verificación de opciones sin gluten en restaurantes”

Aglae Pérez Salcedo
Tutor

Alejandro Rey López.

Colmenarejo, Agosto 2022



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

Resumen Ejecutivo

La búsqueda de restaurantes que tengan opciones sin gluten no es tan fácil hoy en día. En el mercado existen numerosas páginas web con buscadores de restaurantes pero estas parecen no estar pensadas para las personas que pueden tener alergias o intolerancias. Este trabajo propone desarrollar un buscador de restaurantes que ofrecen opciones sin gluten. Esta idea, nace a partir de mi interés por la gastronomía. Cuando fui diagnosticada con celiaquía mi entusiasmo por conocer nuevos restaurantes se vio frenado. Por lo que este buscador pretende poder descubrir las opciones sin gluten que ofrece cualquier tipo de restaurantes, ya sean grandes franquicias como Foster's Hollywood o pequeños restaurantes a nivel local. Para llevar a cabo el desarrollo de este proyecto se realizará primero un estudio de las distintas webs y aplicaciones que ofrece el mercado, para poder tener en cuenta sus características y terminar de cubrir la necesidad que se presenta.

En primer lugar, para poder llevar a cabo la implementación del proyecto, se establecerán los límites y restricciones que presenta el sistema para así poder establecer su alcance. Por otro lado, se concretarán todos los requisitos y funcionalidades a desarrollar. Además, se analizarán las distintas tecnologías que pueden ser usadas para la implementación, diseño y gestión del sistema así como sus ventajas y desventajas a tener en cuenta. Una vez terminada la implementación, se deberán realizar pruebas que puedan verificar la funcionalidad del sistema en su totalidad.

En segundo lugar, se ha de tener en cuenta el marco legal en el que se encuentra el proyecto, a pesar de que este sea una prueba de concepto, es importante conocer la legislación que se aplicaría si la aplicación fuese accesible desde internet.

En tercer lugar, se ha de desarrollar un plan de gestión del proyecto dividiéndolo en distintas etapas además de calcular un presupuesto monetario que representaría la realización del proyecto por una empresa externa formada por un equipo profesional.

Agradecimientos

Quiero agradecerle a mis padres y familiares en primer lugar por haberme apoyado durante estos 6 años y por todo el sacrificio que esto ha conllevado. Por otro lado a mis amigos de la universidad ya que han sido un apoyo fundamental y sin ellos toda esta experiencia no hubiese sido posible. Todas las tardes y fines de semana estudiando y haciendo practicas por fin han dado su fruto.

Por otro lado quiero agradecer a mis amigos Luis y Nacho que me han apoyado y guiado durante la carrera y sobretodo en la etapa del desarrollo del TFG. Por último, Javier gracias por aguantarme todo este año, se que no ha sido fácil pero esta etapa esta llegando a su fin.

ÍNDICE

1. INTRODUCCION	9
1.1 Motivaciones.....	9
1.2 Objetivos.....	10
1.3 Estructura del documento.....	11
1.4 Definiciones.....	12
1.5 Acrónimos.....	12
2. ESTADO DEL ARTE.....	13
2.1 Arquitectura cliente-servidor.....	13
2.1.1 Interacción cliente-servidor.....	14
2.1.2 Ventajas y desventajas del modelo cliente-servidor.....	14
2.2 Front-End.....	15
2.2.1 HTML.....	15
2.2.2 JavaScript	16
2.2.3 CSS.....	16
2.3 Back-End.....	16
2.3.1 Python.....	16
2.3.2 Java.....	17
2.3.3 Ruby	18
2.3.4 PHP.....	18
2.4 Bases de Datos	19
2.4.1 Bases de datos relacionales	19
2.4.2 Bases de datos no relacionales	22
2.5 Arquitectura del sistema.....	24
2.5.1 Arquitectura monolítica.....	24
2.5.2 Arquitectura de Microservicios	25
2.5.3 Arquitectura Hexagonal	26
2.6 Metodologías de trabajo.....	27
2.6.1 Scrum.....	28
2.6.2 Kanban.....	29
3. PLANTEAMIENTO DEL PROBLEMA.....	30
3.1 Necesidad existente en el mercado.....	30
3.2 Soluciones existentes en el mercado.....	31
3.4 Alcance del proyecto	35
3.5 Obstáculos y restricciones existentes	36
3.6 Requerimiento de usuarios potenciales	36
3.7 Identificación de los interesados del sistema	37
3.8 Estimación de recursos	38
3.8.1 Procedimiento de estimación de recursos	38
4. ANÁLISIS DEL SISTEMA.....	40
4.1 Casos de uso.....	40

4.1.1 Relacionado con el registro de Usuario.....	40
4.1.2 Relacionado con el registro de Restaurantes.....	41
4.1.3 Relacionado con la creación de Platos	41
4.1.4 Relacionados con la búsqueda realizada por el usuario	42
4.2 Análisis de requerimientos.....	43
4.2.1 Análisis de requisitos.....	43
4.2.1.1 Requisitos de Usuario.....	44
4.2.1.2 Requisitos Funcionales.....	46
4.2.1.5 Requisitos relacionados con los platos:.....	50
4.2.2 Requisitos no Funcionales.....	51
5. JUSTIFICACIÓN DE LA SOLUCIÓN	53
5.1 Elección de las tecnologías utilizadas	53
5.2 Elección de la Metodología utilizada	55
5.3 Diseño de base de datos.....	56
5.4 Arquitectura	58
5.5 Guía de usuario.....	60
6. VALIDACIÓN DEL SOFTWARE.....	65
6.1 Pruebas.....	65
6.1 Test de Requisitos funcionales.....	66
6.2 Test de Requisitos no funcionales.....	72
6.3 Matriz de trazabilidad	74
7. MARCO REGULADOR.....	76
7.1 Legislación aplicable	76
7.2 Licencias y propiedad intelectual	77
8. GESTION DEL PROYECTO	78
8.1 Organización del trabajo	78
8.2 Planificación	79
8.2.1 Diagrama de Gantt.....	79
8.3 Control de desviaciones.....	81
9. ENTORNO SOCIO-ECONÓMICO	82
9.1 Presupuesto.....	82
9.1.1 Recursos Software.....	83
9.1.2 Recursos Hardware.....	83
9.1.3 Recursos Humanos.....	84
9.1.4 Recursos indirectos.....	85
9.1.5 Resumen total de costes	85
9.2 Impacto socio-económico	86
10. CONCLUSIONES.....	88
10.1 Retrospectiva del proyecto	88
10.2 Problemas encontrados	89
10.3 Conclusiones personales	90

10.4 Trabajos Futuros	92
11. BIBLIOGRAFIA	93

INDICE DE TABLAS

Tabla 1 Ventajas y Desventajas modelo relacional Fuente: Propia.....	14
Tabla 2 Ventajas y Desventajas modelo relacional Fuente: Propia.....	22
Tabla 3 Ventajas y desventajas del modelo no relacional.	24
Tabla 4 Ventajas y desventajas del Arquitectura Monolítica	25
Tabla 5 Ventajas y desventajas de Arquitectura de microservicios.	25
Tabla 6 Ventajas y desventajas de Arquitectura Hexagonal.	27
Tabla 7 Especificaciones ordenador Fuente: Propia.....	36
Tabla 8 Características preferibles de los usuarios. Fuente: Propia	37
Tabla 9 Stakeholders Fuente: Propia	38
Tabla 10 Estimación de recursos .Fuente: Propia.....	39
Tabla 11 Plantilla de requisitos software.....	45
Tabla 12 Requisito software 01.	46
Tabla 13 Requisito software 02.	46
Tabla 14 Requisito software 03.	46
Tabla 15 Requisito software 04.	47
Tabla 16 Requisito software 05.	47
Tabla 17 Requisito software 06.	47
.Tabla 18 Requisito software 07.	48
Tabla 19 Requisito software 08.	48
Tabla 20 Requisito software 09.....	48
Tabla 21 Requisito software 09.	49
Tabla 22 Requisito software 10.	49
Tabla 23 Requisito software 11.	49
Tabla 24 Requisito software 12.	50
Tabla 25 Requisito software 13.	50
Tabla 26 Requisito software 14.	50
Tabla 27 Requisito software 15.	51
Tabla 28 Requisito software 16.	51
Tabla 29 Requisito software no funcional 01.	51
Tabla 30 Requisito software no funcional 02.	52
Tabla 31 Requisito software no funcional 03.	52
Tabla 32 Plantilla de requisitos software 04.....	52
Tabla 35 Plantilla de test.	65
Tabla 36 Test 01	66
Tabla 37 Test 02	66
Tabla 38 Test 03	67
Tabla 39 Test 04	67
Tabla 40 Test 05	68
Tabla 41 Test 06	68
Tabla 42 Test 07	69
Tabla 43 Test 08	69
Tabla 44 Test 09	69
Tabla 45 Test 10	70
Tabla 46 Test 11	70
Tabla 47 Test 12	71
Tabla 48 Test 13	71
Tabla 49 Test 14	72
Tabla 50 Test 15	72

Tabla 51 Test 16	72
Tabla 52 Test 17	73
Tabla 53 Matriz Trazabilidad	74
Tabla 54 Matriz Trazabilidad	75
Tabla 55 Horas invertidas Fuente: Propia	82
Tabla 56 Recursos Software Fuente: Propia.....	83
Tabla 57 Recursos Hardware Fuente: Propia	83
Tabla 58 Recursos Humanos Fuente: Propia.....	84
Tabla 59 Recursos indirectos Fuente: Propia	85
Tabla 60 Resumen Costes Totales Fuente: Propia	85

INDICE DE ECUACIONES

Equation 1 Amortización.....	83
------------------------------	----

1. INTRODUCCION

Salir a comer a un restaurante siendo celiaco a día de hoy es bastante tedioso. Por lo general para poder planificar ir a un restaurante se necesita buscar la página web de este si es que la tiene, buscar el menú y por último asegurarse si la carta contiene leyenda de alérgenos para finalmente comprobar que platos pueden comer.

Hoy en día, existen guías desactualizadas de restaurantes aptos para celiacos o guías de restaurantes que ofrecen íntegramente comida sin gluten pero, no se ofrece todavía una página donde puedas acceder a los menús de cualquier restaurante con facilidad visualizando únicamente los platos sin gluten que ofrecen .

1.1 Motivaciones

La motivación principal y por la que nace la idea de crear una aplicación web donde se pueda acceder fácilmente a las opciones sin gluten que ofrecen los restaurantes proviene principalmente de mi diagnóstico como celiaca. Fui diagnosticada hace 5 años y desde entonces planificar salir a comer a un restaurante siempre me conlleva tiempo de búsqueda, a veces inútil, que acaba limitando no poder descubrir nuevos lugares que ofrecen opciones sin gluten en sus menús.

Desde pequeña he sido siempre muy fan de la gastronomía y de probar platos de distintas culturas y países. Cuando me diagnosticaron celiacía, mis opciones de poder seguir con esta costumbre se redujeron considerablemente. Tras investigar a fondo muchos restaurantes que tenían opciones sin gluten decidí crear una lista con restaurantes que ofrecen platos sin gluten. Esta lista ha sido de gran ayuda para otros conocidos celiacos por lo que la idea de esta app web surge de llevar esto más allá y poder consultar los menús de los restaurantes fácilmente.

1.2 Objetivos

El objetivo general de este trabajo es desarrollar una aplicación web en la que a través de un buscador se pueda visualizar las opciones sin gluten que ofrecen los restaurantes.

Los objetivos principales en detalle son los siguientes:

- Desarrollar una aplicación web que pueda permitir visualizar las opciones sin gluten disponibles en los restaurantes. Esto ahorra tiempo de investigación y proporciona comodidad principalmente a los usuarios que padecen celiaquía.
- Desarrollar una web que permita:
 - Poder Registrarse e Iniciar sesión en el sistema.
 - Acceder a visualizar, modificar y eliminar los datos de registro del usuario.
 - Registrar Restaurantes pudiendo visualizar, modificar o eliminar sus datos.
 - Registrar Platos asociados a los restaurantes pudiendo visualizar sus datos, modificarlos o eliminarlos.
 - Realizar búsquedas por nombre de restaurante o plato.
- Desarrollar el diseño de la aplicación a nivel de arquitectura que permita su crecimiento.

1.3 Estructura del documento

Este documento se ha dividido en diferentes apartados. A continuación se presentará un breve resumen del contenido que incluirán.

2. Introducción: esta sección incluye la descripción de la idea general del proyecto junto con los objetivos y las motivaciones por las que se ha decidido llevar a cabo.
3. Estado del arte : esta sección presenta y profundiza en diferentes tecnologías existentes que se pueden utilizar para llevar a cabo de desarrollo del proyecto tanto a nivel de gestión como a nivel de implementación. Se describen sus principales características y se realiza el estudio de reflejar sus ventajas y desventajas.
4. Planteamiento del problema : este apartado está diseñado para presentar la necesidad que hay en el mercado y hacer un estudio del entorno encontrando las ofertas existentes que solucionan parcialmente el problema presentado. Además se incluye el alcance que tendrá el sistema a desarrollar junto con los interesados que forman parte de este. Por último, se realiza una estimación de los recursos que se usarán
5. Análisis del Sistema: este apartado es el que describe que tiene que hacer el sistema analizando los requisitos que tiene que cumplir.
6. Justificación de la solución: esta sección describe la justificación de las tecnologías finalmente elegidas para la implementación de la aplicación web. Además se presentará el diseño y arquitectura que se ha seguido para realizar el trabajo.
7. Validación software: este apartado incluye las pruebas realizadas al sistema para verificar su funcionalidad.
8. Gestión del proyecto: en esta sección se describe la organización, planificación y las posibles desviaciones que puede sufrir el proyecto a lo largo de su desarrollo
9. Marco Regulatorio : este apartado analiza la legislación que se le aplicaría al sistema, además de cuestiones como las licencias de herramientas utilizadas.
10. Entorno socioeconómico: este apartado desarrolla un presupuesto por la elaboración del proyecto escogiendo la situación de que se llevase a cabo por una empresa compuesta por un equipo de personas. Además se comenta el impacto socio-económico que presenta el proyecto.

1.4 Definiciones

Este apartado contiene definiciones para poder comprender los conceptos necesarios a lo largo del documento

- **Página web:** conjunto de información ofrecida en una determinada dirección de internet.
- **Sin Gluten:** Referente a un alimento, que no contiene trigo, cebada, centeno o malta.
- **Celiaquía:** Enfermedad digestiva que afecta al intestino por la ingesta de gluten en el paciente.
- **Stakeholder:** Personas que de interés del proyecto. Tanto internos (clientes) como externos, personas involucradas en el proyecto, es decir, los integrantes del equipo que desempeña el proyecto.
- **API:** Interfaz de programación de aplicaciones.
- **Framework:** marco de trabajo que ofrece una estructura como base para elaborar un proyecto[1]

1.5 Acrónimos

Este apartado contiene los acrónimos necesarios para poder comprender los conceptos necesarios a lo largo del documento

- APP: Aplicación
- TFG: Trabajo de Fin de grado.
- HTML5 : Lenguaje de marcado para la elaboración de páginas web.
- CSS: Hojas de estilo. Lenguaje de diseño gráfico para documentos estructurados con un lenguaje de marcado.
- URL:Uniform Resource Locator.
- IP: Protocolo de internet.
- DNS: Domain Name System.
- MVC: Model View Controller.
- TIW: Tecnologías informáticas para la web.

2. ESTADO DEL ARTE

Este apartado describe diferentes posibles tecnologías usadas hoy en días para el desarrollo de una aplicación web junto con el análisis de las opciones finalmente elegidas.

2.1 Arquitectura cliente-servidor

La arquitectura Web se basa en el modelo cliente-servidor. El modelo divide las tareas en dos partes, cliente y servidor permitiendo así, intercambiar información entre dos o más procesos que no están relacionados mediante protocolos web. Este modelo, permite a los usuarios poder acceder simultáneamente a la misma base de datos permitiendo que esta guarde la información.[2]

Componentes:

- El cliente: por lo general hablamos de usuarios o una organización que demanda un servicio a través de un navegador como por ejemplo Firefox, Chrome, etc.
- Servidor: También conocido como proveedor de servicios que esta compuesto por un ordenador o conjunto de ellos. Se encarga de realizar alguna acción o devolver información al cliente cuando este realiza una solicitud.

Un ejemplo del sistema cliente-servidor puede ser :

- Navegador web. El navegador le solicita una página web al servidor. El servidor tras recibir la petición, reúne la información necesaria y se la envía al cliente.
- Servicio de Banca online: el usuario le solicita al servidor información o una acción de su cuenta bancaria como por ejemplo retirada de dinero o el saldo disponible, en este caso, el servidor accede a la base de datos del banco para reunir la información y devolvérsela.[3]

2.1.1 Interacción cliente-servidor

El proceso que realiza un navegador con un servidor es el siguiente:

1. El usuario introduce la URL de una página web.
2. El navegador realiza la petición a través de su correspondiente DNS, Domain Name System.
3. Este sistema de DNS traduce una URL asociándola a una IP que va a estar asignada al servidor que aloja la página web.
4. El navegador ya teniendo la IP de la web envía a través de HTTP/HTTPS una solicitud a la IP del servidor.
5. El servidor devuelve los archivos necesarios para presentar la página web.

2.1.2 Ventajas y desventajas del modelo cliente-servidor

Ventajas	Desventajas
• Sistema centralizado. • Control, al estar los archivos centralizados • Seguridad, se puede restringir el acceso a través del uso de credenciales. • La capacidad de ambas partes se puede aumentar por separado fácilmente. • Accesibilidad, fácil acceso sin necesidad de terminal.[4]	• Sistemas poco robustos, al estar centralizado si una parte del sistema, este cae entero. • Costes altos para mantener los servidores • Problemas de tráfico: cuando los clientes realizan tantas peticiones que por la capacidad el servidor no puede gestionarlas todas.

Tabla 1 Ventajas y Desventajas modelo relacional

Fuente: Propia

2.2 Front-End

En una página Web, término Front-End también llamado capa de presentación se refiere a toda parte con la que puede ver o interactuar el cliente. Por lo tanto, el front-end se encarga del desarrollo grafico de la interfaz de usuario.

En su desarrollo interviene el diseño web adaptativo o también llamado “responsive web design” que permite poder acceder a un página web tanto desde un ordenador como desde un móvil o Tablet haciendo que el diseño se adapte al diferente tamaño de pantallas. Las características críticas para un buen desarrollo del front-end son la accesibilidad, velocidad y rendimiento en una web.[5]

A continuación se nombrarán diferentes tecnologías que se usan para el desarrollo del front-end.

2.2.1 HTML

Lenguaje de etiquetas de hipertexto es el fundador y base de las páginas web. Este lenguaje define la estructura de la página web mediante etiquetas con diferente significados como encabezados, títulos, imágenes o párrafos[6]. El término hipertexto indica los enlaces que conectan las páginas entre si, ya sea dentro de una misma pagina web o entre diferentes.

Toda página web necesita el uso de HTML para permitir la visualización de texto o imágenes; aunque por lo general, está acompañado del uso de otros lenguajes como CSS o JavaScript.

