

EASYDISHES

Recetas de cocina y organización

Creado por Antonio Riesco Serrano

Desarrollo de Aplicaciones Web,
2º, 2024, Antonio Riesco Serrano



ENLACES DE INTERÉS

- [Repository de GitHub](#)
- Enlace con la aplicación desplegada: easydishes.es
-

ÍNDICE

A.	DESCRIPCIÓN DEL PROYECTO.....	2
	1.2- Objetivos y alcance	2
B.	ANÁLISIS DEL SECTOR/MERCADO	3
	2.3- Normativa y Documentación técnica específica	5
C.	ANÁLISIS DEL SECTOR/MERCADO	5
	3.1- Diagrama/cronograma de flujo de procesos (Diagrama de Gantt)	5
	3.2- Proceso de Desarrollo Software	6
	3.2.1.1 - Tipos de usuarios	6
	3.2.1.3 - Casos de uso.....	6
	3.2.1.4 - Guía de estilo.....	9
	3.2.1.5 – Prototipo del sitio web.....	9
	3.2.1.6 – Mapa de navegación.....	18
	3.2.2. Fase de desarrollo	18
	3.2.2.1 – Base de datos.....	18
	3.2.2.1.1 Análisis de requisitos de datos de la aplicación	18
	3.2.2.1.2 Diseño lógico de datos.....	19
	3.2.2.1.3 Paso del modelo lógico (E/R) al modelo relacional (tablas)	20
	3.2.2.1.4 Aplicación de reglas de normalización al modelo relacional	21
	3.2.2.1.5 Tipos de datos para el sistema gestor seleccionado.....	21
	3.2.2.1.6 Scripts de creación de tablas e inserciones iniciales.....	22
	3.2.2.2.- Servidor	22
	3.2.2.2.1 Lista de funciones en php.....	22
	3.2.2.3.- Cliente.....	23
	3.2.2.3.1.- Diseño de la interfaz.....	23
	3.2.2.3.2.- Accesibilidad.....	23
	3.2.2.3.3.- Usabilidad.....	24
	3.2.2.3.4.- Desarrollo web en entorno cliente	26
	3.2.3.- Fase de despliegue	27
	3.2.3.1.- Despliegue utilizando un hosting	27
	3.3- Seguimiento y control de incidencias	28
	3.4.- Indicadores de calidad de procesos	28



D.RECURSOS MATERIALES.....	28
4.1.- Inventario, valorado, de medios.....	28
4.2.- Presupuesto económico.....	29
E.RECURSOS HUMANOS.....	29
5.1.- Organización.....	29
5.2.- Contratación	30
5.3.- Prevención de riesgos laborales	32
F. VIABILIDAD TÉCNICA.....	32
6.1.- Estudio de viabilidad técnica.....	32
G. VIABILIDAD ECONÓMICO-FINANCIERA.....	33
7.1.- Inversiones y gastos.....	33
7.2.- Financiación.....	34
7.3.- Viabilidad económico-financiera	34
H. CONCLUSIÓN.....	37
BIBLIOGRAFÍA / WEBGRAFÍA.....	37

A. DESCRIPCIÓN DEL PROYECTO

1.1- Características generales

EasyDishes es una aplicación web que permite a los usuarios crear, compartir y utilizar recetas de cocina. Funciona de manera similar a un blog, con entradas correspondientes con cada receta. También se pueden filtrar las recetas por autor, fecha o etapa, que puede ser: desayuno, comida o cena. Dentro de estas, se pueden compartir en redes sociales y añadir a favoritos. Otra función interesante de la aplicación es la posibilidad de organización del usuario. En base a las recetas, se pueden añadir ingredientes con sus cantidades y unidades a una lista de la compra e imprimir la misma para poder llevarla al supermercado. Por último, se pueden enviar mensajes para contactar con la página.

1.2- Objetivos y alcance

La idea de negocio se trata de una página de recetas que tenga un doble objetivo: permitir a las personas organizarse y compartir recetas de diferentes comidas con ingredientes variados. La necesidad a satisfacer es proveer recetas de cocina tanto para apasionados como para principiantes y permitir simplificar la cesta de la compra pensando en qué ingredientes van a ser necesarios.

Se intentará crear una interfaz con todas las características visuales necesarias sin ser demasiado recargada usando colores y tipografías sencillos de entender y acordes con la temática del proyecto.

Un objetivo de la página debe ser el manejar diferentes recetas, es decir, guardar información de utilidad sobre platos para un usuario que tenga interés en la cocina. Aprender también deber ser posible, habiendo diferentes grados de dificultad en las recetas para ir progresando o consultando recetas de diferentes cocineros o culturas.

EasyDishes

Easydishes



CENTRO INTEGRADO DE
FORMACIÓN PROFESIONAL
Nº 1 - CUENCA

Otro objetivo es permitir la organización de los usuarios en su ámbito personal, lo que incluye menús y lista de la compra que puedan utilizar en su día a día. Por otro lado, crear una comunidad culinaria de gente a la que le interese en menor o mayor medida la cocina. Para esto se debe conseguir que haya una comunidad de usuarios.

Finalmente, puede servir de inspiración para que cada persona cree e innove en sus propias recetas caseras.

El alcance es permitir a miles de usuarios únicos utilizar las páginas y guardar información relativa a los mismos como favoritos o listas de la compra. Se usará una parte en el servidor y otra parte en el cliente. La parte cliente no permitirá acceder a datos sensibles almacenados en la base de datos como usuarios o contraseñas. Para obtener, actualizar o editar dichos datos, se usará una API propia por la cual en cliente en Angular realiza peticiones http al servidor Apache instalado en la misma máquina que devolverá archivos JSON confeccionados en PHP. La aplicación se puede usar sin iniciar sesión con algunas funcionalidades limitadas y otras no, como crear una lista de la compra temporal o visualizar recetas.

B. ANÁLISIS DEL SECTOR/MERCADO

2.1- Prospectiva del Título en el sector

En la actualidad, la programación informática se encuentra muy presente en las vidas de las personas de los países que hacen uso de ella en los diferentes dispositivos tecnológicos.

Por lo tanto, se puede decir que en el mundo actual se hace necesario suplir dichas necesidades tecnológicas continuamente. Ahí es donde entra el trabajo de los programadores informáticos o desarrolladores web. Para suplir esa necesidad se crean empresas de software que ayudan a las personas a desarrollarse en su día a día y que necesita profesionales cualificados que realicen la tarea de la programación. Esta parece ser también la tendencia del mercado, ya que las empresas llevan a cabo todo tipo de programas con muchas y muy variadas aplicaciones.

La programación informática tiene varias ramas como Inteligencia artificial, desarrollo web y multiplataforma, ciencia de datos, desarrollo de videojuegos, etc. En este caso nos centraremos en el desarrollo web. Las tecnologías web son amplias y diversas. Han ido evolucionando durante los años. Las herramientas están en constante cambio y algunas de ellas siguen manteniendo compatibilidad con versiones anteriores por la gran cantidad de código ya escrito que debe seguir funcionando.

Encontramos dos tipos de desarrollo web: cliente y servidor. Dentro del desarrollo cliente, se crean continuamente librerías, frameworks, plataformas para añadir funcionalidades a JavaScript y debido a lo fácil que es actualizarlas usando NPM y otros gestores de paquetes, el sector cambia continuamente. Algunos ejemplos de esto son React, Angular, Vue, Svelte, Next, SCSS, RXJS, Tailwind, Bootstrap, Remix, Nuxt, GraphQL, Astro, además de los clásicos HTML y CSS, que también se actualizan muchas veces basándose en el avance de los anteriores. Esto hace difícil saber cómo será el futuro del frontend, ya que si lo comparamos con hace unos años, hubiese sido imposible ver el avance. En mi opinión, pienso que se crearán nuevas herramientas para hacer más fácil el trabajo del programador y que no pierda el tiempo volviendo a solucionar problemas ya solucionados en el pasado, aunque está bien saber cómo funcionan por si fuese necesario cambiar algo.

En cuanto al desarrollo servidor, hemos comprobado que también se actualiza con rapidez. Hay que tener en cuenta que están apareciendo tecnologías serverless, que prescinden



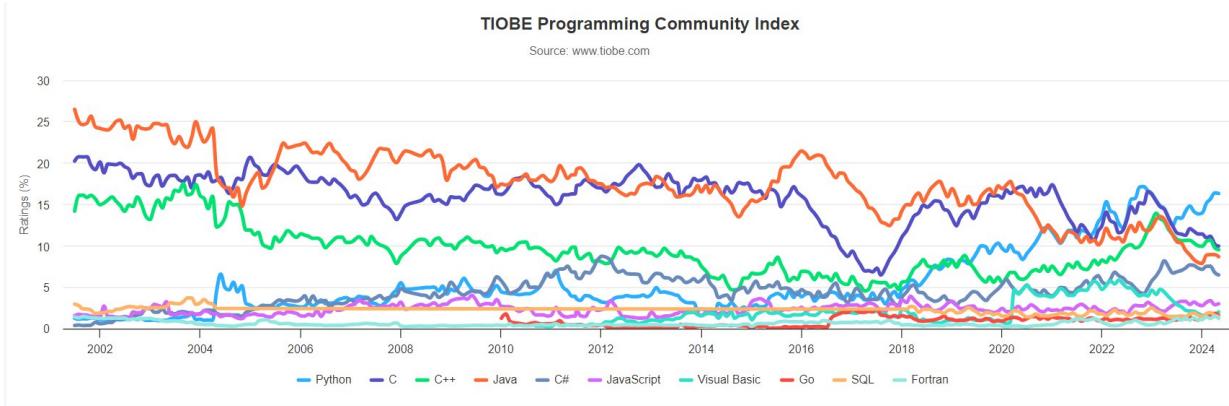
totalmente del servidor. Sin embargo, en la actualidad es más común usarlo con funciones críticas como de guardar información privada y no privada, autenticar usuarios, realizar pagos electrónicos, enviar e-mails, interactuar con la base de datos... Esta estructura clásica seguirá existiendo y proporcionando la posibilidad de crear páginas web funcionales, rápidas y escalables.