En 2014 tuvo el lanzamiento de HTML5 que incluía el uso de elementos multimedia tanto de video como de audio de forma nativa, eliminando así la necesidad de descargar Adobe Flash Player. Hoy en día HTML5 sigue en uso. [5]

2.2.2 JavaScript

Lenguaje de programación ejecutable en los navegadores que permite a los desarrolladores herramientas para crear aplicaciones web interactivas y dinámicas. Este lenguaje permite una mayor flexibilidad y abanico más grade de posibilidades como: crear efectos, animaciones o eventos sin ninguna interacción por parte del usuario o causadas por los usuarios al pulsar en partes de la página como en botones. [7]

2.2.3 CSS

Llamado hojas de estilo o lenguajes de estilo que se usa para describir y personalizar la presentación de documentos HTML. Permite elegir características como la fuente, los colores, tamaño de márgenes, etc. Los beneficios que tienen estas hojas de estilo son:

- Mayor precisión: Te permite poder elegir medidas exactas en pixeles.
- Menores tiempos de carga: Al dividir el contenido, y no estar en el documento de HTML las páginas se cargan más rápido lo que hace bajar el volumen de tráfico del servidor que lo procesa.

2.3 Back-End

El término Back-End también llamado capa de acceso de datos se refiere a la parte de la página web que el usuario no ve. Esta parte es la responsable de procesar todas las peticiones que realizan los usuarios. El back-end se encarga de acceder a la base de datos para consultar o procesar modificaciones.

A continuación se nombrarán diferentes tecnologías que se usan para el desarrollo del front-end.

2.3.1 Python

Lenguaje de programación que nace en década de los 90 por Guido Van Rossum, informático holandés. A día de hoy, es uno de los 3 lenguajes de programación más usado

del mundo[8]. Python se destaca principalmente por su sintaxis simple y expresiva que facilita su uso y desarrollo, además se caracteriza por:

- Multiplataforma: Este lenguaje se puede usar en diferentes sistemas operativos
- Ser un lenguaje orientado a objetos (POO).
- Posee licencia de código abierto: lo que implica que su uso y distribución es libre.
- Posee una gran cantidad de librerías y de módulos.

Este lenguaje de programación ha sido usado para desarrollar aplicaciones tan conocidas como Spotify, Instagram o Pinterest.

2.3.2 Java

Lenguaje de programación creado en 1983 por la empresa de Sun Microsystem y posteriormente comprada por la compañía Oracle en 2010[9]. Se caracteriza por ser un lenguaje de programación orientado a objetos además de ser:

- Un lenguaje robusto: es decir, que es seguro teniendo una gran capacidad frente a los errores que ocurren mientras se ejecuta un programa.
- Multihilo (Multithread) es un lenguaje capaz de realizar tareas de forma paralela y concurrente en un mismo programa
- Capaz de ejecutarse en distintos tipos de hardware.

Dentro del lenguaje de programación de Java existe Java Enterprise Edition, esto es una plataforma que permite el desarrollo de software de aplicaciones que se ejecutan en servidores web. La estructura de Java EE se compone de varias tecnologías como JSP, JSM o JPA[10]

Por otro lado el framework o entorno de trabajo más usadas de Java es Spring. Este entorno te permite desarrollar aplicaciones web de forma sencilla ya que es totalmente modular y además permite la escalabilidad en sus proyectos[11].

Este lenguaje ha sido usado para desarrollar programas como LinkedIn o Yahoo.

2.3.3 Ruby

Lenguaje de programación presentado en 1995 por Yukihiro Matsumoto. Se conoce por ser usado para el desarrollo de aplicaciones web, procesamiento de datos en backend o para otro tipo de aplicaciones software. Sus principales características son :

- Ser un lenguaje orientado a objetos.
- Ser dinámico ya que se pueden crear programas que puedan controlar otros programas, lo que es llamado metaprogramación.
- Es un lenguaje que no necesita ser compilado con Java ya que es un lenguaje interpretado de scripts, lo que quiere decir que el interprete de Ruby solo haría falta que lo tradujese al lenguaje máquina.

Este lenguaje ha sido usado para desarrollar aplicaciones como Airbnb o GitHub.

2.3.4 PHP

Es un lenguaje de programación que se ejecuta en el lado del servidor. Este lenguaje se denomina como “interpretado” ya que no necesita una previa compilación para ser entendido por el lenguaje máquina. Su uso mayoritariamente es para desarrollar páginas web, crear contenido dinámico e interactuar con bases de datos[12].

Las características de este lenguaje son :

- Poder ser incrustado en HTML lo que implica que en el mismo archivo se podrán combinar código de PHP con las etiquetas de HTML.
- Soporta la mayoría de bases de datos como MongoDB o MySQL.
- Fácil de instalar ya que existen paquetes que integran PHP de una forma rápida y sencilla.

Este lenguaje ha sido usado para desarrollar páginas web como Wordpress o Moodle.

2.4 Bases de Datos

Para el desarrollo de la aplicación que se propone en el proyecto, es fundamental una base de datos. A continuación analizaremos un análisis de las bases de datos dividiéndolas en relacionales y no relacionales.

2.4.1 Bases de datos relacionales

Las bases de este tipo de base de datos fueron postuladas en 1970 por Edgar Frank Codd en los laboratorios de IBM en San José[13]. Este tipo de base de datos se caracteriza por ser una recopilación de elementos de datos que poseen relaciones definidas entre ellos.

Estas relaciones son predefinidas ya que se realiza previamente un diseño de la estructura estudiando las características y relaciones que existen entre las entidades creadas.

Los elementos de estas bases de datos se ven representados por una serie de columnas y filas que forman una tabla. Estas tablas guardan la información en donde cada fila, también llamado registro tienen un identificador único denominada clave. La principal razón por la que se estructura de esta forma es para evitar la duplicidad de información haciendo que garantice la integridad de la información .

La estructura de una base de datos relacional se divide en esquema y en datos.

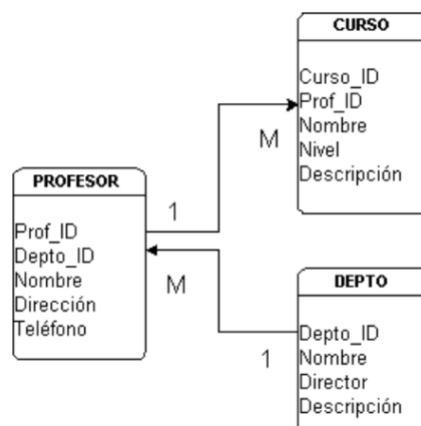


Figure 1 Esquema de datos

Los esquemas definen la estructura de la base de datos además del tipo de relación que hay entre entidades. En estos esquemas, los atributos indican el tipo de dato que son como por ejemplo un número un nombre o una descripción. Además, los esquemas contiene el nombre de las tablas y de sus atributos.

Componentes de las bases de datos relacionales:

- Entidad: Unidad de base de datos al que corresponde una o varias tablas.
- Atributos: Son las características de las entidades que participan en base de datos. Representado como una columna o campo.
- Tuplas: Representan un solo objeto que se muestra en la tabla. Representado como una fila o registro.
- Clave: Dato o campo que distingue de forma única una tupla de las demás en una tabla. Es un identificador único.
- Cardinalidad: Número de filas o tuplas que tiene un base datos.
- Grado: Número de columnas que contiene la tabla.
- Consultas: También llamado Queries. Son búsquedas de información concretas a la tabla que representa la base de datos[14].

Para la realización de consultas a las bases de datos existe el lenguaje de SQL (Structured Query Language). Este lenguaje de consultas está basado en el algebra relacional, por lo que de forma intuitiva y eficiente, se pueden realizar queries que ofrecen integridad referencial. Esto quiere decir, que al borrar un registro de la base de datos se borrará a su vez toda la información que tenía dependiente, indiferentemente si se encuentra en la misma o en distintas tablas.

Existe una amplia variedad de gestores de bases de datos relacionales. Estos son sistemas software que dan la capacidad a los usuarios de almacenar datos para así poder acceder a ellos, actualizarlos o borrarlos. Pondremos de ejemplo para analizar las siguientes:

2.4.1.1 Mysql

Es uno de los sistemas gestión de bases de datos más conocido a día de hoy ya que esta basada en una doble licencia, por una parte es de código abierto y otra parte fue adquirida por Oracle[15].

Las principales características de este gestor de base de datos es:

- Su compatibilidad con MySQL.
- Vistas: en las bases de datos de gran tamaño son fundamentales ya que permiten agrupar distintas tablas en una para filtrar la información.
- Arquitectura cliente servidor: Su arquitectura es cliente-servidor ya que los clientes realizan consultas a través del sistema para obtener datos de vuelta o para modificarlos y actualizarlos.
- Parte de código abierto, permitiendo a las empresas poder hacer uso de este para desarrollar aplicaciones.

2.4.1.2 Oracle

Está considerado uno de los sistemas gestores de bases de datos más completos del mercado. Fabricado por la empresa Oracle Corporation, una de las empresas mas grandes de software a nivel mundial.

Las principales característica que tiene Oracle:

- Herramienta intuitiva y fácil de usar.
- Riguroso control de acceso para la entrada de datos.
- Compatibilidad con sistemas operativos como Windows, Mac os o Linux.
- Lenguaje de diseño de las bases de datos. Se denomina PL/SQL y en un lenguaje específico de Oracle que soporta consultas[16].

Ventajas y desventajas del modelo de bases de datos relacional

Ventajas	Desventajas
Gran rendimiento	Alto coste de mantenimiento
Escritura simple	Dificultad de crecimiento
Alto grado de accesibilidad	Complejidad de instalación
Posibilidad de múltiples usuarios	Problemas de espacio
Preciso	

Tabla 2 Ventajas y Desventajas modelo relacional Fuente: Propia

2.4.2 Bases de datos no relacionales

Las bases de datos no relacionales surgen como solución a los defectos de las bases de datos relacionales ya que se quería encontrar una forma más óptima para tratar grandes cantidades de información de forma más rápida[14].

Este tipo de base de datos está principalmente diseñado para aplicaciones que poseen un gran volumen de datos y que necesitan esquemas flexibles. Para diferenciar las bases de datos no relacionales de las relacionales surgió el termino NoSQL. Estas bases de datos no poseen tablas por lo que permite más flexibilidad a la hora de agrupar datos

Las principales características son:

- Escalabilidad: permite el crecimiento del volumen de los datos sin tener que añadir un alto coste de forma exponencial.
- Alto rendimiento: Estas bases de datos han sido optimizadas para cierto tipo de datos.
- Flexibilidad: la información no se guarda con esquemas exactos sino que se organiza mediante documentos permitiendo tener datos semiestructurados[17].

A continuación mencionaremos un par de ejemplos de sistemas gestores de bases de datos no relacionales.

2.4.2.1 MongoDB

Es un sistema gestor de bases de datos no relacional que trabaja con documentos BSON. Estos documentos son una representación binaria de JSON a diferencia del modelo relacional, este no necesita de esquemas o tablas.

Sus principales características:

- Gran flexibilidad que da a la hora de trabajar.
- Realización de consultas: Este sistema permite todo tipo de consultas. Incluso, a pesar de no ser un sistema relacional soporta el uso de SQL para realizar consultas.
- Escalabilidad: tanto vertical como horizontal ya que permite incrementar los recursos destinados a la memoria o la CPU del servidor además de permitir aumentar la disponibilidad de la aplicación[18].

2.4.2.2 DynamoDB.

Gestor de base de datos ofrecida por Amazon que reside en Amazon Web Services. Este sistema soporta bases de datos no relacionales de clave-valor sin servidor por lo que no sería necesario ningún tipo de hardware que gestionar[19].

Las características de este sistema gestor son :

- Rendimiento a escala: Permite el tratado de tablas de cualquier tamaño.
- Réplica automatizada con tablas globales : Con su capacidad de escalabilidad global. Permite poder hacer una visualización en local tanto de lectura como escritura en segundos.
- Modelo clave-valor: Permite un esquema flexible ya que una fila o registro puede tener diferentes columnas unas de otras sin necesidad de volver a redefinir el esquema como las bases de datos relacionales.

Empresas como Snapchat o Disney hace uso de los recursos de DynamoDB.

Ventajas y desventajas de modelo de bases de dato relacional

Ventajas	Desventajas
Modelo clave valor	Consultas manuales
Rentabilidad de costes	No garantiza integridad
Gran escalabilidad	Soporte limitado
Falta de necesidad de crear diseño y tabla estructurado.	Falta de estandarización
Flexibilidad	

Tabla 3 Ventajas y desventajas del modelo no relacional.

Fuente: Propia

2.5 Arquitectura del sistema

Además de las posibles tecnologías a utilizar en el desarrollo del proyecto, es importante tener en cuenta la arquitectura de la aplicación web. A continuación analizaremos dos opciones de posibles arquitecturas exponiendo sus ventajas y desventajas.

2.5.1 Arquitectura monolítica

Una arquitectura monolítica en un proyecto software suele ser comparado con un “bloque” ya que este se compila y ejecuta como una unidad sin tener ni necesitar ningún tipo de dependencias externas. Estas unidades son totalmente autónomas de las otras aplicaciones. Este tipo de arquitectura era la usada años atrás cuando internet no era accesible por lo que no se podían consumir servicios externos.

A continuación se mencionarán las características que presentan este tipo de estructura:

- Autosuficiencia: como mencionado anteriormente este tipo de arquitectura no requieren de ningún recurso externo para ser ejecutadas.
- Todo el sistema se encuentra en una misma aplicación (Interfaz lógica, de usuario y de acceso)
- Simplicidad de desarrollo pero complejidad para tener escalabilidad.

Teniendo en cuenta algunas características de la arquitectura monolítica enumeraremos las ventajas y desventajas que ofrece [20].

Ventajas	Desventajas
• Bajo costo	• No escalable
• Autonomía	• Poco flexible
• Fácil de testear	• Un solo punto de fallo quedando el sistema inoperable completamente

Tabla 4 Ventajas y desventajas del Arquitectura Monolítica

Fuente: Propia

2.5.2 Arquitectura de Microservicios

La arquitectura de microservicios logra aislar las distintas partes que forman una aplicación con el objetivo que cada parte sea una aplicación propia. Esto quiere decir, esta arquitectura pretende desarrollar distintos servicios que sean independientes pero que se comuniquen unos con los otros estando asociado a un mismo sistema.

Dentro de las arquitecturas compuestas por microservicios existe el API Gateway. Este es un elemento sirve como puerta de entrada para controlar el acceso y funcionalidad de los microservicios[21]

Características de este tipo de arquitectura:

- Velocidad: permite a aplicaciones de gran tamaño ser publicadas de forma rápida.
- No requiere un gran equipo de desarrollo detrás ya que cada microservicio realiza pequeñas funciones.
- Testabilidad : las funcionalidades de los microservicios es reducida lo que los hace fácil de realizar tests.

Ventajas	Desventajas
• Reusabilidad	• Poca trazabilidad
• Gran Escalabilidad	• Latencia
• Flexibilidad	• Coste exponencial

Tabla 5 Ventajas y desventajas de Arquitectura de microservicios.

2.5.3 Arquitectura Hexagonal

Este tipo de arquitectura pretende desarrollar y testear la aplicación de forma aislada del framework, de la base de datos y de todo lo que está alrededor de la aplicación. Para poder aislar la aplicación se presenta una nueva estructura de directorios, esto permite por ejemplo, el desarrollo del núcleo de la aplicación mucho antes de plantear el tipo de base de datos que se va a usar.

La arquitectura se divide en 3 capas: Infraestructura, Aplicación y Dominio. Cada capa solo se puede comunicar con la siguiente por lo que esta Arquitectura también es conocida como la arquitectura de puertos y adaptadores[22].

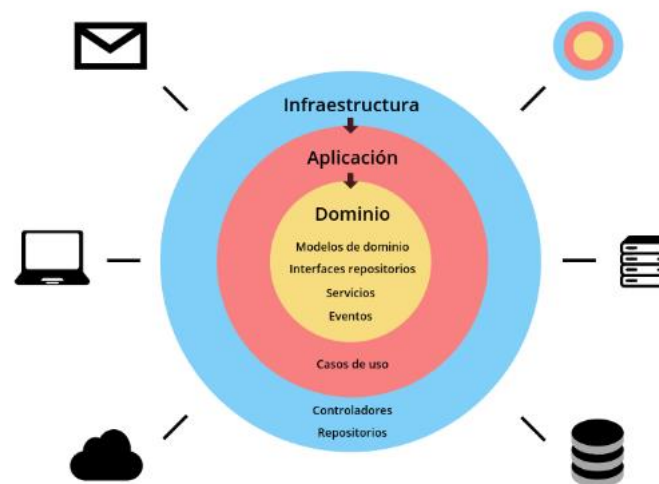


Figure 2 Representación Arquitectura Hexagonal Fuente: <https://Fabio-schetinno.medium.com>

Características de las diferentes capas:

- Infraestructura: capa más cercana al framework y se encarga de traducir todo lo que está por fuera de la aplicación como la base de datos o API's a través de los controladores.
- Aplicación: capa que va a contener todas las acciones o servicios que pueda realizar la aplicación

- Dominio: núcleo del sistema que establece como deben de comunicarse las otras capas con este.

Ventajas	Desventajas
• Independencia de BBDD	• Aumenta la complejidad del sistema
• Gran Testabilidad	• Necesidad documentación guía
• Fácil mantenimiento	

Tabla 6 Ventajas y desventajas de Arquitectura Hexagonal.

2.6 Metodologías de trabajo

Una vez analizadas las posibles tecnologías a utilizar en el proyecto, es importante tener en cuenta las diferentes metodologías que existen para trabajar. Las metodologías de trabajo son muy importantes para llevar a cabo del desarrollo de un proyecto ya que con estas se puede conseguir optimizar y mejorar los recursos de la empresa. Además, establecen las prioridades y reduce los posibles riesgos que pueda tener el proyecto[23].

Hay que tener en cuenta que existen dos tipos de metodologías de trabajo:

- Metodologías tradicionales: llevan la gestión de los proyectos en ciclos de vida que pasan por inicio, planificación, ejecución, seguimiento y cierre[24]. Su principal característica es la definición de los requisitos y metas del proyecto al inicio del proyecto no pudiendo ser modificados posteriormente.
- Metodologías Ágiles: Prioriza el trabajo en equipo sobre el coste o la planificación del tiempo. Es un proceso iterativo que se centra en el feedback y los nuevos requisitos de los clientes a lo largo de las versiones del proyecto.

A continuación, se presentaran distintas posibilidades de metodologías ágiles de trabajo.

2.6.1 Scrum

El término Scrum proviene rugby, durante los partidos el término se usa cuando se quiere reiniciar este. Esta metodología ágil de trabajo nace en 1986 en Japón de la mano de Hirotaka Takeuchi y Ikujiro Nonaka, ambos describían que Scrum era una forma de trabajar en equipo que aumentaba la flexibilidad y aceleraba el proceso de desarrollo de proyectos software.[25]

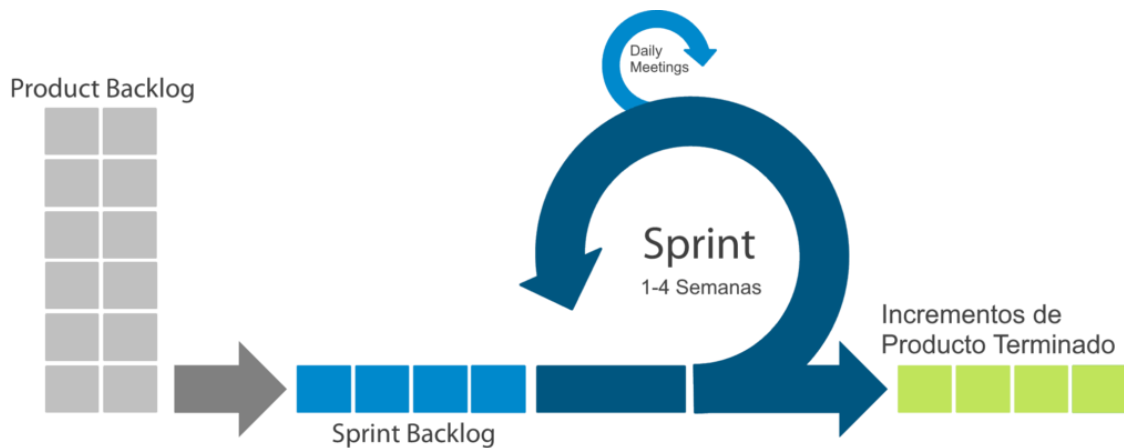


Figure 3 Esquema de trabajo de SCRUM

Fuente: <https://www.iebschool.com/>

Esta metodología permite realizar con éxito proyectos compuestos por múltiples personas de manera flexible y organizada. Scrum se compone de diferentes fases[26]:

- Planificación: llamado Product backlog, en esta fase se planifican las tareas a realizar de mayor prioridad. Con esta metodología no es necesario establecer todas las tareas a realizar desde un principio.
- Ejecución: Una vez creado el Product backlog. Se puede comenzar con el primer Sprint, esto es un tiempo determinado en el que se desarrolla un producto potencialmente entregable para el cliente. Los sprint suelen durar como máximo un mes.
- Control: Una vez llega la fecha límite de un sprint se realiza el Burn Down, es la fase donde se mide el progreso realizado durante el tiempo del Sprint.

Por último, esta metodología requiere de reuniones diarias de máximo 15 minutos llamadas Dailys donde se habla del trabajo que se está realizando y se va continuar a lo largo de ese día.

2.6.2 Kanban

La filosofía que persigue esta metodología ágil es conseguir una mejora continua a través de la extracción de las tareas de una lista que contiene acciones pendientes de ser completadas[26]. Estas listas se ven representadas en unos tableros llamados “tableros kaban” que inicialmente contienen tres columnas. Las tres columnas son formadas por “To do” que implica las tareas por hacer, “In progress” con las tareas que ya se están llevando a cabo pero aún no han sido terminadas y “Done” con las tareas que ya han sido finalizadas.

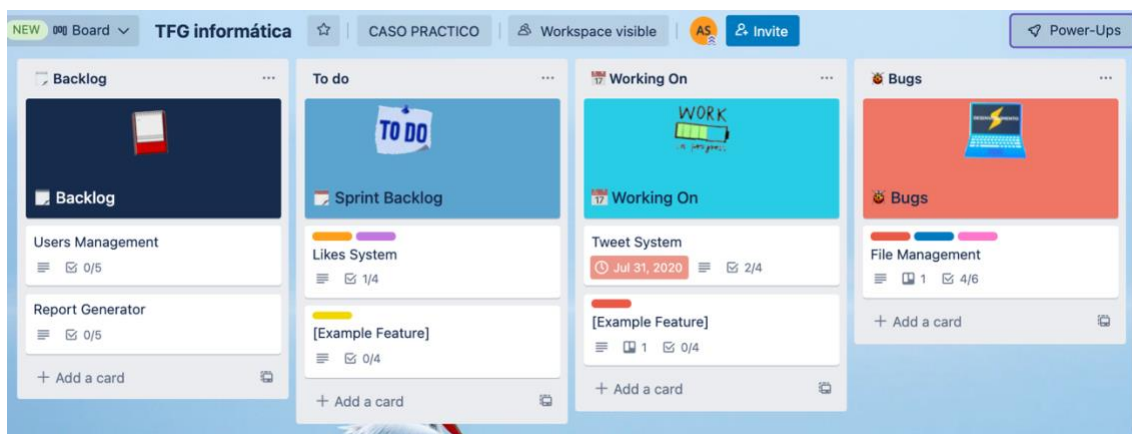


Figure 4 Tablón Kanban Fuente:

El uso de esta metodología permite al proyecto poder tener una visualización general de las acciones que se están completando, las que quedan por completar y las que finalmente ya están completadas. Esto permite que según la disponibilidad de la persona que va a iniciar una acción sepa que tareas tiene disponibles para realizar en ese tiempo y las que se están llevando a cabo hasta ese momento[28].

Más adelante en el documento se desarrollará la justificación de la opción elegida para trabajar.

3. PLANTEAMIENTO DEL PROBLEMA

En este apartado, se detalla el análisis de la situación actual existente junto con la necesidad encontrada en el mercado que este proyecto pretende subsanar.

3.1 Necesidad existente en el mercado

A partir de la pandemia mundial de la Covid-19 la sociedad aprecia de forma diferente el poder salir a comer a un restaurante. Debido a la pandemia, se llevaron a cabo restricciones que conllevaron el cierre de numerosos locales y restaurantes de forma temporal. Una vez llegado el fin del confinamiento, a medida que las restricciones se fueron disipando poco a poco, ir a comer a un restaurante/terraza se convirtió en una vía de escape de la nueva normalidad vivida por la pandemia.

En 2021 la facturación del sector de hostelería, bares y servicios aumentó un 31%[29] intentando recuperarse tras la caída creada por las restricciones de la pandemia de un 43,6%

La gran subida de ventas en el sector vino acompañada de un nuevo hábito de antelación en los clientes a la hora de la elección de un restaurante[30]. Dicha antelación, conllevó la creación de nuevas aplicaciones web para realizar reservas online además del uso de aplicaciones web de búsqueda de restaurantes.

En España, el número de habitantes celiacos se estima que está entre el 1-2% de la población. Esto implica que aproximadamente existen 500.000 personas celiacas[30]. A la hora de realizar una búsqueda de un restaurante y poder hacer una reserva para una persona celiaca, se han de tener en cuenta las restricciones que hay en cuanto a los alimentos. Para poder verificar las opciones sin gluten existe la necesidad de realizar búsquedas más exhaustivas a partir de palabras clave como “sin gluten”, “libre de gluten” o “alérgenos gluten” añadiendo el nombre del restaurante consecutivamente.

A continuación, se va a realizar el análisis sobre las distintas aplicaciones web existentes de búsqueda de restaurantes y cómo estas muestran la disponibilidad de opciones sin

gluten. El objetivo de este análisis es poder demostrar la necesidad de la aplicación que se propone en este proyecto.

3.2 Soluciones existentes en el mercado

A continuación se exponen las diferentes soluciones encontradas por orden de mayor a menor uso e impacto en los clientes. Para cuantificar el impacto y uso de las aplicaciones ordenamos las siguiente páginas las visitas realizadas a estas.

“The Fork”

Plataforma web que permite a los usuarios poder hacer reservas online en más de 80.000 restaurantes en diferentes países. Esta aplicación permite beneficiarse de promociones y descuentos importantes que ayudan a promocionar a los restaurantes pudiendo así tener mayor visibilidad.



Figure 5 The fork Fuente: Thefork.es

En 2014 TripAdvisor, empresa de reseñas turísticas, compró ElTenedor pasándose a llamar thefork.[32]

TheFork, también conocido como ElTenedor nació en 2006 de la mano de un equipo compuesto por tres socios franceses y un español. “La Fourchette” se lanzó en 2006 en Francia y un año después llegó “ElTenedor” en España. En tan solo 4 años estos cuatro

socios consiguieron convertir una pequeña startup en una empresa de referencia a nivel mundial.

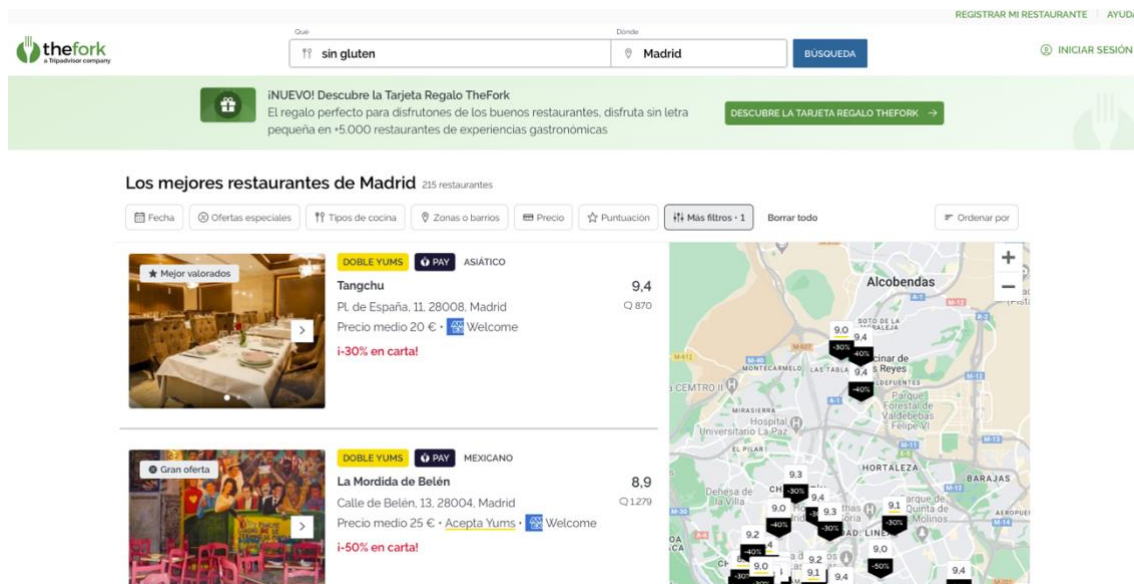


Figure 6 The fork búsqes

Fuente: The fork.es

La principal ventaja que ofrece Thefork es la posibilidad de añadir reseñas en los restaurantes que ofrece el servicio de reservas. Enfocándonos en la utilidad de esta plataforma para encontrar restaurantes sin gluten, estas reseñas son clave. Haciendo la prueba de filtrar por “sin gluten” en su buscador con localización en Madrid encontramos 125 restaurantes disponibles. Por otro lado, investigando más en profundidad los restaurantes obtenidos de la búsqueda, no indican implícitamente tener opciones sin gluten. El término sin gluten es solo mencionado en los comentarios hechos por los clientes.

La desventaja de esta plataforma se centra en el menú disponible de los restaurantes, ya que este no incluye los alérgenos, lo que no permite identificar a una persona celiaca qué puede comer realmente.

Celicidad. Comer sin gluten

App gratuita que permite a través de la localización encontrar los restaurantes sin gluten más cercanos. Esta app permite localizar más 1900 restaurantes que contienen opciones sin gluten en España.[33]



Figure 7 Celicidad Fuente Google Play

Esta aplicación fue creada en 2015 por Juan Luis Quirós un informático cacereño que tras descubrir que su pareja era celiaca decidió crear una aplicación donde se pudieran ver restaurantes que ofreciesen opciones sin gluten[34].

Las ventajas encontradas de esta aplicación:

- Permite los usuarios registrados la posibilidad de recomendar nuevos restaurantes.
- Los usuarios pueden subir fotos de sus visitas a los restaurantes además de reseñas. Adicionalmente, los usuarios más activos reciben descuentos y premios a través de la aplicación

Las desventajas encontradas de esta aplicación:

- Actualización de los restaurantes de forma manual por los creadores de la aplicación, por lo que no hay interacción directa con los restaurantes
- No incluye el menú de los restaurantes por lo que no permite conocer las opciones ofrecidas.
- Restricción de uso de la aplicación solo a través del móvil. Además de la limitación de descarga a través de Google Play por lo que no es accesible a todos los sistemas operativos para los clientes.

CeliCity

App gratuita móvil disponible para Android y IOS que muestra restaurantes y comercios donde se venden productos sin gluten



Figure 8 Celicity Fuente: Celicity.com

CeliCity nace hace 5 años como una red social para celíacos. La idea inicial del proyecto era crear una web donde se pudiesen encontrar alimentos sin gluten difíciles de encontrar en los supermercados. Tras darle varias vueltas a la idea, se decidió crear una app móvil donde los usuarios pudieran añadir reseñas de los restaurantes o de los productos vendidos en los establecimientos.

Esta aplicación, además de ser un buscador de restaurantes y establecimientos, ofrece el servicio de suscripción mensual llamada “BoxFree” que contiene una selección de

productos sin gluten diferentes cada mes[35]. Actualmente en España cuenta con más de 25.000 usuarios.

Las desventajas encontradas:

- A pesar de estar disponible tanto para sistema operativo IOS como Android, al igual que Celicidad, es una aplicación que restringe el uso solamente al móvil.
- Los restaurantes y los establecimientos no incluyen los productos sin gluten que ofrecen

3.4 Alcance del proyecto

Tras haber realizado el estudio de las diferentes soluciones disponibles en el mercado procederemos a presentar las funcionalidades del sistema de este proyecto.

El nombre del sistema será “¿Qué hay sin gluten?” consiste de un buscador web de opciones si gluten en restaurantes. Más en detalle la principal idea es que distintos restaurantes podrán añadir las opciones sin gluten que tienen en sus cartas. Para que posteriormente los clientes :

- Buscando el nombre del restaurante: los clientes podrán visualizar las opciones sin gluten existentes.
- Buscando por el nombre del plato: Los clientes podrán visualizar que restaurantes pueden visitar que tengan esas opción en concreto.
- Buscando por el nombre tipo de categoría del restaurante: Los clientes podrán visualizar que restaurantes tienen ese tipo de categoría.

Este sistema esta disponible para cualquier usuario :

- Clientes: Pueden acceder sin ningún tipo de restricción a la página para realizar búsquedas.

Para poder registrar un restaurante en el sistema es necesario que un usuario se registre para poder añadirlos.

3.5 Obstáculos y restricciones existentes

Teniendo en cuenta que este trabajo ha sido desarrollado como prueba de concepto, el sistema no será accesible desde internet sino que se utilizará el dominio localhost del ordenador destinado al desarrollo del proyecto. Esto implica que aplicación web será ejecutada únicamente en su totalidad desde este ordenador.

Las especificaciones del ordenador con el que se ha realizado el proyecto son:

Modelo	Macbook Pro 13'' Finales 2015
Especificaciones	• 8 GB RAM • 500GB Memoria Flash • Intel Iris Graphics 6100 1536 MB • Procesador : 2,9 GHz Intel Core i5 de doble núcleo
Sistema operativo	MacOS Big Sur

Tabla 7 Especificaciones ordenador Fuente: Propia

El tiempo que ha llevado desarrollar este trabajo ha sido de 8 meses aproximadamente con una dedicación diaria de 2 a 3 horas. Este proyecto ha tenido que ser compatibilizado con otros proyectos que han comprometido gran parte del tiempo de trabajo por lo que es importante considerar esta dificultad agregada al proyecto.

3.6 Requerimiento de usuarios potenciales

A continuación se detallará las características idóneas que deberían tener los usuarios que harán uso de la aplicación.

Características preferibles	
Contenido	• Padecer celiacía o algún tipo de intolerancia al gluten. • Preferencia por la gastronomía sin gluten.
Uso	• Conocimiento básico tecnológico. • Edad comprendida entre los 15-60 años. • Frecuencia de visita de aplicaciones web.

Tabla 8 Características preferibles de los usuarios. Fuente: Propia

3.7 Identificación de los interesados del sistema

Además de conocer las características de los usuarios potenciales es importante conocer las personas involucradas en el sistema tanto a nivel externo como a nivel interno.

En la siguiente tabla se identifican las tareas además de los responsables interesados en el sistema también llamados stakeholders.

Stakeholders	Función
Internos	
Jefe de proyecto	Persona que se encarga de planificar, gestionar y decidir en todas las fases del proyecto.
Analista	Persona encargada de recopilar información del cliente para desarrollar unas especificaciones que satisfagan sus necesidades.
Programador	Encargados del desarrollar e implementación software.
Diseñador	Encargado de desarrollar la interfaz teniendo en cuenta las peticiones del cliente y las especificaciones del software.

Pruebas	Encargados de llevar realizar un plan de pruebas que compruebe correctamente el funcionamiento de la aplicación.
Externos	
Cliente	La empresa para la que se desarrollar la aplicación web. [En este caso el propio alumno hará el papel de empresa]
Inversores	Empresa/s interesada en invertir en la aplicación queriendo ampliar esta misma para potenciar su uso.

Tabla 9 Stakeholders Fuente: Propia

Nota: El papel de todos los stakeholders los asumirá el propio alumno que lleva a cabo este trabajo de fin de grado.

3.8 Estimación de recursos

En esta sección se estiman los recursos necesarios para poder llevar a cabo este proyecto. Teniendo en cuenta las herramientas a utilizar y el tiempo llevado a cabo.

3.8.1 Procedimiento de estimación de recursos

Los recursos necesarios estimados para la elaboración del proyecto se muestran en la siguiente tabla[36]:

Tipo	Estimación de recursos necesarios
Software	El desarrollo del proyecto usará herramientas con licencias gratuitas o otorgadas por ser estudiante universitario Herramientas gratuitas: Zotero Trello Laravel MySQL Workbench

	Servidor interno PHP Herramientas con licencia de estudiante: Office 365
Hardware	Necesidad de un ordenador con acceso a internet además de un monitor adicional. Hardware deseado : Macbook Pro 13’’ 8GB RAM Monitor Benq 27’’
Personal	Como mencionado anteriormente el trabajo es realizado en este proyecto es individual por lo que las funciones desempeñadas por los siguientes stakeholders : • Jefe de proyecto • Analista • Programador • Diseñador • Pruebas Serán todas llevadas a cabo por la estudiante que desarrolla el trabajo de fin de grado.

Tabla 10 Estimación de recursos .Fuente: Propia

4. ANÁLISIS DEL SISTEMA

Este apartado contienen el análisis de los requerimientos y requisitos del sistema

4.1 Casos de uso

Este apartado contienen los casos de uso del sistema. Los casos de uso son una secuencia de acciones que un sistema realiza dando lugar a un resultado de valor observable[37]. Existen diferentes relaciones dependiendo de las interacciones de los diferentes actores del sistema con el propio sistema, estas son:

- Include: implica que para la realización de una acción depende de la realización de otra por lo que la segunda acción es parte necesaria de la primera.
- Extend: implica que para la realización de una acción puede depender o no para realizar otra. Creando así una implicación opcional.