Por último, destacar la importancia de la Inteligencia Artificial con herramientas como chatGPT, Devin, Github Copilot, que ya están ayudando a los programadores ofreciendo código sugerido, patrones de diseño o soluciones únicas. El sector de la IA ha avanzado muy rápido en pocos años y se espera que esa tendencia continúe.

2.2- Evolución y tendencia del sector.

En cuanto a la evolución, se encuentran diferentes datos y estadísticas relacionadas con la programación.

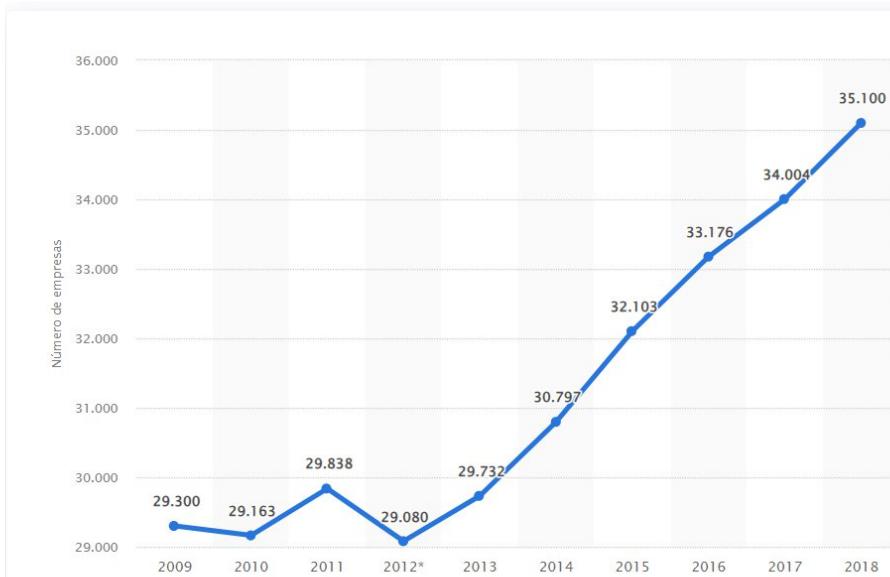
- Lenguajes en crecimiento según el índice tiobe:



Como se puede ver, JavaScript, lenguaje en el que se basa TypeScript (usado en la parte cliente del proyecto) se encuentra en una posición alta en el índice y con casi récord histórico. Esto significa que todo el ecosistema está en constante desarrollo tanto por la comunidad como por empresas que requieren del uso de un lenguaje de alto nivel sencillo y con potencial.

Por otro lado, aunque no aparezca en el gráfico superior, se puede observar el lenguaje PHP en decimosexto lugar. Sigue siendo uno de los lenguajes de programación más famosos, usados y con gran cantidad de funcionalidades útiles para el desarrollo web. Sin embargo, su popularidad ha bajado en los últimos años y se podría esperar que siga bajando

- Evolución del número total de las empresas del sector de las tecnologías de la información, la comunicación y los contenidos (TICC) en España de 2009 a 2018.



El crecimiento es bastante notable durante todos los años posteriores a 2012, lo que indicaría que en los años posteriores vaya a haber un avance todavía mayor.

- Según el INE, El número de empresas del sector TIC que realizaron I+D creció un 7,7% y se situó en 2.073 en el año 2019
- Según [Ontsi](#), la cifra de negocio del Sector TIC en España en el año 2017 crece un 1,2% respecto al año anterior. Con este incremento, mantiene una tendencia positiva en este indicador desde 2012, cuando se situaba en 77.304 millones de euros. En 2017, la cifra alcanzada es de 87.627 millones de euros, en torno a los 10.000 millones por encima.
- Según [Channel Partner](#), se prevee un aumento del 22% en los trabajos de desarrollador web en España hasta 2030, por lo que la programación informática sería de gran ayuda a la economía del país.

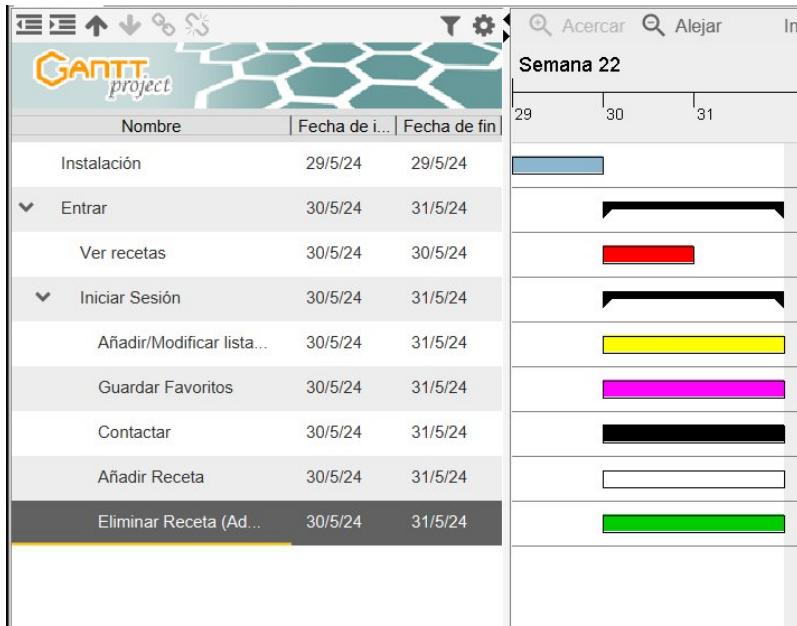
2.3- Normativa y Documentación técnica específica

- [Estándares w3 para la web para accesibilidad](#). Esta documentación recoge patrones para el diseño y el desarrollo de páginas web para HTML, CSS o JavaScript, entre otras. Estos patrones permiten a usuarios con discapacidades o diferentes condiciones consumir el contenido de las páginas web. Las técnicas de accesibilidad suponen añadir texto alternativo a las imágenes, usar tab index, añadir subtítulos a los vídeos o audios, etc.
- Uso de “Clean Code”. Esta técnica hace que la lectura y escritura de código sea más fácil, permitiendo el trabajo en equipo y la sinergia. Para ello, se recomienda dar nombres descriptivos a las variables, constantes, clases, funciones..., refactorizar el código cuando se repita en varios sitios a la vez, evitar el código “spaghetti”, usar una estructura, tabular las sentencias, etc.



C. ANÁLISIS DEL SECTOR/MERCADO

3.1- Diagrama/cronograma de flujo de procesos (Diagrama de Gantt)



Lo que el cronograma representa es como primer paso una instalación de la aplicación web usando el manual. Después, se podrá entrar a la página web. Cuando se entre se podrán realizar cosas básicas como ver recetas. Una vez se haya iniciado sesión, se podrán añadir/modificar listas, guardar favoritos, contactar, añadir receta y eliminarla (sólo si es administrador).

3.2- Proceso de Desarrollo Software

3.2.1.- Fase de análisis

3.2.1.1 - Tipos de usuarios

En el proyecto se incluyen 2 tipos de usuarios:

- Visitante: persona que no necesita registrarse para visualizar recetas, información de la página o página de registro/login. Si intenta acceder a una página que no le corresponde será redirigido o instado a registrarse o iniciar sesión.
- Usuario registrado: persona que ha llenado correctamente el registro y puede acceder a todas las funciones de la página web exceptuando la capacidad de eliminar recetas o realizar operaciones con la base de datos.
- Administrador: cuenta de usuario especial con ciertas capacidades como eliminar recetas (se puede usar para borrar contenido inadecuado) o acceder a la base de datos.

3.2.1.2.- Descripción de requisitos

- Usar una barra de navegación para entrar en las diferentes secciones de la página.
- La pantalla de inicio contiene 6 recetas del día, que serán generadas aleatoriamente cada día. También contiene información de la página.