A continuación, los diferentes casos de uso dividido por categorías en los que su nomenclatura será CU-XX:

4.1.1 Relacionado con el registro de Usuario

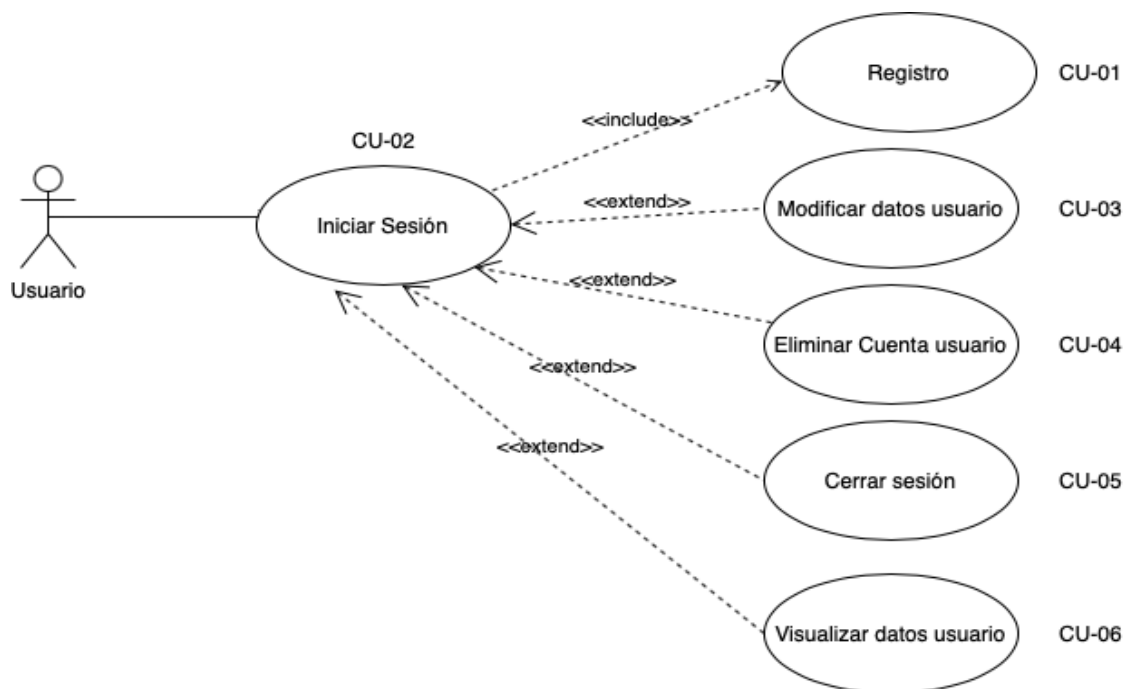


Figure 9 Casos de uso de Usuario.

4.1.2 Relacionado con el registro de Restaurantes

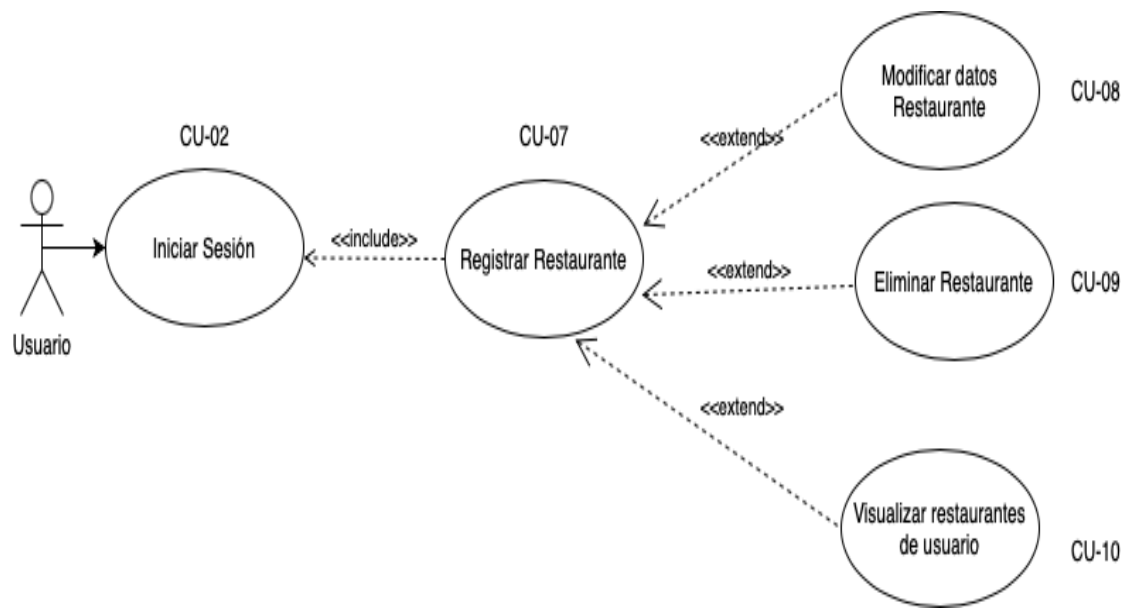


Figure 10 Casos de uso de Restaurante.

4.1.3 Relacionado con la creación de Platos

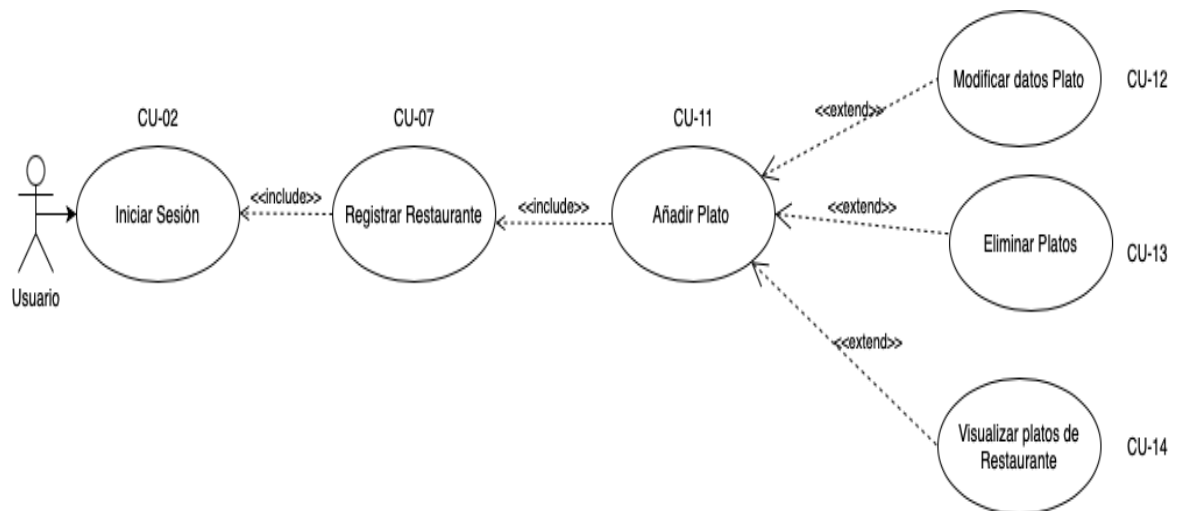


Figure 11 Casos de uso de Platos.

4.1.4 Relacionados con la búsqueda realizada por el usuario

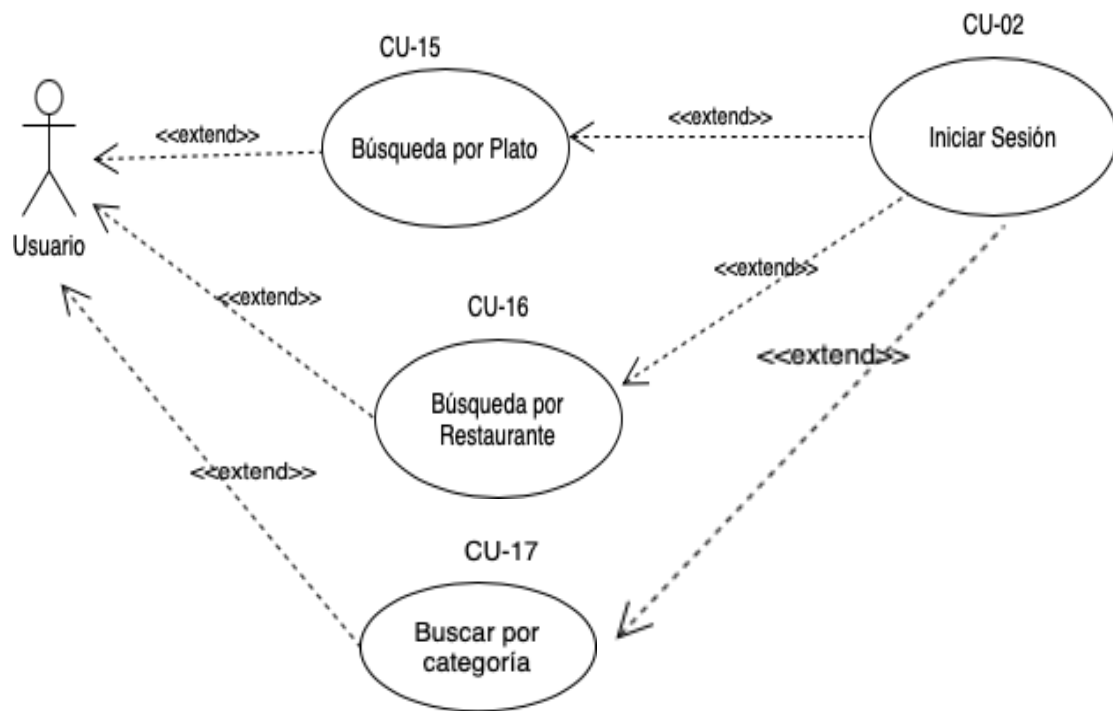


Figure 12 Casos de uso de búsqueda

4.2 Análisis de requerimientos

Para poder desarrollar un proyecto software con éxito, es necesario realizar un análisis en profundidad de las necesidades tecnológicas de la empresa o entorno cliente[38].

El análisis de requerimientos pretende:

- Especificar las características operacionales del sistema.
- Estudio de la necesidad tecnológica.
- Tener en cuenta todas las áreas de trabajo involucradas.
- Describir un plan de proyecto.
- Cumplir con las fechas previstas de entrega de versiones.

4.2.1 Análisis de requisitos

Otro punto a tener en cuenta son los requisitos software. Estos son una descripción de las restricciones y servicios de un sistema software[39]. Esto se traduce en lo que el software debería hacer teniendo en cuenta las circunstancias en las que se encuentra.

Para ellos llevaremos a cabo un análisis de requisitos. El análisis de requisitos es una de las tareas de la Ingeniería de requisitos y permite determinar la calidad del sistema.

Las directrices que debe seguir la especificación de requisitos según el estándar IEEE Std 830 son [40] :

- **Completos:** Todos los requisitos relevantes deben ser incluidos y es necesario añadir las posibles respuestas del sistema tanto como si son válidos como sino.
- **No ambiguos:** Los requisitos solo pueden tener una interpretación.
- **Consistente:** Los requisitos no pueden contradecirse ya que no serían implementables.
- **Verificable:** Todo requisito tiene que ser verificable. Un requisito es verificable cuando existe una prueba a realizar al sistema que pueda comprobar que cumple el requisito.
- **Modificable:** Los requisitos deben ser fáciles de poder modificar.

- **Trazables:** Se tiene que conocer el origen de cada requisito para hacer referencia al diseño implementado.
- **Clasificados:** Los requisitos pueden ser ordenados por importancia o por estabilidad.

Este estándar ha sido diseñado por el instituto de Ingenieros Eléctricos y Electrónicos. Esta es una asociación reconocida a nivel mundial dedicada a la normalización y estandarización en áreas técnicas[37].

4.2.1.1 Requisitos de Usuario

Siguiendo las directrices de el IEEE, es necesario definir una plantilla para los requisitos a describir. Los atributos que la comprenden son:

- **Identificador:** Código identificador de el requisito de manera unívoca. El id tendrá esta estructura UGS-NN:
 - U: Indica requisito de usuario.
 - G: Indica que es un requisito general.
 - S: Admitiría lo valores:
 - C: Requisito de capacidad.
 - R: Requisito de restricción.
 - NN: Número que identifica el requisito.
- **Nombre:** Identifica el requisito de manera breve.
- **Prioridad:** Se clasificara con los siguientes valores:
 - Alta:
 - Media:
 - Baja:
- **Necesidad:** Se clasificara con los siguientes valores:
 - Esencial
 - Deseable
 - Opcional
- **Claridad:** Se clasificara con los siguientes valores:
 - Alta
 - Media

- Baja
- Estabilidad: Describe la duración del requisito.
- Descripción: Explicación del requisito.
- Fuente: Se clasificara con los siguientes valores según quien lo pidiera:
 - Cliente
 - Analista
- Verificabilidad: Se clasificara con los siguientes valores:
 - Alta
 - Media
 - Baja

La plantilla a utilizar con los atributos mencionados anteriormente es la presentada a continuación:

Identificador:	
Nombre:	
Prioridad:	Fuente:
Necesidad:	
Claridad:	Verificabilidad:
Estabilidad:	
Descripción:	

Tabla 11 Plantilla de requisitos software. Fuente: Propia

4.2.1.2 Requisitos Funcionales

Requisitos relacionados con el usuario:

Identificador: UGC-01	
Nombre: Registro usuario	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir registrar a un nuevo usuario. Para ello es necesario un nombre y un correo electrónico que no hayan sido registrados, junto con una contraseña.	

Tabla 12 Requisito software 01.

Identificador: UGC-02	
Nombre: Inicio de sesión	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir iniciar sesión a usuarios que han sido registrados previamente. Permitiéndoles acceder a su área personal.	

Tabla 13 Requisito software 02.

Identificador: UGC-03	
Nombre: Cierre de sesión	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir, previo inicio de sesión, cerrar sesión a los usuarios.	

Tabla 14 Requisito software 03.

Identificador: UGC-04	
Nombre: Eliminar usuario	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir, previo inicio de sesión, eliminar la cuenta de un usuario.	

Tabla 15 Requisito software 04.

Identificador: UGC-05	
Nombre: Visualizar datos usuario	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema deber permitir ver los datos del usuario habiendo este iniciado sesión previamente.	

Tabla 16 Requisito software 05.

Identificador: UGC-06	
Nombre: Modificación de usuario	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe modificar los datos de usuario. El usuario deberá haber iniciado sesión previamente.	

Tabla 17 Requisito software 06.

4.2.1.3 Requisitos relacionados con las búsquedas:

Identificador: UGC-07	
Nombre: Búsqueda por Plato	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir realizar búsquedas por nombre de plato en la barra principal de la página. Esta búsqueda devolverá la lista de restaurantes que contengan platos que cumplan con la petición de la búsqueda.	

Tabla 18 Requisito software 07.

Identificador: UGC-08	
Nombre: Búsqueda por Restaurante	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir realizar búsquedas por nombre de restaurante en la barra principal de la página. Esta búsqueda devolverá la lista de restaurantes que cumplan con la petición de la búsqueda.	

Tabla 19 Requisito software 08.

Identificador: UGC-09	
Nombre: Búsqueda por Categoría	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir realizar búsquedas por la categoría de restaurante en la barra principal de la página. Esta búsqueda devolverá la lista de restaurantes que cumplan con la petición de la búsqueda.	

Tabla 20 Requisito software 09

4.2.1.4 Requisitos relacionados con los restaurantes:

Identificador: UGC-10	
Nombre: Registro de Restaurante	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir una vez iniciado sesión, registrar un restaurante con un nombre, CIF, dirección, contacto y página web	

Tabla 21 Requisito software 09.

Identificador: UGC-11	
Nombre: Modificación de Restaurante	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir una vez iniciado sesión, la modificación de un restaurante pudiendo modificar todos los datos menos el campo de CIF	

Tabla 22 Requisito software 10.

Identificador: UGC-12	
Nombre: Visualizar Restaurante	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir, previo inicio de sesión, visualizar los restaurantes registrados por el usuario.	

Tabla 23 Requisito software 11.

Identificador: UGC-13	
Nombre: Eliminar Restaurante	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir al usuario, previo inicio de sesión, eliminar un restaurante que ha sido registrado previamente	

Tabla 24 Requisito software 12.

4.2.1.5 Requisitos relacionados con los platos:

Identificador: UGC-14	
Nombre: Registrar plato	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir una vez iniciado sesión y habiendo registrado un restaurante, registrar un plato asociado.	

Tabla 25 Requisito software 13.

Identificador: UGC-15	
Nombre: Visualizar plato	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir una vez iniciado sesión, visualizar los restaurantes que ha registrado previamente junto los platos asociados al restaurante.	

Tabla 26 Requisito software 14.

Identificador: UGC-16	
Nombre: Modificar plato	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir una vez iniciado sesión, modificar los platos asociados a los restaurantes registrado previamente.	

Tabla 27 Requisito software 15.

Identificador: UGC-17	
Nombre: Eliminar plato	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir una vez iniciado sesión, visualizar los restaurantes que ha registrado previamente junto los platos asociados al restaurante.	

Tabla 28 Requisito software 16.

4.2.2 Requisitos no Funcionales

Los siguientes requisitos representan una serie de condiciones y restricciones que debe cumplir el sistema

Identificador: UGR-01	
Nombre: Navegar en la web	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe permitir al usuario navegar por la página sin la necesidad de haber iniciado sesión	

Tabla 29 Requisito software no funcional 01.

Identificador: UGR-02	
Nombre: Compatibilidad de navegadores	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe funcionar en los navegadores de Google Chrome, Mozilla Firefox y Safari.	

Tabla 30 Requisito software no funcional 02.

Identificador: UGR-03	
Nombre: Persistencia de datos	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: El sistema debe conectarse con una base de datos para la persistencia de la información	

Tabla 31 Requisito software no funcional 03.

Identificador: UGR-04	
Nombre: Actualización de datos	
Prioridad: Alta	Fuente: Cliente
Necesidad: Esencial	
Claridad: Alta	Verificabilidad: Alta
Estabilidad:	
Descripción: La información de contiene el sistema en la base de datos debe ser persistente con la que muestra en la interfaz.	

Tabla 32 Plantilla de requisitos software 04.

5. JUSTIFICACIÓN DE LA SOLUCIÓN

5.1 Elección de las tecnologías utilizadas

Este apartado pretende explicar la elección de las tecnologías que se usarán para llevar a cabo la implementación del proyecto.

Empezando por la parte del BackEnd se hará uso de PHP ya que se trata de un lenguaje altamente popular por ser destinado principalmente para el desarrollo de aplicaciones web. Este lenguaje permite ser incrustado en HTML lo que facilita la integración de backend con el front end del que se hablará posteriormente. PHP al ser tan popular cuenta con una gran comunidad de desarrolladores lo que hace que haya una gran cantidad de información, guías o referencias accesibles que son de gran utilidad a la hora de tener algún error a lo largo del desarrollo[41].

Continuando con el BackEnd, PHP cuenta con diferentes frameworks destinados para el desarrollo de aplicaciones. Estos, permiten simplificar el uso de patrones de arquitectura como el Model-View-Controller que entraremos en detalle más adelante en el documento. Además los frameworks permiten desde automatizar tareas que son repetitivas a facilitar el desarrollo de pruebas.

El framework que se utilizará será Laravel. Las principales características por las que se llevo a su elección son [42]:

- Gran cantidad de documentación disponible además de una sólida comunidad de usuarios
- Marco que proporciona alta seguridad en sus proyectos.
- Compatibilidad con diferentes bases datos.
- Compatible con el web server que posee la version de PHP 5.4 embebido.

Laravel además cuenta con “Artisan” esto es una extensión de linea de comandos a descargar. Artisan no solo proporciona una gran cantidad de comandos que ayudan a construir la aplicación sino que permite crearlos pudiendo así automatizar tareas ejecutando tan solo una línea. Por otro lado permite realizar “migraciones” conectando con la base de datos esto permite cualquier tipo de creación modificación o eliminación

de datos desde el código PHP en distintas clases. Por último, laravel tiene un soporte nativo para caché que te permite hacer que la aplicación se ejecute tan rápido como si estuviera en producción local[43].

Por el lado del FrontEnd las tecnologías usadas han sido HTML, JavaScript y CSS. Estas dos últimas serán utilizadas para hacer la parte del diseño tener una mejor estética y profesional. La parte del FrontEnd cuenta también con lógica desarrollada dentro de la aplicación por lo que, mencionado anteriormente, se hará uso de archivos php que permiten incrustar el código de HTML en ellos.