- Permitir visualizar recetas con su modo de preparación e ingredientes. Para eso se debe hacer click en su correspondiente tarjeta, que incluye el título de la receta y una pequeña imagen de pre-visualización. También se debe poder compartir la receta en redes sociales.
- Guardar recetas en favoritos para verlas después. Se pueden ver desde la pestaña "favoritos". Es necesario iniciar sesión.
- Añadir los ingredientes, cantidad y unidades de una receta en una lista de la compra. Es necesario iniciar sesión.
- Visualizar tarjetas de recetas según la fecha, el autor y la comida en la página de recetas.
- Registrarse e iniciar sesión en la página tomando como usuario el DNI de la persona.
- Eliminar todos los ingredientes de la lista de la compra.
- Añadir ingredientes, cantidad y unidad nuevos en la lista de la compra que no estén en ninguna receta.
- Eliminar ingredientes de la lista de la compra.
- Generar PDF con la lista de la compra.
- Contactar con el equipo de la empresa mediante un formulario en el que se indicará correo, nombre, apellidos, teléfono y el mensaje a enviar.
- Página de error 404 si la ID de la receta está mal o se intenta acceder a una página inexistente.
- Contactar con los perfiles oficiales de EasyDishes desde la página principal.

3.2.1.3 - Casos de uso

- Ver receta

Descripción	El usuario quiere visualizar una receta
Actor	Visitante, Usuario registrado
Pre-condición	Entrar en una URL con el ID de la receta
Escenarios	Flujo de eventos principal: ID es correcta Flujo de eventos secundario: ID no existe
Pos-condición	Flujo de eventos principal: se muestra la página de la receta Flujo de eventos secundario: redirige a una página de error 404

- Iniciar sesión

Descripción	El usuario quiere autenticarse
Actor	Visitante
Pre-condición	Entrar en una URL desde la barra de navegación
Escenarios	Flujo de eventos principal: los datos son correctos Flujo de eventos secundario: los datos son incorrectos
Pos-condición	Flujo de eventos principal: permite acceder a los beneficios del usuario registrado Flujo de eventos secundario: se vuelven a pedir los datos y se muestra un error

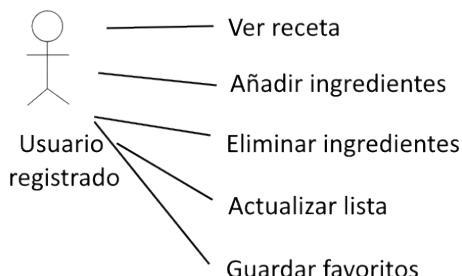
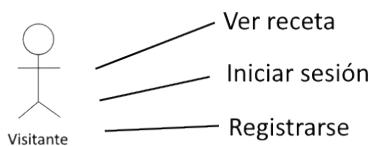
- Registrarse

Descripción	El usuario quiere registrarse
Actor	Visitante
Pre-condición	Entrar en una URL desde la barra de navegación
Escenarios	Flujo de eventos principal: los datos son correctos Flujo de eventos secundario: los datos son incorrectos

Pos-condición	Flujo de eventos principal: notifica de que la cuenta se ha creado correctamente Flujo de eventos secundario: se vuelven a pedir los datos y se muestra un error en los campos incorrectos
– Añadir ingredientes de la lista de la compra	
Descripción	El usuario o el sistema añade un ingrediente
Actor	Usuario registrado, sistema
Pre-condición	Pulsar el botón añadir de la receta o lista
Escenarios	Flujo de eventos principal: el ingrediente es nuevo Flujo de eventos secundario: el ingrediente ya existe
Pos-condición	Flujo de eventos principal: añade el nuevo ingrediente Flujo de eventos secundario: se suma la cantidad nueva
– Eliminar ingredientes de la lista de la compra	
Descripción	El usuario quiere eliminar un ingrediente
Actor	Usuario registrado
Pre-condición	Pulsar el botón de la papelera
Escenarios	Flujo de eventos principal: se busca el ingrediente en el almacenamiento
Pos-condición	Flujo de eventos principal: se elimina el ingrediente Flujo de eventos secundario: se vuelven a pedir los datos y se muestra un error
– Actualizar lista de la compra	
Descripción	El usuario quiere ver sus listas de la compra
Actor	Usuario registrado
Pre-condición	Entrar en una URL desde la barra de navegación
Escenarios	Flujo de eventos principal: la lista existe Flujo de eventos secundario: la lista no existe
Pos-condición	Flujo de eventos principal: carga los elementos de la lista Flujo de eventos secundario: se carga una lista vacía
– Guardar favoritos	
Descripción	El usuario quiere guardar una receta en favoritos
Actor	Usuario registrado
Pre-condición	Pulsar en el botón de añadir a favoritos (★)
Escenarios	Flujo de eventos principal: la receta no está en favoritos Flujo de eventos secundario: la receta está en favoritos
Pos-condición	Flujo de eventos principal: añade la receta Flujo de eventos secundario: no se realiza ninguna acción
– Contactar	
Descripción	El usuario quiere mandar un mensaje al creador de la página
Actor	Usuario registrado
Pre-condición	Pulsar en el botón de contacto
Escenarios	Flujo de eventos principal: el mensaje es válido (sin palabras malsonantes) Flujo de eventos secundario: el mensaje es inválido
Pos-condición	Flujo de eventos principal: guarda el mensaje en la base de datos Flujo de eventos secundario: muestra el error de mensaje inválido
– Añadir receta	
Descripción	El usuario quiere añadir una receta



Easydishes	
Actor	Usuario registrado
Pre-condición	Pulsar en el botón de añadir
Escenarios	Flujo de eventos principal: los datos son correctos Flujo de eventos secundario: los datos son incorrectos
Pos-condición	Flujo de eventos principal: añade la receta Flujo de eventos secundario: muestra los errores en el formulario



3.2.1.4 - Guía de estilo

La aplicación tiene 2 colores principales: blanco y gris. Se apuesta por un diseño minimalista sin usar gran cantidad de colores. El color gris es perteneciente a la clase “secondary” de Bootstrap. Se utilizan otros colores en diferentes partes de la aplicación, por ejemplo al aceptar (color verde) o rechazar/eliminar (color rojo) una acción. También cuando algún proceso es satisfactorio o en botones de compartir en redes sociales.

En la página se utilizan 2 fuentes principales: Roboto y “Itdiploma”. Ésta última imita la caligrafía manual y se usa principalmente para los títulos y el logotipo de la página. Por otro lado, Roboto es la fuente usada en todos los demás sitios y es la fuente principal.

Las tarjetas son siempre iguales, tienen un fondo gris y un botón para realizar una acción como ver la receta o filtrar algunas recetas.

En algunas páginas se utilizan modales, como al añadir ingredientes.



3.2.1.5 – Prototipo del sitio web

Página de inicio:

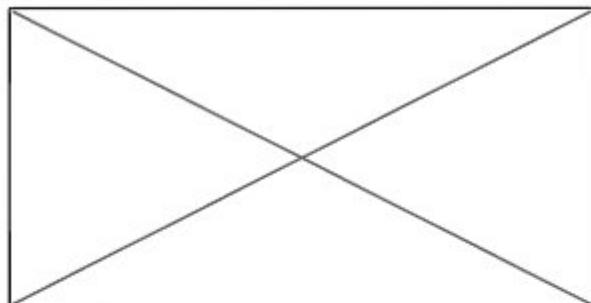
Página de inicio

The wireframe shows the layout of the homepage. At the top, there's a header bar with the title 'EASYDISHES' and navigation links for 'Recetas', 'Favoritos', and 'Lista de la compra'. Below the header is a section titled 'Recetas del día' containing six recipe cards, each with a placeholder image, a name ('Nombre de la receta'), a 'VER' button, and a star icon. Underneath this is a 'Información' section with a large text area containing placeholder text (Lorem ipsum). At the bottom, there's a 'Redes sociales' section with icons for Facebook, X (Twitter), and LinkedIn, and a 'CONTACTAR' button.

Receta:



EASYDISHES

[Recetas](#) [Favoritos](#) [Lista de la compra](#)

INGREDIENTES

- Ingrediente 1
- Ingrediente 2
- Ingrediente 3
- Ingrediente 4
- Ingrediente 5

[AÑADIR A LISTA](#)

PREPARACIÓN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse arcu risus, cursus eget nulla at, convallis laoreet est. Vestibulum risus tortor, eleifend at lorem id, fringilla facilisis massa. Vestibulum ornare, augue eu dapibus dictum, nisi massa mollis tellus, quis finibus dolor quam a diam. Curabitur fermentum sollicitudin felis non pellentesque. Phasellus eu ex vestibulum, tristique lectus a, luctus ipsum. Sed dictum lorem sed elementum eleifend. Duis et orci ac nibh pharetra tristique. Pellentesque tortor nunc, sodales sed pharetra eget, rhoncus vel nisl. Mauris viverra gravida ante id luctus. Fusce eget nulla ut ipsum blandit suscipit sed in nisl. Integer placerat viverra lacus, vitae mattis turpis euismod vel.