El editor de texto elegido para desarrollar el proyecto será Visual Studio Code. Este editor está destinado a soportar lenguajes de desarrollo web. Las ventajas por las que se ha escogido este editor son[44]:

- Integración con Git para poder llevar un buen control de versiones.
- Edición de código autocompletado y resaltado de sintaxis.
- Depuración de posibles errores de código.
- Personalización de la vista para adaptarla al gusto del usuario.

En cuanto al diseño de la base de datos, el modelo elegido para el desarrollar el proyecto será el relacional. El modelo relacional ha sido el seleccionado debido a que la estructura del diseño presenta una estructura fija y sencilla que mantiene relaciones dentro de ellas de forma uniforme. Más adelante se detallará el modelo diseñado para la base de datos.

Por último el sistema gestor de bases de datos será MySQL, este ofrece un buen rendimiento además de una gran cantidad de documentación que hace su uso más sencillo a la hora de encontrar posibles fallos al crear la base de datos.

5.2 Elección de la Metodología utilizada

Finalmente, para el desarrollo de este proyecto la metodología elegida ha sido la metodología Kanban[45]. Debido a que se centra en encontrar un balance en el equipo entre las tareas de trabajo que tienen que llevar a cabo y el tiempo y la disponibilidad de los integrantes.

Teniendo en cuenta que el trabajo ha sido realizado íntegramente por la estudiante que realiza el TFG, la tabla kanban ha sido de gran ayuda para tener claro puntos como:

- Tareas pendientes por hacer.
- Funcionalidades pendientes de realizar pruebas.
- Pruebas Fallidas a resolver.
- Tareas ya completadas.

Esto permitía tener una visión global del proyecto, así pudiendo tener en cuenta las tareas que quedaban pendientes por hacer para calcular el tiempo que conllevaría terminarlás.

A continuación una captura de la tabla Kanban creada en Trello :

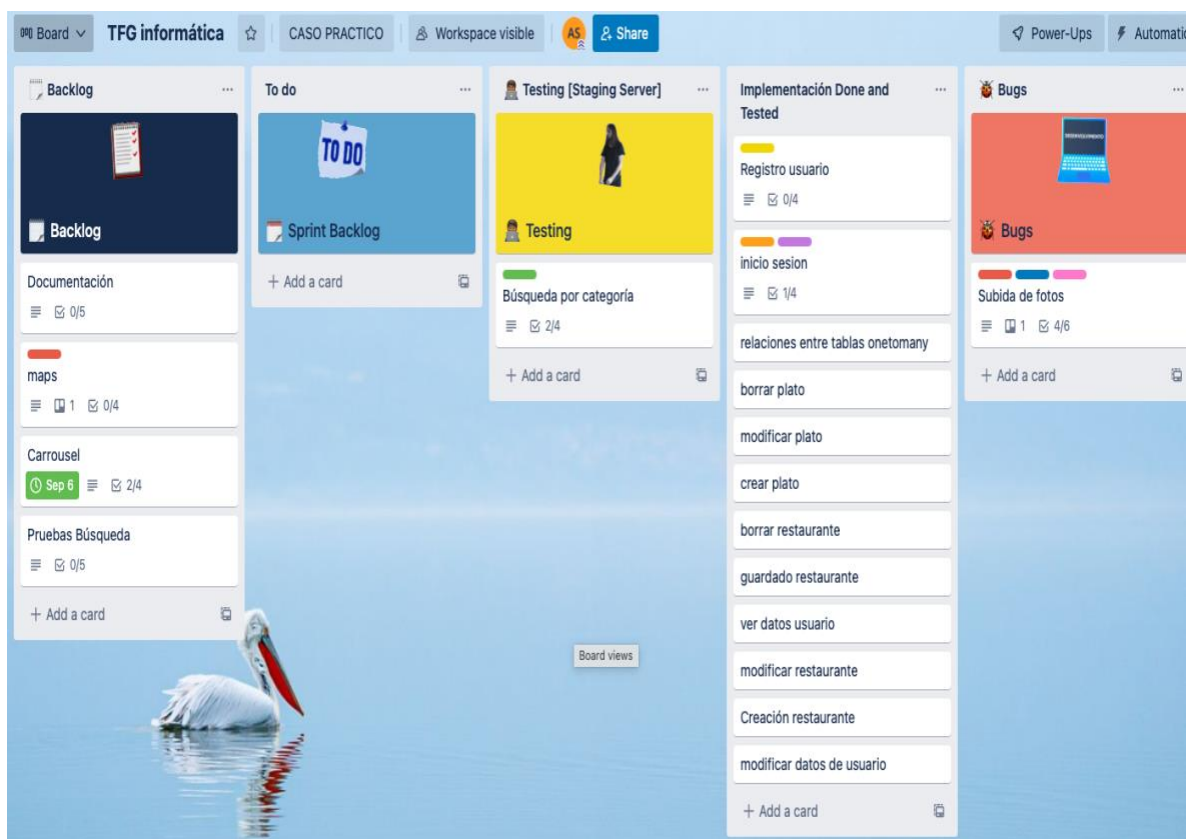


Figure Kanban Fuente : Propia creada en trello

Nota: Esta captura fue tomada en un punto intermedio del desarrollo del proyecto.

5.3 Diseño de base de datos

A continuación el esquema relacional de la base de datos creada para el proyecto.

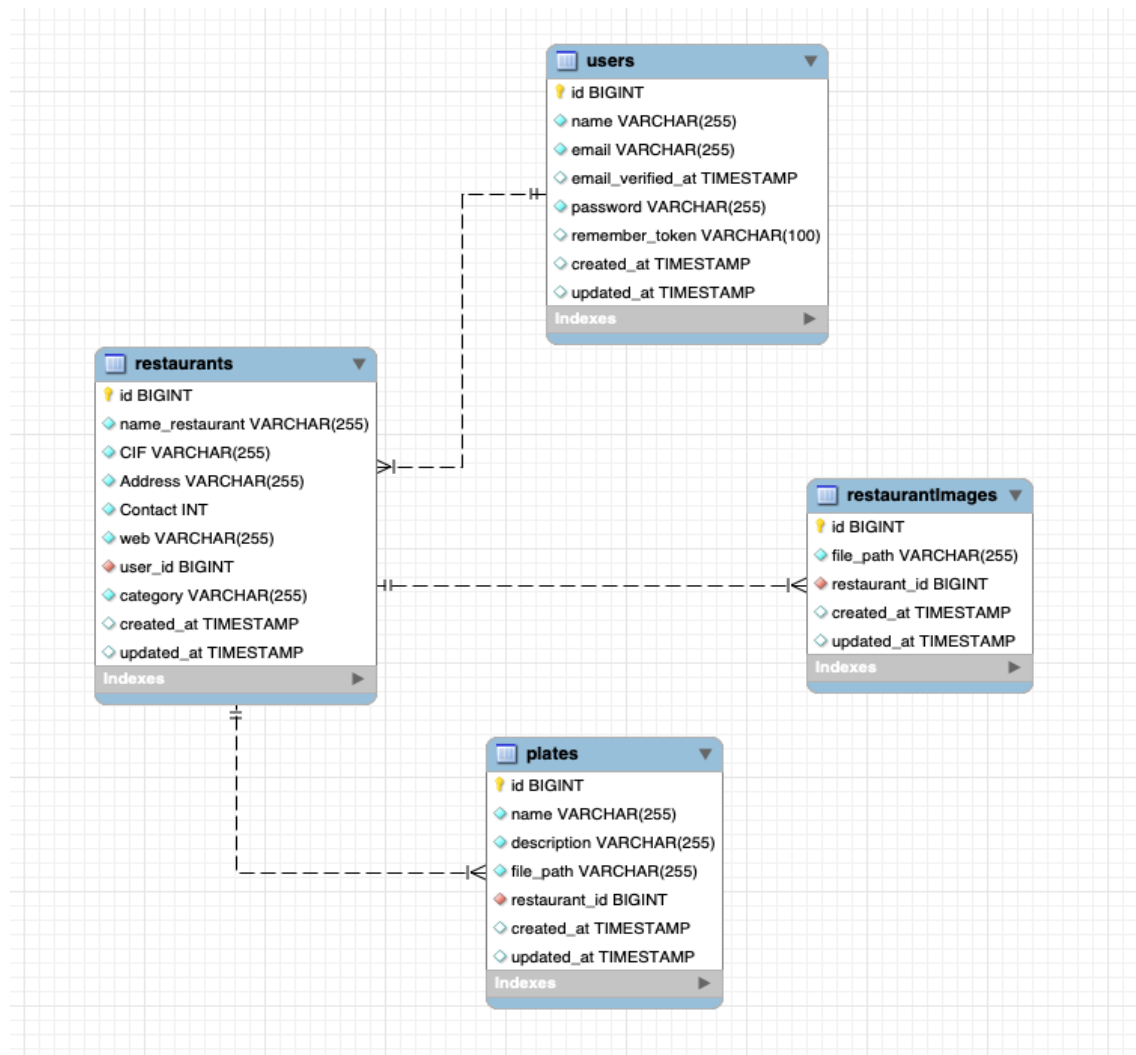


Figure 14 Esquema Relacional

Fuente : MySQL Workbench

Este esquema ha sido realizado con la herramienta de MySQL Workbench. A modo aclarativo, las líneas discontinuas entre las tablas indican las relaciones existentes entre las tablas a partir de atributos que no son ambos el primario. Por ejemplo entre la tabla users y la tabla restaurants existe una relación “one to many” es decir, un usuario puede tener numerosos restaurantes pero el restaurante tiene asociado un único usuario. La

relación esta preestablecida entre la clave primaria de la tabla users y el atributo user_id de la tabla restaurants.

La estructura de las tablas se detalla a continuación:

- Tabla Users: en esta tabla se almacenas los siguientes datos del usuario
 - Id: identificador unívoco del usuario
 - Name: nombre de usuario.
 - Email: este campo será parte de la clave primaria ya que no pueden haber dos usuarios con el mismo email
 - Password: contraseña para poder acceder a la cuenta del usuario al iniciar sesión.

- Tabla Restaurants: en esta tabla se almacenan los restaurantes que estan relacionados a los usuarios por el userID. Los atributos son los siguientes:
 - Id : identificador unívoco del Restaurante.
 - Name_restaurant: nombre del restaurante.
 - CIF: código de identificación fiscal del restaurante.
 - Address: dirección del restaurante.
 - Web: Página web del restaurante/empresa.
 - userId: identificador del usuario que ha registrado el restaurante.
 - Category: Tipo de categoría del restaurante.

La tabla Restaurants tiene una Foreign Key a la tabla Users por el atributo userId

- Tabla Plates: esta tabla contiene platos asociados a restaurantes:
 - Id: identificador unívoco del Plato
 - Name:
 - Description: breve descripción del plato
 - File_path: nombre del archivo .png de la foto del Plato.
 - Restaurant_id: identificador del restaurante en el que ha sido añadido el plato.

La tabla Plates tiene una Foreign Key a la tabla Restaurants por el atributo Restaurant_Id.

- Tabla RestaurantImages: tabla que contiene las diferentes images asociadas al restaurante:
 - Restaurant_id: identificador del restaurante en el que ha sido añadido la foto del restaurante.
 - File_path: nombre del archivo .png de la foto del restaurante.

La tabla RestaurantImages tiene una Foreign Key a la tabla de Restaurant por el atributo Restaurant_Id.

5.4 Arquitectura

La arquitectura desarrollada en la implementación es la del Model-View-Controller. Este es un patrón de arquitectura software que permite separar la parte lógica de la aplicación de la vista de esta misma[46].

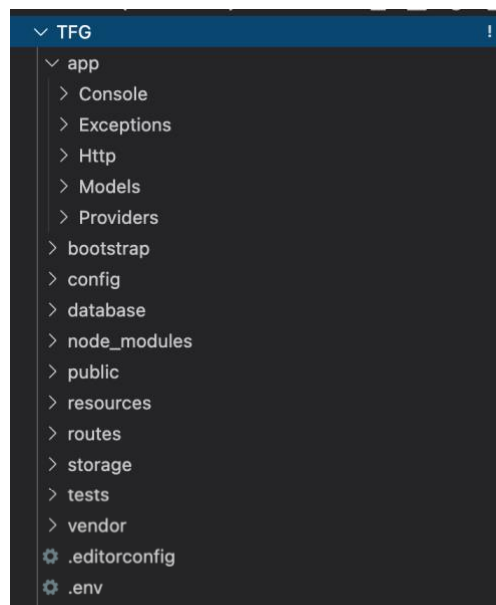


Figure Patrón MVC

En esta imagen se presenta la estructura que tiene el proyecto. Por un lado se encuentran los controladores en la carpeta de http, por otro lado el modelo se encuentra en la carpeta Models y por último la vista se encuentra en la carpeta resources. A continuación una captura de

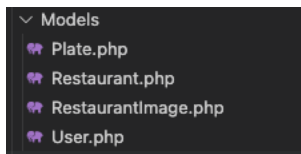


Figure Model

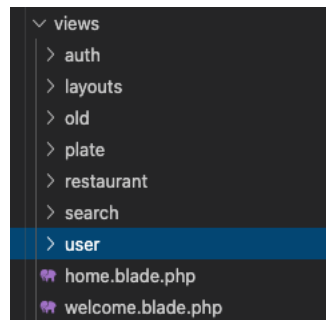


Figure View

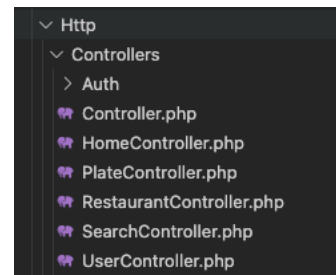


Figure Controller

¿Cómo funciona este patrón? para un mejor entendimiento se usará un ejemplo para ver su lógica.

1. El cliente o usuario realiza la petición de iniciar sesión → Rellena sus credenciales y pulsa iniciar sesión desde la vista Login.php
2. El controlador captura esta petición ya que existe una ruta que le traduce a que controlador debe dirigirse → UserController.php
3. El controlador realiza la llamada al modelo correspondiente → User.php
4. El modelo se encarga de interactuar con la bases de datos para comprobar si esta registrado en la base de datos → Tabla Users
5. El controlador recibe la información proporcionada por el modelo y la devuelve a la vista.
6. La vista redirige la aplicación a la pagina principal tras iniciar sesión correctamente → vista welcome.php

Este patrón facilita el manejo de errores y hace que el proyecto pueda ser escalable añadiendo con facilidad nuevas funcionalidades. Además permite la reutilización de sus componenetes.

5.5 Guía de usuario.

Al entrar en la página web encontramos la página principal :

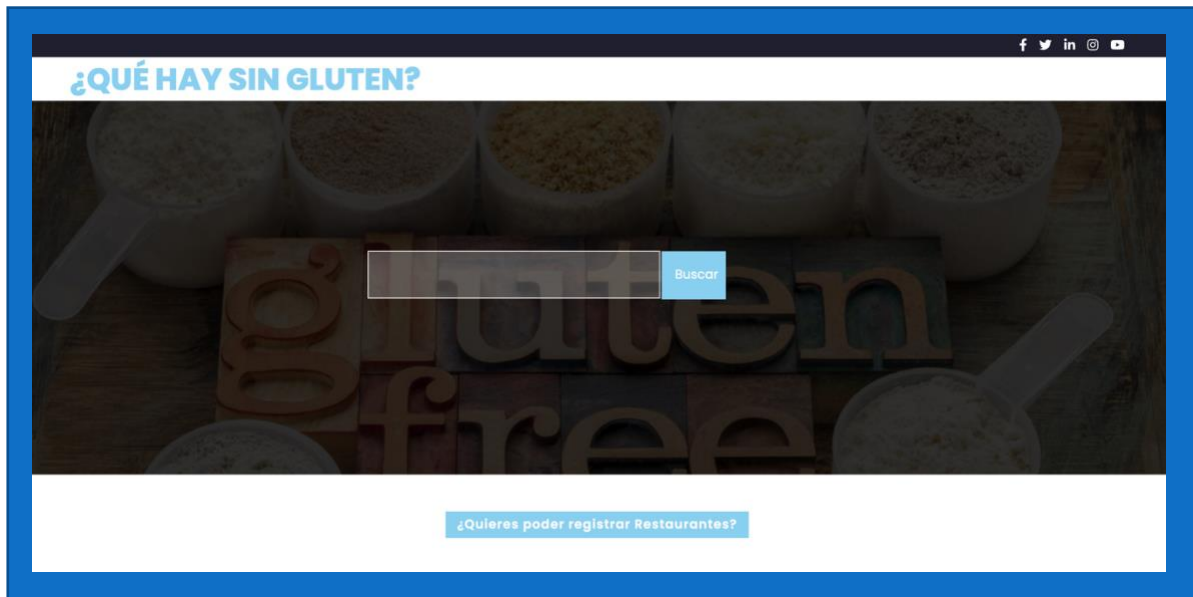


Figure home 1

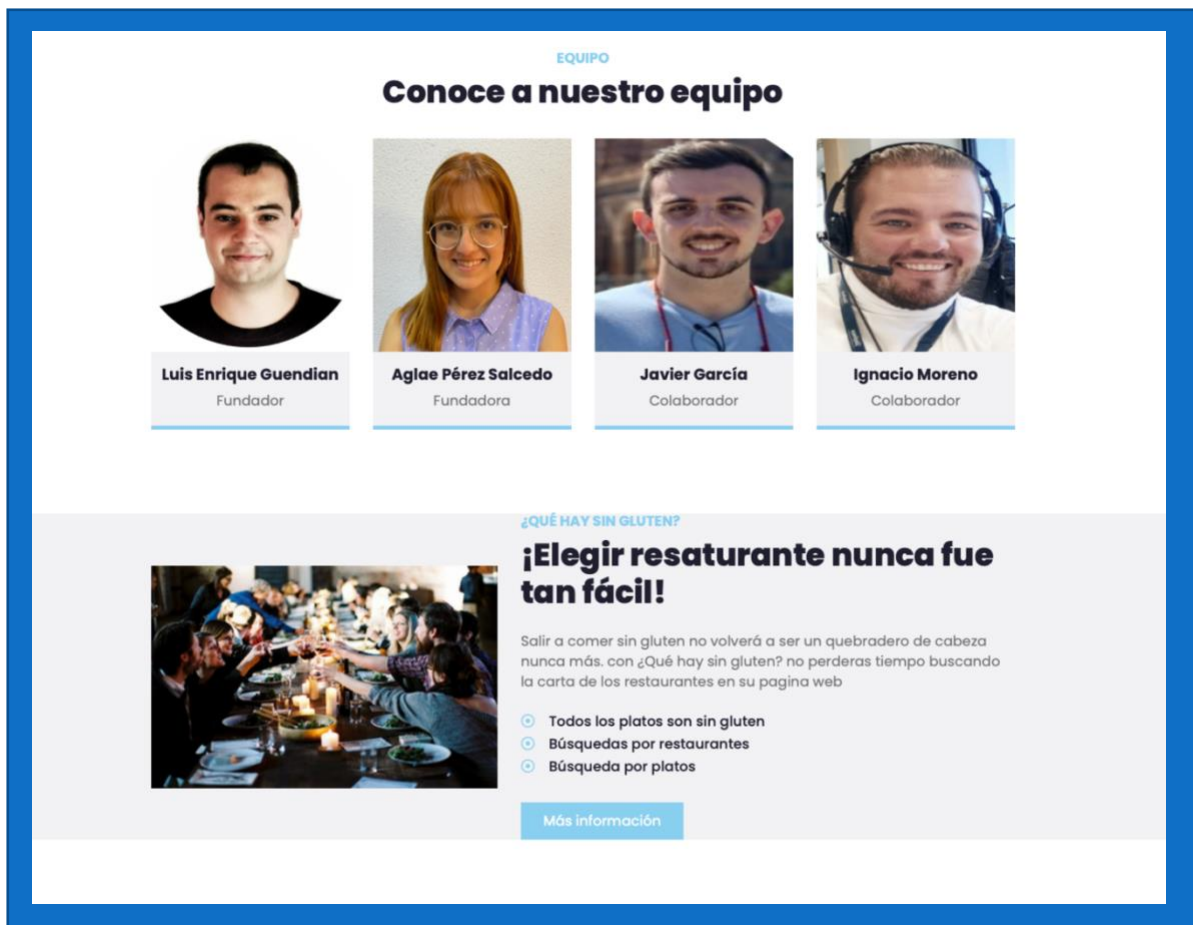


Figure home 2

En esta primera página el usuario puede navegar a través de ella sin necesidad de iniciar sesión.

Ahora bien, si un cliente quisiera registrar un restaurante en la página: el usuario deberá pinchar en ¿quieres poder registrar restaurantes?

Si el usuario tiene cuenta, debe rellenar sus credenciales y pulsar en iniciar sesión.

The screenshot shows the login page for '¿QUÉ HAY SIN GLUTEN?'. The header features the site's logo and social media icons. The main content area contains a login form titled 'Iniciar Sesión' with fields for 'Email' and 'Contraseña', a 'Recuérdame' checkbox, and buttons for 'Iniciar Sesión' and '¿No tienes cuenta? Regístrate'. The footer is divided into three sections: 'Contáctanos' with contact information, 'Links' with navigation options, and 'Suscripción' with an email sign-up form.

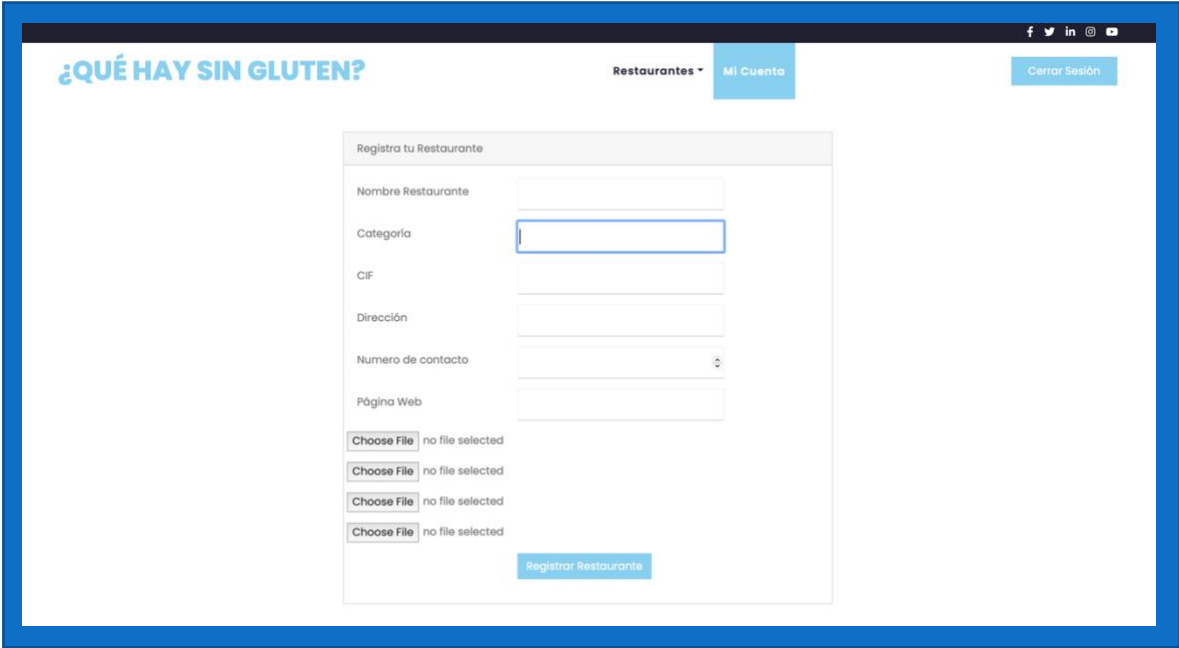
Figure login

Si el usuario no tiene cuenta, debe pulsa en “¿no tienes cuenta? Regístrate”

The screenshot shows the registration page for '¿QUÉ HAY SIN GLUTEN?'. The header features the site's logo and social media icons. The main content area contains a registration form titled 'Registrarse' with fields for 'Nombre', 'Email', 'Contraseña', and 'Confirma tu Contraseña', and a 'Registrarse' button. The footer is identical to the login page, with 'Contáctanos', 'Links', and 'Suscripción' sections.

Figure Registro

Una vez iniciado sesión el usuario puede acceder a registrar un restaurante pulsando en “Restaurantes” y luego pulsando en “Añade tu restaurante” que lo redirigirá al siguiente formulario



The screenshot shows a web form titled "Registra tu Restaurante" on the website "¿QUÉ HAY SIN GLUTEN?". The form contains the following fields: "Nombre Restaurante", "Categoría", "CIF", "Dirección", "Numero de contacto", and "Página Web". Below these fields are four "Choose File" buttons, each with the text "no file selected". At the bottom right of the form is a blue button labeled "Registrar Restaurante". The website header includes the logo, a navigation menu with "Restaurantes" and "Mi Cuenta", and a "Cerrar Sesión" button.

Figure Registro Restaurante

El usuario deberá de rellenar la información solicitada y posteriormente pulsa en Registrar Restaurante.

Tras registrar el restaurante el usuario podrá visualizar las fotos subidas en un slider o carrusel de fotos

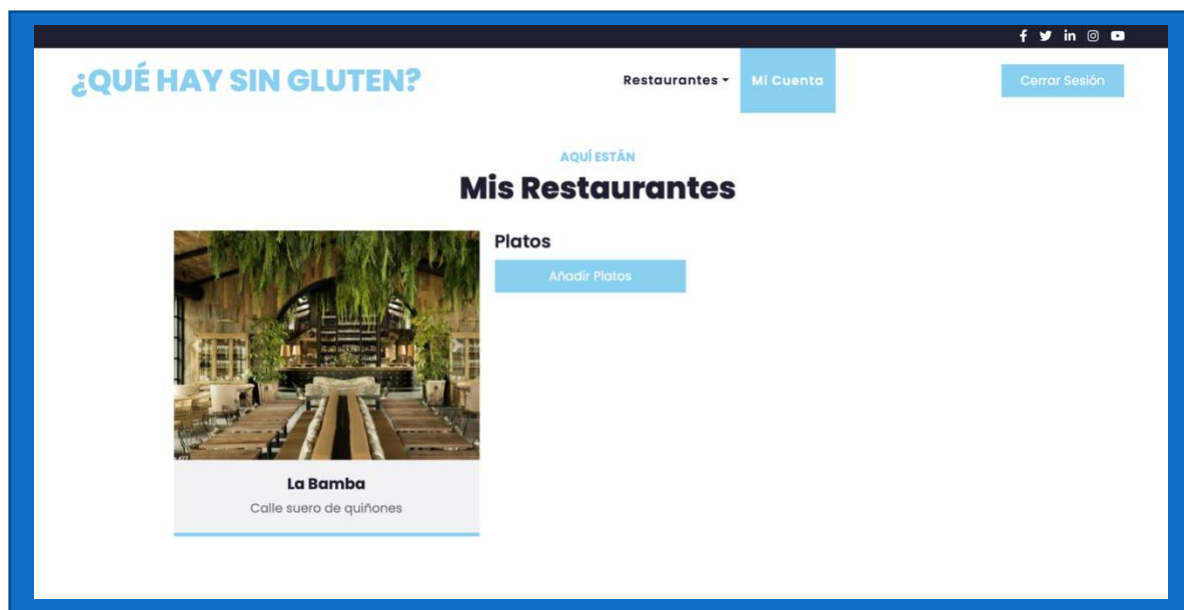


Figure inicio sesión

Si el usuario quisiera añadir un plato al restaurante este debe pulsar en añadir plato
Y después de rellenar la formación necesaria pulsar “registrar” plato.

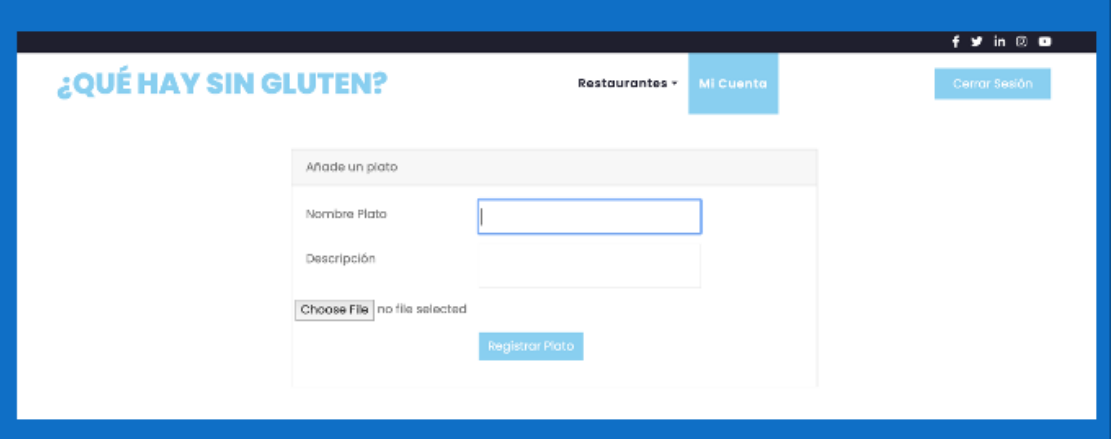
The screenshot shows the 'Añadir un plato' (Add a dish) form on the '¿QUÉ HAY SIN GLUTEN?' website. The form is titled 'Añadir un plato' and contains three input fields: 'Nombre Plato' (Dish Name), 'Descripción' (Description), and a file upload section with a 'Choose File' button and the text 'no file selected'. A 'Registrar Plato' (Register Dish) button is located at the bottom right of the form. The website header includes the logo '¿QUÉ HAY SIN GLUTEN?', navigation links 'Restaurantes' and 'Mi Cuenta', and a 'Cerrar Sesión' (Logout) button.

Figure Registrar plato

El cliente podría añadir la cantidad de platos que quisiera.



Figure visualizar Restaurante y platos

Además para modificar o eliminar la información tanto del restaurante como del plato solo sería necesario pulsar en ellos para que la página se redirigiera a un formulario donde poder modificar o eliminar la información registrada

Modificar/Eliminar datos de un restaurante:

The screenshot shows a web application interface for editing a restaurant. The header includes the logo '¿QUÉ HAY SIN GLUTEN?', navigation links for 'Restaurantes' and 'Mi Cuenta', and a 'Cerrar Sesión' button. The main form, titled 'Editando Restaurante', contains the following fields: 'Nombre Restaurante' (La Bamba), 'CIF' (Q2826000H), 'Dirección' (Calle suero de quíñones), 'Numero de contacto' (616497929), and 'Página Web' (labamba.com). There is also a 'Choose File' button with the text 'no file selected'. At the bottom of the form are two buttons: 'Editar Restaurante' (blue) and 'Borrar Restaurante' (red).

Figure Modificar/Eliminar datos Restaurante

Modificar/Eliminar datos de un plato

The screenshot shows a web application interface for editing a dish. The header is identical to the previous screenshot. The main form, titled 'Modifica tu plato', contains the following fields: 'Nombre Plato' (Pollo al curry), 'Descripción' (Con salsa picante), and a 'Choose File' button with the text 'no file selected'. At the bottom of the form are two buttons: 'Editar Plato' (blue) and 'Borrar plato' (red).

Figure Modificar/Eliminar datos Plato

Por otro lado, si un cliente no quisiera registrarse podría realizar búsquedas por el nombre de un restaurante, el nombre de un plato o la categoría del restaurante. Además si pulsa en uno de los restaurantes que salen como resultado lo llevaría a ver la información detallada de este.

6. VALIDACIÓN DEL SOFTWARE

6.1 Pruebas

Este apartado incluye los test realizados al sistema. El esquema de test será el mostrado a continuación :

Identificador: T-XX-YY
Nombre:
Requisitos contemplados:
Casos de uso :
Precondiciones:
Estado:
Descripción:
Resultado:

Tabla 33 Plantilla de test.

Los elementos que incluyen los test son los siguientes:

- Identificador: Código identificador de el requisito de manera unívoca.
 - T-XX-YY :
 - XX puede ser F de funcionales o NF de no funcionales.
 - YY indica el número de prueba.
- Nombre: Representación de una breve descripción del test.
- Requisitos contemplados: los que cubre y quedarían verificados por el test.
- Casos de uso: los que están asociados al requisito.
- Precondiciones: la situación en la que debe ejecutarse el test.
- Resultado: el esperado al ejecutar el test.
- Estabilidad: Verificado o Fallido.
- Descripción: Explicación del requisito.

6.1 Test de Requisitos funcionales

Test relacionados con el inicio de sesión:

Identificador: T-F-01
Nombre: Inicio de sesión
Requisitos contemplados: UGC-02
Casos de uso : CU-02
Precondiciones: El usuario debe haberse registrado previamente.
Estado: Verificado
Descripción: 1. El usuario pulsa el link de “Quiero registrar un restaurante”. 2. El usuario es redirigido al formulario de iniciar sesión. 3. El usuario accederá con el email y contraseña con el que se ha registrado
Resultado: El usuario tras iniciar sesión visualizará la página de “Mis Restaurantes”

Tabla 34 Test 01

Identificador: T-F-02
Nombre: Fallo en Inicio de sesión con email no registrado
Requisitos contemplados: UGC-02
Casos de uso : CU-02
Precondiciones: El email por introducir debe haberse registrado previamente.
Estado: Verificado
Descripción: 1. El usuario pulsa el link de “Quiero registrar un restaurante”. 2. El usuario es redirigido al formulario de iniciar sesión. 3. El usuario rellena el formulario con un email que no ha sido registrado.
Resultado: El usuario no puede iniciar sesión. El sistema indica un error “Las credenciales no se encuentran en la base de datos”

Tabla 35 Test 02

Test relacionados con el registro y el usuario:

Identificador: T-F-03
Nombre: Registro de un usuario con email ya registrado
Requisitos contemplados: UGC-01
Casos de uso : CU-01
Precondiciones: El email por introducir debe haberse registrado previamente.
Estado: Verificado
Descripción: 1. El usuario pulsa el link de “Quiero registrar un restaurante”. 2. El usuario es redirigido al formulario de iniciar sesión. 3. El usuario pulsa en ¿No tienes cuenta? Regístrate 4. El usuario registra un email con el que se ha registrado un usuario previamente.
Resultado: El usuario no puede registrarse. El sistema indica un error “Este email ya está registrado con otro usuario”

Tabla 36 Test 03

Identificador: T-F-04
Nombre: Visualización y Modificación de datos de usuario
Requisitos contemplados: UGC-05, UGC-06
Casos de uso : CU-03
Precondiciones: El usuario debe estar previamente registrado y haber iniciado sesión
Estado: Verificado
Descripción: 1. El usuario pulsa en “mi cuenta” 2. El usuario cambia su nombre. 3. El usuario escribe su contraseña. 4.El usuario pulsa en modificar.
Resultado: El usuario tiene otro nombre que se puede visualizar en “mi cuenta”.

Tabla 37 Test 04

Identificador: T-F-05
Nombre: Cerrar Sesión
Requisitos contemplados: UGC-03
Casos de uso : CU-05
Precondiciones: El usuario debe estar previamente registrado y haber iniciado sesión
Estado: Verificado
Descripción: 1. El usuario pulsa en “cerrar sesión”
Resultado: El usuario es dirigido a la página principal.

Tabla 38 Test 05

Identificador: T-F-06
Nombre: Eliminar cuenta de usuario
Requisitos contemplados: UGC-04
Casos de uso : CU-04
Precondiciones: El usuario debe estar previamente registrado y haber iniciado sesión
Estado: Verificado
Descripción: El Usuario: 1. Pulsa en “Mi cuenta” 2. Pulsa en “Borrar usuario”
Resultado: El usuario es redirigido a la página principal.

Tabla 39 Test 06

Test relacionados con restaurantes y platos:

Identificador: T-F-07
Nombre: Registro de un restaurante y un plato
Requisitos contemplados: UGC-10, UGC-14
Casos de uso : CU-7,CU-11
Precondiciones: El usuario debe estar previamente registrado y haber iniciado sesión
Estado: Verificado
Descripción: El Usuario: 1. Pulsa en “Añadir tu Restaurante” 2. Rellena el formulario y pulsa en “Registrar Restaurante”. 3. Visualiza la página de “Mis restaurantes” 4. Pulsa en “añadir plato” al lado del restaurante. 5. Rellena el formulario y pulsa en “Registrar Plato”
Resultado: El usuario visualiza el restaurante y el plato que ha registrado.

Tabla 40 Test 07

Identificador: T-F-08
Nombre: Eliminar un Restaurante
Requisitos contemplados: UGC-13
Casos de uso : CU-9
Precondiciones: El usuario debe estar previamente registrado y haber iniciado sesión
Estado: Verificado
Descripción: El Usuario: 1. Pulsa en “Mis restaurantes” 2. Pulsa en “Editar Restaurante” 3. Pulsa en “Borrar Restaurante”
Resultado: El usuario es redirigido a la página “Mis Restaurantes” y el restaurante y los platos asociados al restaurante han sido borrados

Tabla 41 Test 08

Identificador: T-F-09
Nombre: Modificar y visualizar datos de un Restaurante
Requisitos contemplados: UGC-11, UGC-12
Casos de uso : CU-08, CU-10
Precondiciones: El usuario debe estar previamente registrado y haber iniciado sesión
Estado: Verificado
Descripción: El Usuario: 1. Pulsa en “Mis restaurantes” 2. Pulsa en “Editar Restaurante” 3. Cambia el número de contacto del restaurante 4. Pulsa “modificar”
Resultado: El usuario es redirigido a la página “Mis Restaurantes” pudiendo visualizar el restaurante que ha sido cambiado de nombre.

Tabla 42 Test 09

Identificador: T-F-10
Nombre: Visualizar Restaurantes y Platos registrados por un usuario
Requisitos contemplados: UGC-12, UGC-15
Casos de uso : CU-10,CU-14
Precondiciones: El usuario debe: 1.Estar previamente registrado y haber iniciado sesión. 2.haber registrado al menos un restaurante y un plato asociado.
Estado: Verificado
Descripción: El Usuario: 1. Pulsa en “Mis restaurantes”
Resultado: El usuario es redirigido a la página “Mis Restaurantes” pudiendo visualizar los restaurantes y platos asociados que ha registrado previamente.

Tabla 43 Test 10

Identificador: T-F-11
Nombre: Modificar un plato y Eliminar Platos registrados por un usuario
Requisitos contemplados: UGC-16,UGC-17
Casos de uso : CU-13
Precondiciones: El usuario debe: 1.Estar previamente registrado y haber iniciado sesión. 2.haber registrado al menos un restaurante y un plato asociado.
Estado: Verificado
Descripción: El Usuario: 1. Pulsa en “Mis restaurantes” 2. Pulsa en el plato a Modificar 3.Modififca el nombre del plato y pulsa en modificar. 4. Pulsa en el plato modificado 3. pulsa “Borrar Plato”
Resultado: El usuario es redirigido a la página “Mis Restaurantes” pudiendo visualizar los restaurantes y platos asociados sin el plato borrado.