[Compartir](#)[★ Añadir a favoritos ★](#)

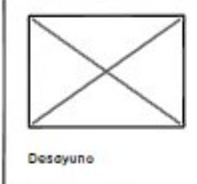
Recetas:

Página "Recetas"

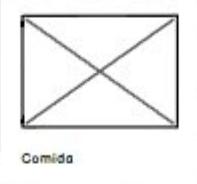
EASYDISHES

[Recetas](#) [Favoritos](#) [Lista de la compra](#)

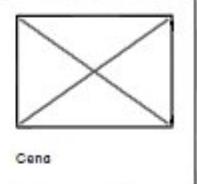
[Ver por etapa](#)



Desayuno



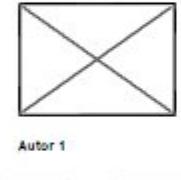
Comida



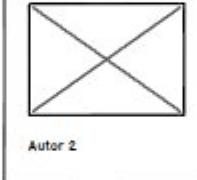
Cena

[VER](#)

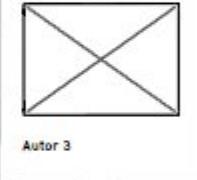
[Ver por autor](#)



Autor 1



Autor 2



Autor 3

[VER](#)

Página recopilatoria de recetas o favoritos



EASYDISHES

Recetas Favoritos Lista de la compra

FAVORITOS / RECETAS

Nombre de la receta

VER

AÑADIR RECETA

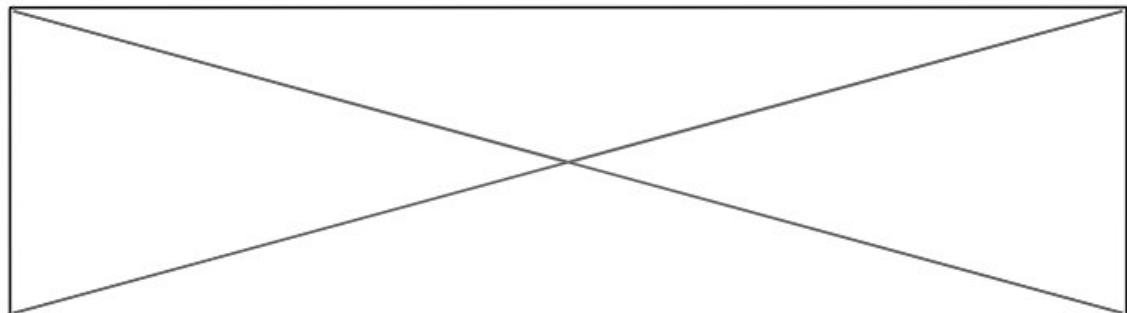
Lista de la compra:



EASYDISHES

[Recetas](#) [Favoritos](#) [Lista de la compra](#)

LISTA DE LA COMPRA



Leche --- 1 L
Huevos --- 1 Decena
Pan --- 100 g
Arroz --- 200 g
Patatas --- 2 kg



[ELIMINAR TODO](#)

[AÑADIR](#)

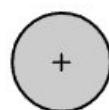
Añadir receta:



EASYDISHES

[Recetas](#) [Favoritos](#) [Lista de la compra](#)

AÑADIR RECETA



SUBIR IMAGEN

Ingredientes

Nombre

Cantidad

 3

Etapa

Modo de preparación

ACEPTAR

Contacto:



EASYDISHES

[Recetas](#) [Favoritos](#) [Lista de la compra](#)

CONTACTO

Mensaje

Receta

Inicio de sesión

Iniciar Sesión

DNI

Contraseña

Enviar

[Si no tienes cuenta, puedes crearla aquí](#)

Registro



REGISTRO

DNI

Nombre

Primer apellido

Segundo apellido

Teléfono

Correo Electrónico

Contraseña

Repetir contraseña

Enviar

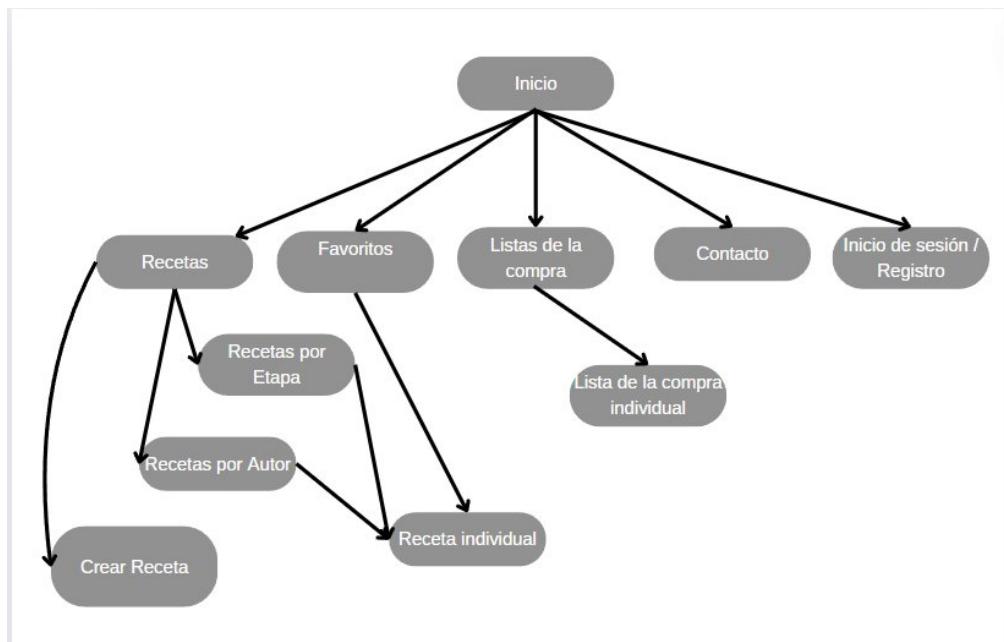
¿Ya tienes cuenta? Inicia sesión

Listas

Tús listas de la compra

Añadir
Lista 1: Fecha
Lista 2: Fecha
Lista 3: Fecha

3.2.1.6 – Mapa de navegación



3.2.2. Fase de desarrollo

3.2.2.1 – Base de datos

3.2.2.1.1 Análisis de requisitos de datos de la aplicación

- Identificación de entidades e interrelaciones entre entidades. Se encuentran varias entidades. Las entidades que clasificamos, que representan objetos imaginarios con diferentes propiedades, son “recetas”, “preparación”, “ingredientes” “listascompra” (listas de la compra), usuarios y mensajes. La receta se relaciona (R-P) con la preparación, ya que cada receta dispone de una descripción del cocinado de la receta, que se guarda en la tabla “preparación”. Otro componente de la página de receta es una lista de ingredientes. Por cada receta, habrá varios ingredientes y también cada ingrediente puede estar presente en varias recetas. Además, toda



receta debe contener como mínimo 1 ingrediente, aunque un ingrediente puede no pertenecer a ninguna receta. Esto genera la relación RI.

Por otro lado, una lista de la compra estará formada por varios ingredientes y puede haber distintas listas que tengan el mismo ingrediente. Se genera una relación de 1 a muchos por lo tanto llamada IL.

Otra relación es RM, que une las entidades recetas y mensajes. Todo mensaje va a tratar sobre una receta y una receta puede estar presente en ninguno o varios mensajes.

Está presente también la relación “Autoría”, que permite que un usuario pueda crear ninguna o muchas recetas y que haya un sólo autor por cada receta. La receta también se relaciona con el usuario en “Favoritos”. Esta relación permite que un usuario pueda guardar las IDs de diferentes recetas. Un usuario puede tener muchos favoritos y una receta puede ser favorita de varias personas.

Por último, se encuentra la relación UL. Esta relación de 1 a muchos se basa en que una la lista de la compra sólo puede pertenecer a 1 usuario pero 1 usuario puede no tener listas creadas, tener 1 o tener varias.

- B. Identificación de atributos de cada entidad. Se empieza viendo que la entidad recetas guarda bastantes datos importantes que luego serán visibles por los diferentes usuarios. El atributo Etapa podrá determinar si se trata de desayuno, comida o cena, la fecha de creación de la receta también se guarda, el nombre de la receta, un identificador ID y una imagen en BASE 64. La entidad preparación guarda el identificador y la descripción de los pasos a seguir de la receta.

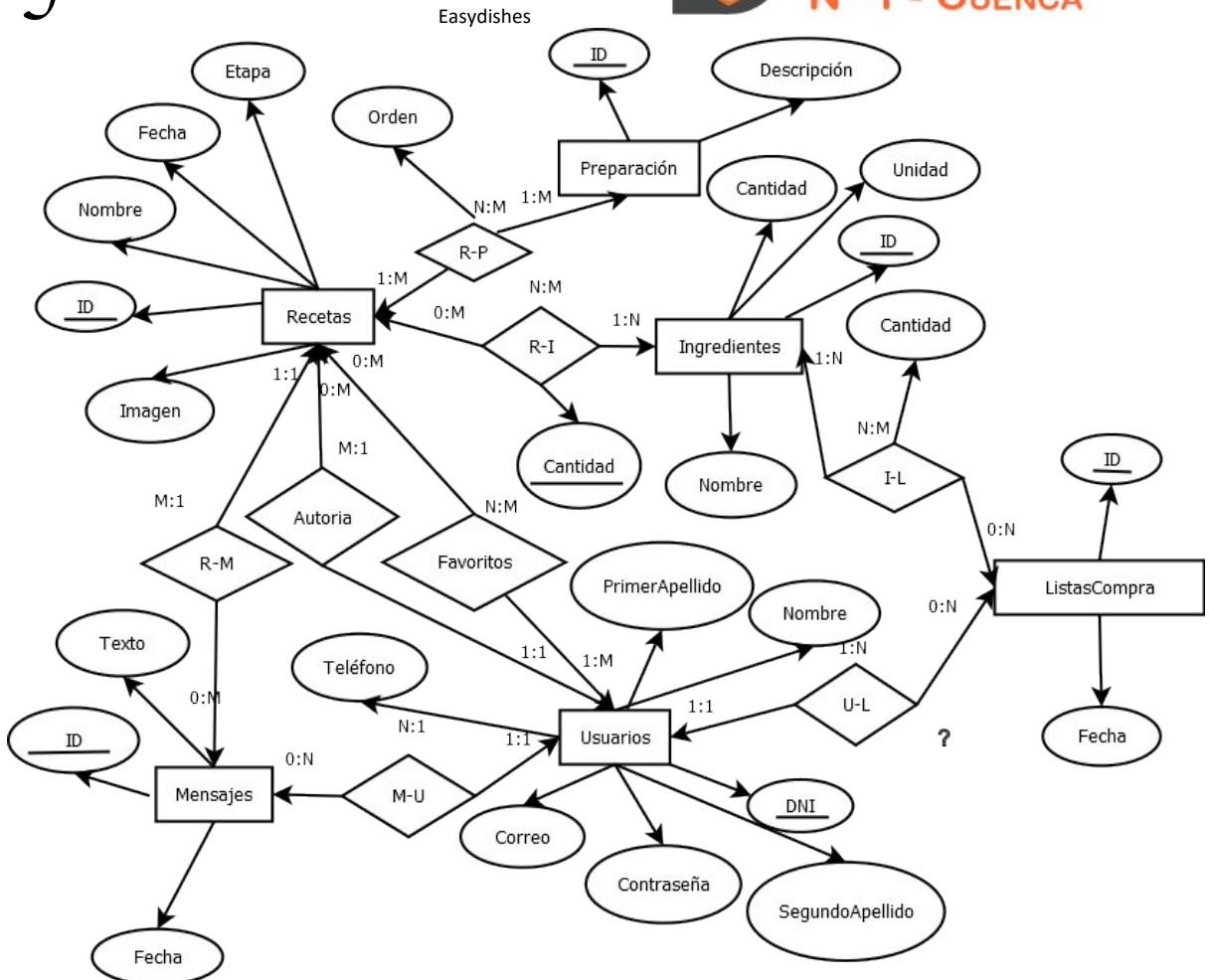
Por otro lado, los ingredientes son un conjunto de datos que incluyen nombre, cantidad, unidad y el identificador. La lista de la compra está compuesta del identificador y de la fecha, ambos datos se muestran en la aplicación final para distinguir las distintas listas.

La entidad usuarios guardará todos los datos personales del usuario. Esto incluye el dni, nombre, los 2 apellidos, el número de teléfono, correo electrónico y la contraseña encriptada.

Finalmente, esta presente la entidad mensaje, que además del identificador, guardará el texto del mensaje.

3.2.2.1.2 Diseño lógico de datos

El diagrama E-R es el siguiente:



3.2.2.1.3 Paso del modelo lógico (E/R) al modelo relacional (tablas)

En el diagrama E-R se puede ver que existen relaciones de muchos a muchos y de una a muchos. Las primeras se convertirán en una tabla nueva, que estará compuesta como clave primaria por las 2 ids de las dos entidades y como clave ajena estas a su respectiva entidad. La segunda se convertirá en una clave ajena en una tabla que apunte a la clave primaria de otra tabla.

- A. Atributos de las tablas (todas las cadenas son varchar(255) y el tamaño por defecto de los otros tipos de datos):

- Recetas: id: cadena, nombre: cadena, fecha: fecha, imagen: texto largo, autor: cadena.
- Preparación: id:cadena, descripcion: texto mediano
- Ingredientes: id:cadena, nombre:cadena, cantidad: entero, unidad: cadena
- Listascompra: id: cadena, fecha: fecha, usuario: cadena
- Mensajes: id:cadena, texto: cadena, fecha: fecha, receta: cadena, dniUsuario: cadena
- Usuarios: dni: cadena, nombre: cadena, primerApellido: cadena, segundoApellido: cadena, telefono: cadena, correo: cadena, contraseña: cadena
- R-P(receta: cadena, preparacion: cadena, orden: cadena)
- R-I(receta: cadena, ingrediente: cadena, cantidad: entero)

EasyDishes



CENTRO INTEGRADO DE
FORMACIÓN PROFESIONAL
Nº 1 - CUENCA

Easydishes

- I-L(ingrediente: cadena, lista: cadena, cantidad: entero)
- Favoritos(receta: cadena, usuario: cadena)

B.Las claves primarias son las siguientes:

- Recetas: id
- Preparacion: id
- Ingredientes: id
- Listascompra: id
- Usuarios: dni
- R-P: receta y preparacion
- R-I: receta y ingrediente
- I-L: ingrediente y lista
- Favoritos: receta y usuario

Las claves ajenas son las siguientes:

- Recetas: Autor-> Usuarios(DNI)
- Listascompra: usuario->Usuarios(DNI)
- Mensajes: dniUsuario->Usuarios(DNI)
- R-P: receta->Recetas(ID), preparacion->Preparacion(ID)
- R-I: receta->Recetas(ID), ingrediente->Ingredientes(ID)
- I-L: ingrediente->Ingredientes(ID), lista->listasCompra(ID)
- Favoritos: receta->Recetas(ID), usuario->Usuarios(dni)

C. Las reglas de integridad:

- Integridad de clave: las claves primarias elegidas nunca podrán ser nulas.
- Integridad de referencia: todas las claves ajenas apuntan a otra clave primaria existente en otras tablas que siempre habrá sido creada previamente.
- Muchos campos permiten el valor nulo, aunque casi nunca es usado en la aplicación final.
- No se permite crear una receta, lista o añadir favoritos sin crear antes un registro de un usuario

Las reglas de seguridad:

- Las contraseñas serán almacenadas encriptadas
- Las imágenes serán almacenadas en un campo de LONGTEXT, para que se puedan almacenar imágenes grandes.
- La forma de acceder a la base de datos es a través de una serie de scripts almacenados en el servidor de forma ajena al usuario de la página.
- Se utiliza la última versión compatible de MySQL

D. El modelo relacional se puede obtener con los datos proporcionados arriba.

3.2.2.1.4 Aplicación de reglas de normalización al modelo relacional

No se ha visto necesario normalizar el modelo relacional

3.2.2.1.5 Tipos de datos para el sistema gestor seleccionado

La mayoría de los tipos de datos de los campos de las tablas son varchar2 de una longitud de 255 caracteres. Además de esos, se encuentran 3 campos de tipo DATE llamados fecha en recetas, listas y mensajes. Por otro lado, las cantidades serán de



tipo INT, el campo imagen de recetas es de tipo LONGTEXT para almacenar archivos grandes y la preparación es MEDIUMTEXT para que no haya problemas con recetas largas.

3.2.2.1.6 Scripts de creación de tablas e inserciones iniciales

El archivo .php con la creación de las tablas se encuentra en servidor/installacion/installacion.php dentro del repositorio de github.

<https://github.com/luigitimeisreal/EasyDishes/blob/main/servidor/installacion/installacion.php>

3.2.2.2.- Servidor

3.2.2.2.1 Lista de funciones en php

En mi caso, he ordenado en diferentes archivos el código php al que se accede haciendo peticiones POST y GET al servidor.