Tabla 44 Test 11

Test de requisitos de búsquedas:

Identificador: T-F-12
Nombre: Búsqueda por nombre del Plato
Requisitos contemplados: UGC-7
Casos de uso : CU-15
Precondiciones:
Estado: Verificado
Descripción: El Usuario realiza una búsqueda por el nombre de un plato
Resultado: La página devuelve los restaurantes que tengan platos con el nombre exacto o que contengan parte del nombre.

Tabla 45 Test 12

Identificador: T-F-13
Nombre: Búsqueda por nombre del Restaurante
Requisitos contemplados: UGC-8
Casos de uso : CU-16
Precondiciones:
Estado: Verificado
Descripción: El Usuario realiza una búsqueda por el nombre del restaurante
Resultado: La página devuelve los restaurantes con el nombre exacto o que contenga parte del nombre.

Tabla 46 Test 13

Identificador: T-F-14
Nombre: Búsqueda por categoría de restaurante
Requisitos contemplados: UGC-09
Casos de uso : CU-17
Precondiciones:
Estado: Verificado
Descripción: El Usuario realiza una búsqueda por categoría de restaurante.
Resultado: La página devuelve los restaurantes la categoría exacta o que contenga parte del nombre.

Tabla 47 Test 14

6.2 Test de Requisitos no funcionales

Identificador: T-NF-01
Nombre: Soporte de distintas plataformas
Requisitos contemplados: UGR-02
Precondiciones:
Estado: Verificado
Descripción: Puede acceder a la aplicación web desde los navegadores Google Chrome, Mozilla Firefox y Safari
Resultado: El usuario puede navegar por la web en los tres navegadores.

Tabla 48 Test 15

Identificador: T-NF-02
Nombre: Persistencia de datos
Requisitos contemplados: UGR-03
Precondiciones:
Estado: Verificado
Descripción: El usuario registra inicia sesión y registra un restaurante
Resultado: El sistema se conecta con la base de datos para insertar una fila en su tabla Restaurants.

Tabla 49 Test 16

Identificador: T-NF-03
Nombre: Actualización de datos
Requisitos contemplados: UGR-04
Precondiciones:
Estado: Verificado
Descripción: El usuario accede a su cuenta y modifica su nombre
Resultado: El usuario si vuelve a acceder a “mi cuenta” verá su nombre modificado.

Tabla 50 Test 17

Identificador-NF-04
Nombre: Navegar en la web
Requisitos contemplados: UGR-01
Precondiciones:
Estado: Verificado
Descripción: El usuario sin iniciar sesión accede a la web.
Resultado: El usuario sin necesidad de iniciar sesión puede navegar por la web realizando búsquedas.

6.3 Matriz de trazabilidad

La matriz de trazabilidad se realiza entre los requisitos del sistema y los test propuestos y validados para asegurar que todos los requerimientos de sistema son cubiertos por una prueba verificada.

A continuación la matriz de trazabilidad:

	T-F-01	T-F-02	T-F-03	T-F-04	T-F-05	T-F-06	T-F-07	T-F-08	T-F-09
UGC-01									
UGC-02									
UGC-03									
UGC-04									
UGC-05									
UGC-06									
UGC-07									
UGC-08									
UGC-09									
UGC-10									
UGC-11									
UGC-12									
UGC-13									
UGC-14									
UGC-15									
UGC-16									
UGC-17									
UGR-01									
UGR-02									
UGR-03									
UGR-04									

Tabla 51 Matriz Trazabilidad

	T-F-10	T-F-11	T-F-12	T-F-13	T-F-14	T-NF-01	T-NF-02	T-NF-03	T-NF-04
UGC-01									
UGC-02									
UGC-03									
UGC-04									
UGC-05									
UGC-06									
UGC-07									
UGC-08									
UGC-09									
UGC-10									
UGC-11									
UGC-12									
UGC-13									
UGC-14									
UGC-15									
UGC-16									
UGC-17									
UGR-01									
UGR-02									
UGR-03									
UGR-04									

Tabla 52 Matriz Trazabilidad

7. MARCO REGULADOR

7.1 Legislación aplicable

Esta sección incluye los aspectos que son relevantes en la legislación que se aplica a la implementación del sistema[47]. La implementación de este proyecto es una prueba de concepto por lo que no es accesible desde internet por lo que la legislación siguiente solo será aplicable si el proyecto es finalmente implementado para que se pueda acceder desde internet.

Existen cuatro normativas básicas[48]:

- **Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (LSSI)[49]:** Regula las actividades económicas que se realizan a través de internet, obligando a al prestador mostrar cierta información como:
 - Nombre o denominación social junto con datos de contacto.
 - Número de identidad fiscal(NIF/CIF)
 - Aviso legal: donde se ha de mostrar obligatoriamente:
 - Propietario o titular de la web.
 - Datos de colegiación (si aplicase)
 - Política de cookies : permitiendo aceptar o rechazarlas.
 - Métodos de pago: forma de pago, garantía de producto o servicio.
- **Real Decreto Legislativo 1/2007, de 16 de noviembre[50]:** garantiza la defensa de consumidores y usuarios creando un marco de protección alrededor de estos.
- **Reglamento (UE) 2016/679 Del Parlamento Europeo y Del Consejo de 27 de abril de 2016[51]:** Quiere contribuir en crear un espacio de justicia, libertad y seguridad en una unión económica.
- **Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.[52]:** Pretende parar la recogida indiscriminada de datos de carácter personal ya que si se recogen estos se ha de informar el uso de los datos y el periodo de conservación de estos. Dentro de esto se encuentra:
 - Permitir la visualización de la información compartida.
 - Permitir la modificación de la información compartida.

- Permitir eliminar la información compartida.
- Limitar el uso de la información compartida.
- **Ley 7/1998, de 13 de abril, sobre condiciones generales de la contratación.[53]:** obliga a explicar con todo detalle las características del proceso de compra, reserva o contratación que se realizará a través de la web
- **Ley Orgánica de Protección de Datos (LOPD), Ley 15/1999 [54]:** Existen una serie de requisitos a cumplir para toda empresa que disponga de datos de sus clientes o usuarios:
 - Informar a los usuarios de los datos recogidos y para qué se van a utilizar
 - Consentimiento del usuario mediante la aceptación de la política de privacidad.
 - Proporcionar opción de dar modificar, consultar o eliminar sus datos.
 - Medidas de seguridad: Pedir cambio de contraseña a los usuarios anualmente
 - Registro de un fichero de datos en la agencia Española de datos.

7.2 Licencias y propiedad intelectual

El sistema ha sido desarrollado con el uso de varias tecnologías/herramientas detalladas a continuación:

- Visual Studio Code : Editor de texto usado para la realización del proyecto. Su licencia ha sido gratuita[55].
- Laravel: Este framework de código abierto permite poder desarrollar aplicaciones web. Su licencia ha sido gratuita[56]. Este ha sido el framework principal de todo el proyecto. Además, se ha hecho uso de:
 - Artisan para permitir el uso de la interfaz por línea de comandos.
 - Composer: manejador de paquetes que permite instalar librerías y dependencias
- MySQL Workbench[15]: versión 8.0 Community. Sistema Gestor de Bases de datos cuya licencia de uso ha sido gratuita.
- Web server: servidor web integrado en la versión de PHP . (Licencia Gratuita)

8. GESTION DEL PROYECTO

Esta sección contiene la parte del trabajo relacionada con la gestión del proyecto. Esto, conlleva una estimación inicial de cómo se desarrollará el proyecto durante el tiempo pudiendo ver así si el proyecto es viable o poder encontrar posibles desviaciones. Anticiparse a los posibles problemas permite corregir los posibles errores antes de que ocurran.

Se incluirán la planificación para la construcción e instalación del proyecto además de la planificación temporal como debe ir progresando el proyecto.

8.1 Organización del trabajo

Cada fase del proyecto conlleva la participación de diferentes stakeholders. El fin de este apartado es crear un resumen para saber que stakeholder participará en cada fase del documento.

Fase	Stakeholders participantes
Fase 1: Inicial - Planificación	-Cliente y Jefe de Proyecto.
Reunión inicial donde se habla del proyecto a desarrollar y sus funciones básicas	
Reunión de planteamiento de requisitos : Segunda reunión donde se recogen los deseos de implementación del cliente	
Reunión de traducción de deseos del cliente a requisitos del sistema viables y que cumplan con el producto esperado.	-Jefe de proyecto y Analista
Fase 2: Media – Desarrollo	-Analista y diseñador
Reunión de viabilidad de diseño en conjunto con los requisitos estipulados por el cliente.	
Después de tener el diseño, este se da a los programadores par que vaya realizando	

versiones a probar por los encargados de pruebas.	-Programadores y pruebas
Beta: versión prueba a disposición del cliente para que pueda tener una demostración de la aplicación antes de que esté completamente terminada. Pudiendo así el cliente dar su aprobación o pedir posibles cambios	-Cliente y Jefe de proyecto
Fase 3: Final - Instalación	-Pruebas y Cliente.
Una vez el cliente aprueba el prototipo se pasa a la parte de integración con el sistema del cliente y en esta fase están encargados el personal de pruebas	
Post-integración - Mantenimiento	-Programadores
Mantenimiento posterior de la aplicación una vez integrada e instalada al cliente	

8.2 Planificación

8.2.1 Diagrama de Gantt

A continuación se muestra el diagrama de Gantt, herramienta que se usa para mostrar el tiempo de dedicación estimado a diferentes tareas a lo largo del tiempo total determinado para la realización del proyecto.

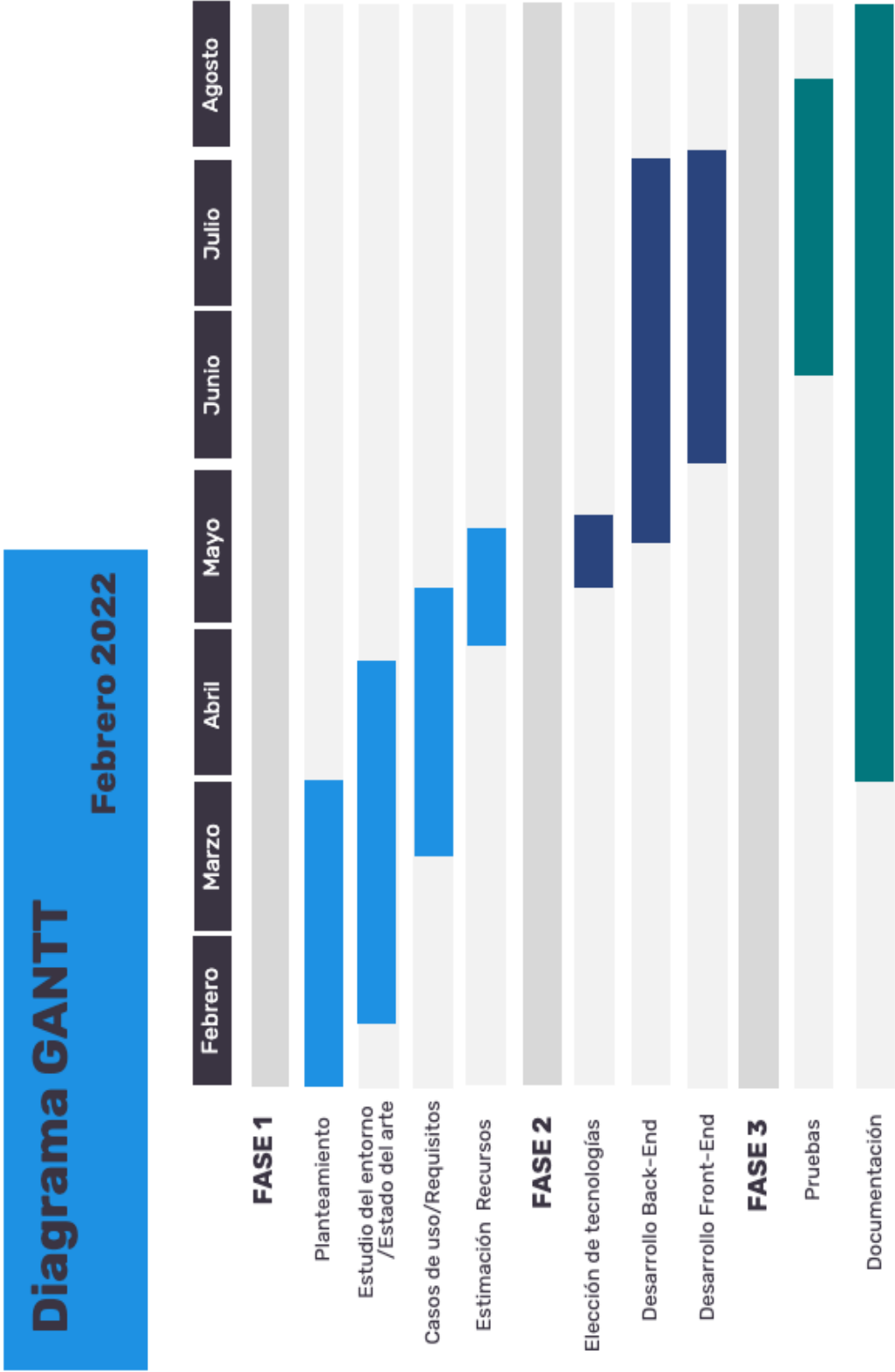


Figure X Diagrama Gannt

8.3 Control de desviaciones

Este apartado especifica cómo se gestiona el control de las posibles desviaciones del proyecto. Las siguientes acciones se llevarán a cabo para poder evitar desviaciones del plan inicial estipulado:

Id	Actividad	Encargado	Participantes
1	Reuniones Bisemanales para la comprobación del correcto seguimiento del plan	Jefe de proyectos	Todo el equipo
2	Reunión a final de semana para comentar problemas o posibles “blockers”	Analista	Analista Programadores
3	Reunión mensual con el cliente para informar del desarrollo del proyecto	Jefe de proyecto	Jefe de proyecto Cliente

Si se diese la situación de tener que realizar cambios en la planificación de alguna de las fases del proyecto se aprovecharía la actividad número 3 para informar al cliente del cambio de fechas.

9. ENTORNO SOCIO-ECONÓMICO

Una vez realizada la estimación de recursos se puede analizar los costes que serán necesario para el desempeño del proyecto. Esto incluye el coste del personal, hardware o cualquier otro tipo de recurso necesitado a partir de los otros. Además se detallara el impacto socioeconómico estimado que tendrá en los clientes finales.

9.1 Presupuesto

Este proyecto ha sido desarrollado en un plazo de 8 meses que estaría comprendidos entre inicios desde Febrero de 2022 y finales de Agosto de 2022. Teniendo en cuenta la duración del proyecto, podemos desglosar el trabajo mensual y diario realizado:

Mes	Horas invertidas
Febrero	30 horas
Marzo	50 horas
Abril	50 horas
Mayo	40 horas
Junio	40 horas
Julio	80 horas
Agosto	80 horas
Total	370 horas

Tabla 53 Horas invertidas Fuente: Propia

9.1.1 Recursos Software

Nombre del producto software	Tipo de Licencia	Precio
Zotero	Libre	0€
Trello	Libre	0€
Laravel	Libre	0€
MySQL Workbench	Libre	0€
Servidor interno Laravel	Libre	0€
Office 365	Estudiante	0€
Total	-	0€

Tabla 54 Recursos Software Fuente: Propia

9.1.2 Recursos Hardware

Para poder calcular el coste de los recursos tangibles es necesario tener en cuenta la amortización de estos.

Nombre del producto hardware	Amortización	Precio
Macbook Pro 13''	200€	1.500€
Monitor Benq 27''	27€	199€
	227€	1.699€
Total		1.926€

Tabla 55 Recursos Hardware Fuente: Propia

La tasa de amortización anual para los equipos electrónicos es del 20%

$$\text{Amortización} = (\text{Precio} * \text{tasa amortización}) * \left(\frac{\text{meses de uso}}{12}\right)$$

Equation 1 Amortización

9.1.3 Recursos Humanos

Estos costes se han calculado en base a un escenario donde el proyecto se hubiese desarrollado por un equipo de trabajadores profesionales. Un equipo formado por personas de diferentes roles que estuviesen encargados de diferentes tareas en el proyecto.

En la tabla se muestran los costes de cada uno de los empleados teniendo en cuenta la seguridad social que implica un 35%. El proyecto tienen de duración 8 meses.

Rol	Coste Mensual
Jefe de proyecto	1.200€
Analista	1.400€
Programador	2.000€
Diseñador	1.100€
Pruebas	1.000€
Total mensual	6.700€
Total proyecto	10.720€

Tabla 56 Recursos Humanos Fuente: Propia

El total del proyecto refleja un 20% del total anual ya que el equipo estaría ocupado con este proyecto una parte proporcional del tiempo a esta cantidad. El resto del salario estaría destinado a otros proyectos del equipo.

9.1.4 Recursos indirectos

Estos costes representan otro tipo de gastos en los que se ha incurrido durante el desarrollo del proyecto.

Descripción	Coste Mensual	Coste Total
Electricidad	50€	400€
Internet	15€	15€
Oficinas (para reuniones)	500€	4.000€
Total	595€	4.415€

Tabla 57 Recursos indirectos Fuente: Propia

9.1.5 Resumen total de costes

La siguiente tabla muestra el resumen total de costes del proyecto.

Descripción	Coste
Recursos Software	0€
Recursos Hardware	1.926€
Recursos Humanos	10.720€
Recursos Indirectos	4.415€
Total	17.061€

Tabla 58 Resumen Costes Totales Fuente: Propia

9.2 Impacto socio-económico

El mayor impacto del proyecto de desarrollo de la aplicación web recae en la estudiante que ha realizado el proyecto de fin de grado.

A nivel de implementación, este proyecto ha permitido descubrir y aprender el uso de un nuevo framework que se usa a nivel profesional a día de hoy en multitud de empresas.

El uso de laravel, después de su curva de aprendizaje, ha permitido poder entender en profundidad la arquitectura del Model-View-Controller ya que al proporcionar estructuras básicas de un proyecto (estructurado de carpetas por capas) permite poder centrar tu atención totalmente en desarrollar la lógica detrás del modelo para posteriormente validarlo[57].

A nivel de gestión, este proyecto permite tener una visión general de todas las partes que se ha de tener en cuenta para poder llevar a cabo un proyecto de desarrollo software en su totalidad. Esto implica desde el estudio previo del entorno, las posibles tecnologías a usar, la estimación de costes o los puntos legales a tener en cuenta.

En cuanto a la aplicación en si, una vez se haya publicado y sea accesible a través de internet por cualquier usuario, permitirá una nueva forma de búsqueda inequívoca de opciones sin gluten. Esta aplicación permite de forma sencilla, el poder encontrar restaurante con las opciones sin gluten que el usuario verdaderamente busca. Esto permite:

- Despreocupar al usuario de realizar ctrl-f en todas la páginas en busca del término “sin gluten”
- Evita tener que leer todos los comentarios de reseñas en busca del término “sin gluten”
- Ahorrar el tiempo invertido de buscar restaurante a restaurante en busca de las opciones sin gluten.