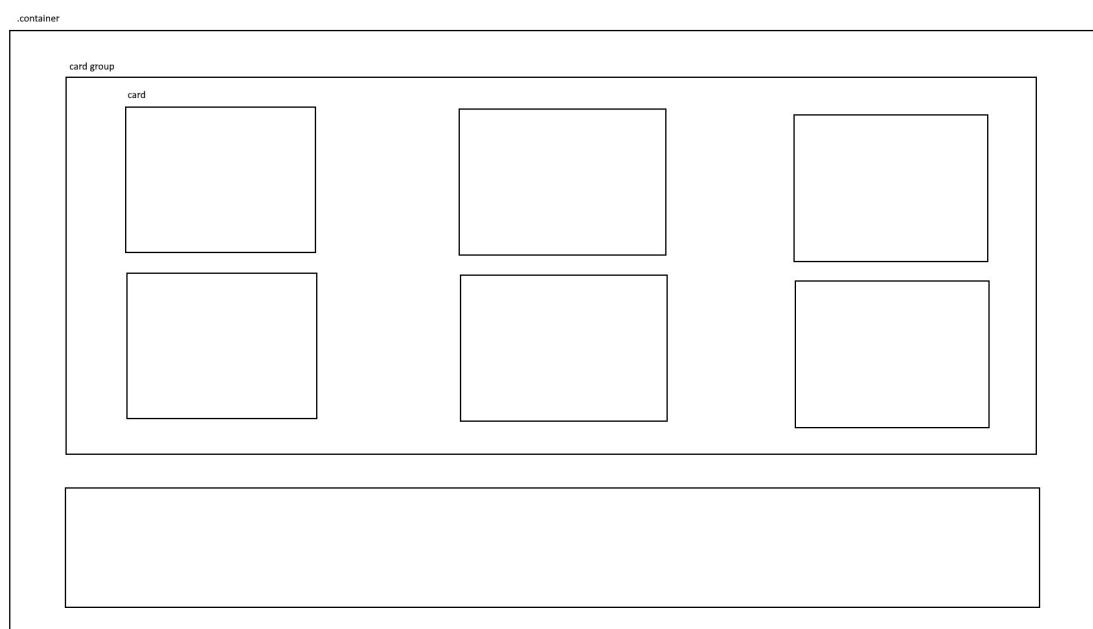
Fichero	Descripción
receta.php	Obtiene dada una ID de receta su nombre, etapa, preparación, ingredientes, etc.
registro.php	Permite registrar al usuario en el sitio web
inicioSesion.php	Dado un usuario y contraseña, devuelve si los datos son correctos
anyadirReceta.php	Crea una receta, preparación e ingredientes en la base de datos dados unos datos introducidos por el usuario
obtener3Usuarios.php	Obtiene 3 usuarios aleatorios para mostrar en la página "Recetas"
recetasEtapa.php	Obtiene las recetas de una determinada etapa (desayuno, comida o cena)
recetasAutor.php	Obtiene las recetas de un determinado autor
recetasnombreID.php	Obtiene el nombre e ID de todas las recetas
subirMensaje.php	Guarda el mensaje escrito por el usuario en la base de datos
subirFavorito.php	Añade un registro en la tabla favoritos con el usuario actual y la receta a añadir
obtenerFavoritos.php	Obtiene todos los favoritos de un determinado usuario
eliminarFavorito.php	Elimina de la tabla favoritos la receta indicada
obtenerRecetasDia.php	Obtiene 6 recetas aleatorias para mostrar en recetas del día
obtenerFechaAutorLista.php	Obtiene la fecha y el autor de una determinada ID de lista
obtenerIngredientesLista.php	Dada una ID de lista, obtiene cantidad, unidad y nombre de los ingredientes de esa lista.
eliminarIngredienteLista.php	Elimina un ingrediente de una determinada lista
eliminarTodosIngredientesLista.php	Elimina todos los ingredientes de una lista
anyadirIngredienteLista.php	Añade un ingrediente a una lista
obtenerListasIdFechas.php	Obtiene la ID y fecha de las listas de un determinado usuario
anyadirLista.php	Crea un nuevo registro en la base de datos listascompra



3.2.2.3.- Cliente

3.2.2.3.1.- Diseño de la interfaz

La interfaz está formada siempre por un contenedor de la clase “container” de bootstrap. Este contenedor proporciona márgenes y padding a la aplicación fácilmente. Luego, dependiendo de la página se cambia el contenido de la página. Por ejemplo, en las páginas en las que hay tarjetas, se usa un card group como se puede ver en la imagen de abajo. En la página receta también aparece un sistema de grid con 2 columnas, en una está la imagen y en la otra los ingredientes. En el resto de páginas, se estructura colocando los elementos unos debajo de otros. Ejemplo de la página de inicio:



Para los estilos, se han usado las clases de Bootstrap. Por ejemplo, .col y .row para definir columnas y filas, las clases de margin(m), padding(p) para añadir o no espacio, las palabras sm, lg, md... Para definir los tamaños en distintos tipos de pantalla, las imágenes que se escalan de manera correcta. Las hojas de estilo personalizadas se encuentran en el archivo styles.css de la ruta raíz y en los diferentes componentes de Angular de la aplicación. En el archivo general se definen las 2 fuentes principales que se han comentado anteriormente en la guía de estilo. Otros estilos personalizados son las tarjetas, que tienen una sombra gris y las diferentes imágenes de las tarjetas son siempre del mismo tamaño. También se incluye una separación de botones y tamaños de fuente.

3.2.2.3.2.- Accesibilidad

- Todas las imágenes de la página tienen atributos “alt”, que describen su contenido

- Los enlaces dan nombres descriptivos, exceptuando algunos que se entienden según el contexto. Tienen atributos “title” para indicar mejor su contenido. No hay enlaces vacíos. Además, se ven bien en distintas situaciones.
- No se usan tablas en la página
- En la mayoría de los campos de los formularios se usa un label para indicar el contenido del input. El botón para subir los datos de los formularios tiene nombre descriptivo, lo que permite entender mejor la operación a realizar. Por otro lado, se puede usar el teclado para enviar los formularios y los botones de enviar son del tipo input:submit. Por último, en todos los campos que no sean correctos, se indicará con un mensaje en color rojo los fallos que tienen.

A continuación se muestra la página “recetas” con el análisis de accesibilidad:

Como se puede ver en la imagen, la página no tiene ningún error y 4 características interesantes, aunque aparezcan 4 alertas no muy graves.

3.2.2.3.3.- Usabilidad

Para evaluar la usabilidad se usará el método WAMMI (Web Site Analysis and MeasureMent Inventory). Se trata de un cuestionario que contiene 20 preguntas sobre la web. Estas son las preguntas que se tratan(en inglés):



Easydishes

Statements 1 - 10 of 20

	Strongly Agree	Strongly Disagree
This website has much that is of interest to me.	<input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	○ ○ ○ ○ ○
It is difficult to move around this website.	○ ○ ○ ○ ○	<input type="radio"/> <input checked="" type="radio"/> <input type="radio"/>
I can quickly find what I want on this website.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
This website seems logical to me.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
This website needs more introductory explanations.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
The pages on this website are very attractive.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
I feel in control when I'm using this website.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
This website is too slow.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
This website helps me find what I am looking for.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
Learning to find my way around this website is a problem.	○ ○ ○ ○ ○	○ ○ ○ ○ ○

Statements 11 - 20 of 20

	Strongly Agree	Strongly Disagree
I don't like using this website.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
I can easily contact the people I want to on this website.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
I feel efficient when I'm using this website.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
It is difficult to tell if this website has what I want.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
Using this website for the first time is easy.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
This website has some annoying features.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
Remembering where I am on this website is difficult.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
Using this website is a waste of time.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
I get what I expect when I click on things on this website.	○ ○ ○ ○ ○	○ ○ ○ ○ ○
Everything on this website is easy to understand.	○ ○ ○ ○ ○	○ ○ ○ ○ ○

A continuación podemos ver un test creado por mí, aunque debería medirse con todos los usuarios posibles:



Statements 1 - 10 of 20

	Strongly Agree	Strongly Disagree			
This website has much that is of interest to me.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is difficult to move around this website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
I can quickly find what I want on this website.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
This website seems logical to me.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This website needs more introductory explanations.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
The pages on this website are very attractive.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel in control when I'm using this website.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This website is too slow.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
This website helps me find what I am looking for.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Learning to find my way around this website is a problem.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Statements 11 - 20 of 20

	Strongly Agree	Strongly Disagree			
I don't like using this website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
I can easily contact the people I want to on this website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
I feel efficient when I'm using this website.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is difficult to tell if this website has what I want.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Using this website for the first time is easy.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This website has some annoying features.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Remembering where I am on this website is difficult.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Using this website is a waste of time.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
I get what I expect when I click on things on this website.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Everything on this website is easy to understand.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3.2.2.3.4.- Desarrollo web en entorno cliente

- A. Formularios y su validación. La forma de gestionar los formularios en la aplicación es usando Reactive Forms. Esta característica permite crear formularios que se enlazan con una variable usando FormGroup con diferentes campos que se unen usando el formControlName. El formulario creado contiene todos los campos, con una o varias validaciones y hasta que no estén correctos, será inválido (la propiedad valid será falsa) y gracias a ello podemos evitar enviar los datos al servidor. Para cada error que pueda aparecer, se creará un mensaje correspondiente que mostraremos al usuario para que pueda corregirlo. Estos formularios se encuentran en las páginas añadir receta, contacto, lista, login y registro. No se usa el enlace bidireccional.
- B. Manejo y gestión de eventos. Con angular, se puede añadir eventos a cualquier elemento html utilizando los paréntesis y el nombre del evento. Casi todos los eventos son los usuales de javascript, aunque se añaden algunos. En la aplicación, se usa el evento "ngSubmit" con todos los formularios. Cuando sucede este evento la aplicación envía los datos al backend para que realice las operaciones oportunas. Otro evento que se usa, en la página "añadir receta" es el change. Esto permite que cuando el input de la imagen cambie, se pueda almacenar la información de dicho archivo para poderlo luego almacenar en BASE64 y enviarlo al servidor.