Esta nueva forma de buscar opciones sin gluten ahorra una cantidad de tiempo considerable a la hora de decidir querer planificar ir a un restaurante. Por lo que la aplicación se estima que tendrá un gran impacto en la población de personas celiacas o intolerantes al gluten además de permitir a los restaurantes poder publicitar las opciones sin gluten disponibles de forma clara.

10. CONCLUSIONES

Esta sección se dividirá en 4 apartados diferentes constando de: retrospectiva del proyecto problemas encontrados, conclusiones, conclusiones personales y futuros trabajos a realizar.

10.1 Retrospectiva del proyecto

Este apartado pretende volver al inicio del proyecto en el momento que se decidieron los objetivos y explicar como ha sido posible llegar a alcanzarlos.

El primer objetivo del trabajo fue crear una aplicación web que permitiese a los clientes realizar búsquedas de restaurantes sin gluten. Para desarrollar esto, primero se llevo a cabo un estudio de las distintas ofertas en el mercado para tener en cuentas sus características y poder encontrar un servicio que las empresas no estaban ofreciendo.

Tras el estudio de entono, se llevo acabo el estudio de las posibles tecnologías disponibles para su desarrollo. Estas tecnologías se escogieron tras numerosas horas invertidas para entender su uso y terminaron siendo las más idóneas para poder ofrecer una buena experiencia de usuario.

El segundo objetivo hacía alusión a todas las funcionalidades que se debían desarrollar en la implementación. Estas funcionalidades han sido posibles de llevar mayoritariamente por el uso de PHP junto el framework de Laravel. Este, al permitir desarrollar la arquitectura de MVC de forma sencilla permitía avanzar de forma ágil en el proyecto.

Por último, uno de los objetivo fue desarrollar un diseño de la aplicación que permitiese su crecimiento por lo que el proyecto se implementó teniendo en cuenta la clara separación de las funcionalidades para facilitar el poder añadir mas en un futuro.

10.2 Problemas encontrados

Los principales problemas encontrados en el desarrollo del proyecto fueron los siguientes:

En primer lugar, la dificultad a la hora de tomar la decisión sobre que tecnologías se usarían para implementar el proyecto. Inicialmente el proyecto se quiso desarrollar con las tecnologías utilizadas en la asignatura de Tecnologías informáticas para la web. Hay que tener en cuenta que las tecnologías habían evolucionado mucho desde que las usé por lo que la configuración del entorno se convirtió en un problema para desarrollar el trabajo. La inversión de tiempo fue frustrante ya que el avance era mínimo para la cantidad de horas invertidas. Finalmente opté por aprender de cero otras tecnologías/herramientas para la implementación de la web. Más adelante en el apartado conclusiones personales se comentará los beneficios de esta última decisión.

En segundo lugar, la falta de una clara definición del alcance del proyecto. Al inicio del proyecto se quiso abarcar muchas funcionalidades para la web, lo que hacía complicado y atrasaba el inicio de la implementación de este mismo. Finalmente se decidió empezar con las funcionalidades básicas y a partir de ahí una vez verificado se irían añadiendo nuevas funcionalidades y se actualizaría el documento. Esto a nivel del proyecto no afecta de forma económica en ningún caso pero es importante tenerlo en cuenta como aprendizaje para futuros proyectos llevados a cabo en un ámbito profesional. Ya que si existe un punto de inflexión en el proyecto debiendo tomar otra dirección supondría grandes costes adicionales a los previstos.

10.3 Conclusiones personales

En referencia a los nuevos conocimientos adquiridos, tras encontrar numerosos problemas al intentar usar las tecnologías aplicadas en asignatura de TIW, finalmente opté por aprender de cero a usar otras tecnologías. En este caso fueron PHP y el framework de Laravel. Después de haberle invertido el tiempo a poder aprender su uso para la implementación de aplicaciones web, he adquirido una serie de nuevos conocimientos que previamente no tenía y que aprovecharé en el futuro. Por lo que en conclusión, el decidir cambiar las tecnologías a pesar de tener que invertirle más tiempo fue muy positivo en términos de aprendizaje.

Haciendo referencia a la aplicación de conocimientos previos, este proyecto permite poder aplicar conocimientos de distintas asignaturas cursadas a lo largo de la carrera como por ejemplo:

- En la parte del documento y la gestión del proyecto:
 - Desarrollo de proyectos de desarrollo software: para estructurar toda la parte de casos de uso y requisitos
 - Tecnologías ágiles de desarrollo software: para poder aplicar las metodologías de trabajo aprendidas.
 - Métodos y técnicas de trabajo corporativo: para la parte de estudio del entorno.
- En la parte de la implementación
 - Interfaces de usuario: para la parte del front-end.
 - Diseño de bases de datos: para el correcto diseño de la base de datos del proyecto y su lógica
 - Tecnologías informáticas para la web : para la aplicación de la arquitectura de MVC y la implementación de los CRUD.

En cuanto a la parte de metodologías de trabajo, El uso de la metodología Kanban y sus tableros fue realmente útil para poder llevar el control del progreso a lo largo de todo el proyecto tanto a nivel de tareas como a nivel global. La metodología se ha podido seguir estrictamente y ha terminado siendo la idónea teniendo en cuenta que el proyecto era realizado por una única persona.

Por otro lado, en cuanto al tiempo invertido, es importante destacar que este proyecto se ha llevado a cabo en paralelo con otro trabajo de fin de grado en dirección y administración de empresas además de trabajar a tiempo completo para una empresa externa. Esto ha hecho que el tiempo invertido se viera alargado en el tiempo para su desarrollarlo. A modo de conclusión, llevar a acabo varios proyectos en paralelo y de ámbitos completamente distintos ha hecho que mejorase las capacidades de priorización de tareas, establecimiento concreto de objetivos semanales/mensuales o gestión de tiempos.

Finalmente y como conclusión, después de todos los meses invertidos en el proyecto tanto para la parte del diseño como para la parte de la implementación se considera que el proyecto cumple con los objetivos propuestos. Tanto a nivel de la documentación desarrollada como a nivel de la implementación de la aplicación web que, teniendo en cuenta que una prueba de concepto, es plenamente funcional.

10.4 Trabajos Futuros

Por último, a modo de añadir nuevas funcionalidades o mejoras que por diversos motivos no se encontraban en el alcance del proyecto pero sería interesante implementar en un futuro:

- Crear un tipo de usuario restaurante y otro cliente: diferenciando en si los dos tipos de cuentas que en la base de datos se traduciría en dos tablas diferentes “RestaurantUsers” y “Users”
- Añadir funcionalidades a la cuenta RestaurantUsers (usuarios que tienen restaurantes):
 - Poder añadir descuentos en los restaurantes : los restaurantes permitirían visualizar un código promocional que se canjearía al pagar la cuenta.
- Añadir funcionalidades a la cuenta de User (usuario que no tiene restaurantes):
 - Comentarios sobre los restaurantes.
 - Valoraciones de una a cinco estrellas de platos.
 - Valoración de una a cinco estrellas el servicio de los restaurantes.
 - Posibilidad de marcar como favorito platos o restaurantes y que se guarden en la cuenta del usuario.
- Añadir funcionalidad de suscripción para los usuarios: Al suscribirse les llegaría un newsletter semanal/mensual con los restaurantes más populares. Para esto, la web podría beneficiarse cobrando a los restaurantes una tasa por aparecer en esos newsletters.
- Reservas: Poder realizar reservas en los restaurantes a través de la propia web.

11. BIBLIOGRAFIA

- [1] «Framework: qué es, para qué sirve y algunos ejemplos», *Edix España*, 19 de agosto de 2021. <https://www.edix.com/es/instituto/framework/> (accedido 8 de septiembre de 2022).
- [2] S. Sulyman, «Client-Server Model», *IOSR J. Comput. Eng.*, vol. 16, pp. 57-71, ene. 2014, doi: 10.9790/0661-16195771.
- [3] «saiful_azmi.pdf». Accedido: 3 de mayo de 2022. [En línea]. Disponible en: http://norlizakatuk.weebly.com/uploads/2/6/6/0/26606863/saiful_azmi.pdf
- [4] S. Formosa, «Coming of Age: Investigating The Conception of a Census Web-Mapping Service for the Maltese Islands», 2000. doi: 10.13140/RG.2.1.2788.5601.
- [5] «11 Best Front-End Technologies To Use In 2022». <https://trio.dev/blog/front-end-technologies> (accedido 3 de mayo de 2022).
- [6] «HTML: Lenguaje de etiquetas de hipertexto | MDN». <https://developer.mozilla.org/es/docs/Web/HTML> (accedido 3 de mayo de 2022).
- [7] «JavaScript | MDN». <https://developer.mozilla.org/es/docs/Web/JavaScript> (accedido 3 de mayo de 2022).
- [8] «Los lenguajes de programación más usados en la actualidad». <https://www.universia.net/es/actualidad/empleo/lenguajes-programacion-mas-usados-actualidad-1136443.html> (accedido 2 de agosto de 2022).
- [9] «Historia del lenguaje Java | Manual Web». <https://www.manualweb.net/java/historia-java/> (accedido 2 de agosto de 2022).
- [10] «Aplicaciones Java SE y Java EE - Documentación de IBM». <https://www.ibm.com/docs/es/odm/8.5.1?topic=application-java-se-java-ee-applications> (accedido 9 de agosto de 2022).
- [11] «Qué es Spring framework», *OpenWebinars.net*, 3 de diciembre de 2018. <https://openwebinars.net/blog/que-es-spring-framework/> (accedido 8 de agosto de 2022).
- [12] «Qué es PHP: Características y usos | OpenWebinars». <https://openwebinars.net/blog/que-es-php/> (accedido 25 de agosto de 2022).
- [13] «2.4 Orígenes y Antecedentes de las Bases de Datos | Dataprix TI». <https://www.dataprix.com/es/mineria-datos-aplicada-encuesta-permanente-hogares/24-origenes-y-antecedentes-bases-datos> (accedido 8 de septiembre de 2022).
- [14] «Bases de datos relacional ¿Qué es y sus características?», *Ayuda Ley Protección Datos*. <https://ayudaleyprotecciondatos.es/bases-de-datos/relacional/> (accedido 9 de agosto de 2022).
- [15] «MySQL :: Why MySQL?» <https://www.mysql.com/why-mysql/> (accedido 10 de agosto de 2022).
- [16] «Qué es una base de datos | Oracle España». <https://www.oracle.com/es/database/what-is-database/> (accedido 8 de septiembre de 2022).
- [17] «Bases de datos no relacionales | Bases de datos de gráficos | AWS», *Amazon Web Services, Inc.* <https://aws.amazon.com/es/nosql/> (accedido 11 de agosto de 2022).
- [18] «Características de MongoDB», *OpenWebinars.net*, 7 de agosto de 2020. <https://openwebinars.net/blog/caracteristicas-de-mongodb/> (accedido 13 de agosto de 2022).
- [19] «AWS | Servicio de base de datos gestionada NoSQL (DynamoDB)», *Amazon Web Services, Inc.* <https://aws.amazon.com/es/dynamodb/> (accedido 13 de agosto de 2022).

- [20] «Comparación entre la arquitectura monolítica y la arquitectura de microservicios». <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith> (accedido 16 de agosto de 2022).
- [21] «Arquitectura de Microservicios». <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/microservicios> (accedido 16 de agosto de 2022).
- [22] apiumhub, «5 principales patrones de Arquitectura de Software». <https://apiumhub.com/es/tech-blog-barcelona/principales-patrones-arquitectura-software/> (accedido 26 de agosto de 2022).
- [23] «¿Tienes una metodología de trabajo? (Ejemplo paso a paso)», *Estrategia Práctica*, 2 de julio de 2017. <https://www.estrategiapractica.com/tienes-una-metodologia-trabajo/> (accedido 14 de agosto de 2022).
- [24] F. J. Mory, «Metodologías Ágiles vs Tradicionales», *Revista Empresarial & Laboral*, 7 de diciembre de 2021. <https://revistaempresarial.com/tecnologia/metodologias-agiles-vs-tradicionales/> (accedido 14 de agosto de 2022).
- [25] «¿Qué es el scrum y para qué sirve? – EBF – El blog de los mejores máster en A Coruña». <https://ebf.com.es/blog/los-origenes-de-scrum-de-donde-proviene-el-termino/> (accedido 14 de agosto de 2022).
- [26] «Las 5 etapas en los “Sprints” de un desarrollo Scrum», *OBS Business School*. <https://www.obsbusiness.school/blog/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum> (accedido 7 de septiembre de 2022).
- [27] Asana, «¿Qué es la metodología Kanban y cómo funciona? • Asana», *Asana*. <https://asana.com/es/resources/what-is-kanban> (accedido 28 de febrero de 2022).
- [28] «¿Qué es Kanban? Principales características y funciones», *Kanban Software for Agile Project Management*. <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban> (accedido 7 de septiembre de 2022).
- [29] «La facturación hostelera subió el 31 % en 2021 tras el bache de la Covid». https://www.foodretail.es/horeca/facturacion-hostelera-subio-bache-Covid_0_1632136789.html (accedido 5 de abril de 2022).
- [30] elEconomista.es, «La pandemia deja un comensal más digital y previsor, según TheFork». <https://www.eleconomista.es/retail/noticias/11421763/10/21/La-pandemia-deja-un-comensal-mas-digital-y-previsor-segun-TheFork.html> (accedido 5 de abril de 2022).
- [31] «¿Por qué no existe un censo sobre la enfermedad celíaca en España? Dificultades para el registro -Dr. Schär Institute». <https://www.drschaer.com/es/institute/n/censo-enfermedad-celiaca-espana> (accedido 6 de abril de 2022).
- [32] «Tripadvisor ultima la compra de “El Tenedor” | Tecnología | EL MUNDO». <https://www.elmundo.es/tecnologia/2014/05/07/536a0c5f268e3e8d148b4574.html> (accedido 6 de abril de 2022).
- [33] «Celicidad. Comer sin gluten - Aplicaciones en Google Play». <https://play.google.com/store/apps/details?id=ww.corp.celicidad&hl=es&gl=ES> (accedido 5 de abril de 2022).
- [34] «Quien somos», *Celicidad*. <https://celicidad.net/quien-somos/> (accedido 21 de abril de 2022).
- [35] «CeliCity, una aplicación que muestra restaurantes y comercios de productos sin gluten». <https://www.vitonica.com/equipamiento/celicity-una-aplicacion-que-muestra-restaurantes-y-comercios-de-productos-sin-gluten> (accedido 6 de abril de 2022).
- [36] E. Delgado López-Cózar, N. Robinson García, y D. Torres Salinas, «Manipular

- Google Scholar Citations y Google Scholar Metrics: simple, sencillo y tentador», *Manipulating Google Scholar Citations and Google Scholar Metrics: simple, easy and tempting*, may 2012, Accedido: 28 de febrero de 2022. [En línea]. Disponible en: <https://digibug.ugr.es/handle/10481/20469>
- [37] G. D. S. Hadad, «Complejidad en la Ingeniería de Requisitos», ago. 2012, Accedido: 15 de agosto de 2022. [En línea]. Disponible en: <http://repositorio.ub.edu.ar/handle/123456789/802>
- [38] 66734c026a9c090eca39c3b398fd8169, «¿Qué es un Análisis de Requerimientos?», *Espacios Business Media*, 19 de septiembre de 2016. <https://www.espacios.media/que-es-un-analisis-de-requerimientos/> (accedido 15 de agosto de 2022).
- [39] «Introducción al análisis de requisitos (A.R.) - Análisis y diseño detallado de aplicaciones informáticas de gestión». <https://sites.google.com/site/adai6jfm/home/introduccion-al-analisis-de-requisitos-ar> (accedido 15 de agosto de 2022).
- [40] «ieee830.pdf». Accedido: 15 de agosto de 2022. [En línea]. Disponible en: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- [41] staffcreativa, «12 Ventajas de la Programación PHP que Debes Saber», *Staff Creativa*, 24 de julio de 2014. <https://www.staffcreativa.pe/blog/ventajas-programacion-php/> (accedido 7 de septiembre de 2022).
- [42] «¿Qué es Laravel y para qué es útil? | Rootstack». <https://rootstack.com/es/learning/que-es-laravel-y-para-que-es-util> (accedido 7 de septiembre de 2022).
- [43] «Desarrollo de Software Web con Laravel: Beneficios e Inconvenientes», *Novadevs*. <https://novadevs.com/publicaciones/por-que-usar-laravel-en-el-desarrollo-de-software/> (accedido 8 de septiembre de 2022).
- [44] «Qué es Visual Studio Code y qué ventajas ofrece», *OpenWebinars.net*, 22 de julio de 2022. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/> (accedido 7 de septiembre de 2022).
- [45] L. Castellano Lendínez, «Kanban. Metodología para aumentar la eficiencia de los procesos», *3C Tecnol. Innov. Apl. Pyme*, vol. 29, n.º 1, pp. 30-41, mar. 2019, doi: 10.17993/3ctecno/2019.v8n1e29/30-41.
- [46] «Qué es MVC». <https://desarrolloweb.com/articulos/que-es-mvc.html> (accedido 8 de septiembre de 2022).
- [47] «BOE.es - BOE-A-2021-5032 Real Decreto 203/2021, de 30 de marzo, por el que se aprueba el Reglamento de actuación y funcionamiento del sector público por medios electrónicos.» <https://www.boe.es/buscar/doc.php?id=BOE-A-2021-5032> (accedido 4 de septiembre de 2022).
- [48] «¿Qué aspectos legales debe cumplir una página web? –»». <https://www.grupovadillo.com/aspectos-legales-pagina-web/> (accedido 6 de septiembre de 2022).
- [49] «BOE.es - BOE-A-2002-13758 Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico.» <https://www.boe.es/buscar/act.php?id=BOE-A-2002-13758> (accedido 6 de septiembre de 2022).
- [50] «BOE.es - BOE-A-2007-20555 Real Decreto Legislativo 1/2007, de 16 de noviembre, por el que se aprueba el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios y otras leyes complementarias.» <https://www.boe.es/buscar/act.php?id=BOE-A-2007-20555> (accedido 8 de septiembre de 2022).

- [51] «REGLAMENTO (UE) 2016/ 679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO - de 27 de abril de 2016 - relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/ 46/ CE (Reglamento general de protección de datos)», p. 88.
- [52] Jefatura del Estado, *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*, vol. BOE-A-2018-16673. 2018, pp. 119788-119857. Accedido: 6 de septiembre de 2022. [En línea]. Disponible en: <https://www.boe.es/eli/es/lo/2018/12/05/3>
- [53] «BOE.es - BOE-A-1998-8789 Ley 7/1998, de 13 de abril, sobre condiciones generales de la contratación.» <https://www.boe.es/buscar/act.php?id=BOE-A-1998-8789> (accedido 6 de septiembre de 2022).
- [54] «BOE.es - BOE-A-1999-23750 Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.» <https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750> (accedido 6 de septiembre de 2022).
- [55] «Visual Studio Code - Code Editing. Redefined». <https://code.visualstudio.com/> (accedido 8 de septiembre de 2022).
- [56] «Installation - Laravel - The PHP Framework For Web Artisans». <https://laravel.com/docs/4.2> (accedido 8 de septiembre de 2022).
- [57] «Ventajas de Laravel para Desarrollo de Aplicaciones Web», *J - Desarrollo Aplicaciones Web*, 23 de mayo de 2022. <https://juannunezblasco.es/ventajas-laravel-desarrollo-aplicaciones-web/> (accedido 6 de septiembre de 2022).