Por último, podemos ver el evento click por ejemplo en los botones de eliminar de los diferentes componentes. Lo único que hace es ejecutar una función cuando se hace click en el elemento.

- C. Gestión y almacenamiento de datos e información en el cliente. La forma elegida para guardar los diferentes datos es usando LocalStorage. Hay un servicio de angular específicamente creado para eso. En el localStorage se guarda la información de inicio de sesión una vez se ha autenticado el usuario. En función de si esta información existe, se cargarán determinados elementos o no. Por ejemplo, no se puede ver el botón de añadir a favoritos si no hay ningún usuario autenticado. También ayuda al funcionamiento del guard, que permite que sólo se pueda acceder a determinadas páginas si se ha iniciado sesión. Otro elemento que se guarda en el localStorage es la fecha, para que en el momento que cambia se generen otras 6 recetas nuevas del día y se guarden también en el almacenamiento.
- D. Modificación del DOM. De esta tarea se encarga el componente cargado en cuestión. Para modificar los elementos se usan bucles @for, @if o data binding, lo que permite cambiar dinámicamente el contenido de la web. Por ejemplo, en la página receta, usando ngOnInit() se cargan las recetas y sus propiedades (nombre, etapa, fecha, autor, imagen, ingredientes y preparación). Esto se guarda en una variable dentro del componente y automáticamente modifica el DOM usando data binding con las nuevas propiedades de la receta.
- E. Animaciones, efectos, cambios dinámicos de estilos. En determinadas páginas, se ha añadido un mensaje que sólo aparece cuando la operación es satisfactoria. En otros lugares, se ha usado un alert que avisa de lo mismo al usuario.
- F. Comunicación AJAX. La aplicación se comunica continuamente con el servidor para obtener y enviar datos en la base de datos. Para ello existen herramientas en JavaScript como las promesas o async/await. Aunque estas herramientas funcionan bien, se ha preferido usar RXJS para gestionar la asincronía.
- G. Comunicación asíncrona con el servidor. Se utilizarán observables a los que se puede suscribir usando el servicio “httpClient”. Hay un observable para cada petición a cualquiera de los archivos mostrados previamente en funciones de php. Posteriormente, se usará el método subscribe de cada observable para obtener los resultados de la petición.

3.2.3.- Fase de despliegue

3.2.3.1.- Despliegue utilizando un hosting

1. Para desplegar la aplicación era necesario usar un hosting que fuese compatible con PHP 7.4, Apache 2, MySQL y Angular 17. Era también preferible que fuese compatible con SSL para poder usar HTTPS en la web y que proporcionase un dominio propio. Debía tener la capacidad de almacenamiento necesaria para albergar la aplicación, la base de datos y otros datos como los assets de la web. Por todo esto se eligió el hosting “Loading”, aunque habría varias alternativas también validas.
2. Para subir los archivos del proyecto al servidor, se hace el comando “ng build” en Angular para genere una versión más ligera del proyecto lista para producción.



Posteriormente, se suben las carpetas de los archivos php: api e instalación, no sin antes cambiar el nombre del usuario y contraseña de la base de datos en los scripts, ya que no podemos usar root como nombre de usuario por limitaciones del hosting.

3. Para la creación de tablas, lo ideal sería generar una base de datos con sus tablas por script pero esto no era posible por lo que se ejecutaron las consultas en instalacion.php en phpMyAdmin, que era la única forma de crear bases de datos permitida por el hosting.
4. Se debe instalar la base de datos en el hosting primero, aunque en el mío venía pre-instalada. Luego se debe crear un usuario de la BDD, para lo que existe una interfaz gráfica dentro del hosting que permite hacerlo. Finalmente, en los scripts de PHP, se modifica el código para conectarse por el puerto 3306 y cambiar el nombre de usuario y contraseña a los nuevos.
5. Como tenemos configurado el hosting para que cargue las rutas de angular, no podemos acceder a los archivos PHP porque este lo ve como una ruta inexistente y muestra un error 404. La solución es usar un fichero .htaccess que permita ejecutar en algunas ocasiones la aplicación de angular y otras la api de PHP, que es muy necesaria para el correcto funcionamiento de la página.
Es recomendable añadir por lo menos 3 usuarios, 6 recetas de diferentes etapas y borrar el localStorage antes de mostrar al exterior la aplicación.

3.3- Seguimiento y control de incidencias

Para este proceso en particular, la página contiene una sección de “contacto”, en la que los usuarios pueden mandar información sobre cualquier problema que hayan tenido y la receta con la que está relacionado. Esta información será seguida de cerca por los administradores, que podrán actuar para solucionarla lo más rápido posible.

3.4.- Indicadores de calidad de procesos

- Un indicador de calidad es si está optimizado para dispositivos móviles. La página no requiere hacer scroll horizontal en estos dispositivos, evitando confundir a los usuarios. Por otro lado, el texto es legible en todas las resoluciones
- Por otro lado, las imágenes cargan correctamente, sin ser demasiado grandes cuando el espacio es pequeño.
- La página usa SSL, lo que hace que la conexión sea segura.
- Los encabezados de la web están bien definidos, con uno correspondiente por cada ruta.
- Se usan estilos de codificación uniformes en toda la aplicación

D.RECURSOS MATERIALES

4.1.- Inventario, valorado, de medios

- Ordenador personal para cada empleado.
- 2 monitores por trabajador.
- Servidor web NAS / Hosting web
- Suite de ofimática de Microsoft Office.



- Software de servidor Apache.
- Interprete de código PHP.
- Base de datos y sistema gestor de bases de datos.
- Escritorios y mesas.
- Cables de red.
- Flexos.
- Lápices y Bolígrafos.
- Pizarras.
- Base de datos MYSQL o MariaDB

4.2.- Presupuesto económico

Costes variables	Costes fijos
Gastos en reparaciones tanto del local como de las herramientas de uso que no tengan garantía — 200 - 800 €	Servidor VPS KVM 2 — 6,99 € al mes — https://www.hostinger.es/servidor-vps
Transporte de copias físicas según la demanda — 100 - 400 €	Nóminas de los empleados. 1754, 55 € por trabajador
Fabricación de copias físicas según la demanda — 1000 - 10000 €	Alquiler de la oficina - 1200 €
Pagos a los bancos por el uso de tarjeta de pago virtual — 500 - 2000 €	Contratación de la empresa de limpieza - 7680 €
	Cuota de autónomos — 230 - 500 € en 2023, 225 - 530 € en 2024
	Suministros de luz, agua, teléfono y fibra óptica — 812 €
	Tickets de descuento en cafeterías para los empleados — 200 €
	Contratación de abogado para temas legales — 1100 €
	Ordenadores y Periféricos como ratón o teclado (400 € cada uno) y monitores (150 € cada uno). Total: 400 x 10 + 150 x 10 = 5500 € .
	Microsoft Office Empresa — 200€
	Herramientas de oficina: flexos, cables de red, lápices, pizarras... — 150€

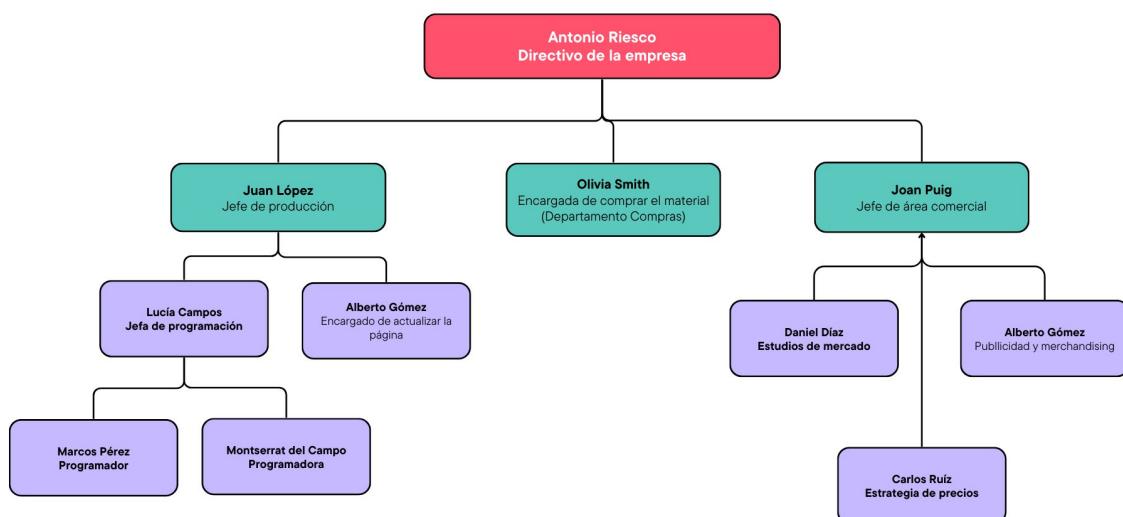
El presupuesto máximo total es **31.933,54**

E.RECURSOS HUMANOS

5.1.- Organización

Todos los empleados trabajan para el director(que sería yo), el cual intentaría hacer que todos estén a gusto con su trabajo y promover la colaboración, creatividad, etc. En el

departamento de producción, por un lado, estarán los programadores que trabajarán para mejorar la web, añadir mejoras, crear el producto... (sección desarrollo). Por otro lado, en este mismo departamento habrá un empleado trabajando para actualizar los productos de la tienda (no tendrá conocimiento técnico, sólo usará un panel de administración). Por otro lado, habrá otro trabajador encargado de comprar el material (departamento compras) necesario para el resto de personas en la empresa (ordenadores para los desarrolladores, datáfonos, etc.). El área comercial estará formada por varios empleados que se dedicarán a realizar estudios de mercado, publicidad, estrategias de precios o merchandising. No habrá departamento económico-financiero y uno de los programadores será también el empleado de administración dedicado a la atención al cliente (ganará más dinero). No hay departamento de recursos humanos.



5.2.- Contratación

Descripción del puesto	Condiciones laborales	Perfil profesional
Denominación: Programador Web Full-Stack	Salario: 1800€ brutos al mes, 3 pagas extras	<ul style="list-style-type: none"> – Formación y titulación: Grado superior Desarrollo de Aplicaciones Web o Desarrollo de Aplicaciones Multiplataforma, se valorarán cursos online realizados.
Departamento: Producción, sección desarrollo	Horario y jornada: De 8 a 13:30h y de 16 a 19h + guardias en fin de semana	Conocimientos específicos: Creación y desarrollo de aplicaciones informáticas



	Easydishes de 9 a 13:00	usando programación, teniendo capacidades de depuración y siguiendo estándares de "Clean Code"
Nivel de responsabilidad: Empleado normal	Lugar de trabajo: Oficina	Experiencia profesional: No es necesaria
Tareas a realizar: Desarrollar código Depurar código Realizar pruebas	Tipo de contrato: Indefinido	Habilidades profesionales: <ul style="list-style-type: none"> – Conocimientos de programación en JavaScript y PHP – Lenguajes HTML y CSS – Ser capaz de depurar código – Se valorará conocimiento en redes y sistemas operativos
		Actitudes y habilidades personales: <ul style="list-style-type: none"> – Capacidad de resolución de problemas – Capacidad de comprensión lectora – Trabajo en equipo – Puntualidad

Coste de personal:

Cada salario de la empresa se calculará así:

Calculo salario neto trabajador:

Salario bruto = 2000 + 100(Pagas extra prorrataeda) = 2100€

Seguridad social trabajador = 4,8% + 1,55% + 0,1% de BCCC = 6,45% de 2100 = 135,45€

IRPF del trabajador = 10% de 2100 = 210 €

Salario neto = 2100 - 135,45 - 210 = 1754,55 €

Cuotas a pagar por la empresa del trabajador contratado:

CC= 24,10% de 2100=506,1 €

CP=2% de 2100= 42€

Desempleo= 5,50% de 2100= 115,5 €

FOGASA= 0,2% de 2100= 4,2 €

FP= 0,6% de 2100 = 12,6 €

TOTAL seguridad social a cargo de la empresa = 453,60€

No habría ninguna bonificación aplicable.



5.3.- Prevención de riesgos laborales

En una empresa de informática y programación como esta, hay ciertos riesgos debidos al trabajo en la oficina o de forma remota. Aquí se detallan unos cuantos de estos:

- Ergonomía y buenos hábitos posturales. Uno de los riesgos es mantener posturas incorrectas en el trabajo, en ocasiones los empleados buscan la comodidad pero esto puede ser peligroso si mantenemos una postura que pueda provocar dolores en espalda, cuello, lumbares, etc. También se debe prestar atención a los movimientos repetitivos, que son acciones de esfuerzos y movimientos rápidos que suceden continuamente.
- Móvilario. Es importante que las sillas sean correctas: deben estar hechas de materiales cómodos, estar bien reguladas y dar un buen soporte lumbar. Por otro lado, los muebles deben estar en buen estado y no defectuosos, deteriorados o sucios. También debe haber suficiente espacio personal por trabajador.
- Prevención de incendios. Debe haber abundantes extintores, alarmas anti-incendios y salidas de emergencia. Por otro lado, se debe tener cuidado con equipos eléctricos defectuosos, cables rotos o uso incorrecto de radiadores.

F. VIABILIDAD TÉCNICA

6.1.- Estudio de viabilidad técnica

Para realizar la tarea de la programación, durante la historia ha habido distintas herramientas, visuales o no. En la actualidad se usan IDEs como Visual Studio Code (el utilizado por mí). Este programa permite el auto-completado de código, destacado de sintaxis, depuración, soporte para la mayoría de lenguajes, atajos de teclado, gestión de archivos, uso de terminal, capacidad de instalación de extensiones, etc.

El proyecto se trata de una aplicación web. Esto quiere decir que en algún momento debe usar los lenguajes HTML y CSS. Son herramientas que permiten la estructuración, maquetación y la parte visual y son esenciales en cualquier aplicación web. También se ha usado un framework de HTML y CSS llamado Bootstrap principalmente por sus diferentes componentes, su aspecto visual, la forma de maquetar y diseño responsive.

Una vez está creada la parte visual, es el turno de la interacción. Esto se hace mediante el uso de JavaScript, que es un lenguaje de programación usado por los navegadores que nos permite realizar todo tipo de funcionalidades interactivas. A partir de JavaScript, se desarrolló TypeScript, una versión con tipado estático junto con otras funcionalidades. Por último, se utilizó Angular, que es una plataforma basada en TypeScript que permite la componentización, crear aplicaciones de una sola página, creación de rutas, comunicación asíncrona con RXJS, formularios reactivos, etc.

Por otro lado, hay que pensar en la forma en la que las aplicaciones guardan información de todo tipo como en el caso de mi aplicación usuarios, contraseñas recetas o listas de la compra. Esto se consigue con una base de datos relacional. Permite organizar la información en tablas que se relacionan entre sí usando claves ajena. Estas tablas están formadas por filas y columnas, que forman registros con diferentes entradas de datos. Para comunicarse con la base de datos o realizar operaciones que no se corresponden directamente con la vista de la página se usa el lenguaje PHP. Éste es un lenguaje de programación dedicado casi exclusivamente al desarrollo web con muchas funciones como



conexión a la base de datos, compresión de archivos, uso de sesiones y cookies, enviar correos o generar imágenes y PDFs.

Para interpretar archivos PHP es necesario un software de servidor que permita acceder desde el navegador a archivos php y enviarlo al interprete de PHP para que lo transforme en una página web dinámica. El elegido es Apache2, es de código abierto y funciona en varias aplicaciones diferentes.

Finalmente, es necesario llevar a cabo un seguimiento de los proyectos de programación en repositorios, que nos permita crear varias ramas a las que se puede hacer "merge", volver atrás del proceso de desarrollo, ver el estado del desarrollo, arreglar conflictos y colaborar con otros programadores. Para esto se usa git. Junto a GIT se usa GitHub, que ofrece un servicio en la nube para albergar diferentes repositorios.

G. VIABILIDAD ECONÓMICO-FINANCIERA

7.1.- Inversiones y gastos

Para empezar, vamos a clasificar los elementos de gestión financiera de la empresa en gastos e inversiones. Los gastos que vamos a asumir son el alquiler de las oficinas, sueldos y seguridad social de los empleados, material ofimático, suministros, autónomos y abogados además de otros imprevistos que puedan ser necesarios.

Las inversiones van a suponer un gran beneficio a la empresa, ya que harán uso de ellas muy a menudo en el día a día. Entre otras, de mayor a menor importancia, encontramos software y hardware informático, instalaciones, mobiliario e inversión en otras empresas que puedan generar valor. Ahora vamos a observar con mayor detalle cada una de estas partidas:

Gastos:

- Alquiler de las oficinas —> Este gasto depende de el lugar geográfico donde se encuentre, de su tamaño, de la cantidad de personal que va a trabajar. Unas oficinas de tamaño normal suelen rondar los 7000 € en una ciudad como Madrid.
- Sueldos —> Hay dos tipos de empleados con sus correspondientes sueldos. El primero, de rango menor, cobrará 1800 €(Hay 7) y el segundo 2100 € (Hay 3). Total: $1800 * 7 + 2100 * 3 = 18900$ €.
- Seguridad social —> Podemos establecer que aproximadamente, un 30% del salario bruto pueda ser la cantidad necesaria para la seguridad social. Podemos calcular el dinero: $0,3 \times 1800 \times 7 + 0,3 \times 2100 \times 3 = 3780 + 1890 = 5670$ €.
- Material ofimático —> Esto incluye bolígrafos, post-its, borradores, folios, etc. Aproximadamente 200 €.

- Luz —> 70€
- Agua —> 25 €
- Teléfono —> 10€
- Fibra óptica —> 30 €
- Total Suministros —> $70 + 25 + 10 + 30 = 135$ €.
- Autónomos —> 400 €.
- Abogados —> 1200 €
- Total gastos —> $7000 + 18900 + 5670 + 200 + 70 + 25 + 10 + 30 + 135 + 400 + 1200 = 33640$ €.

Inversiones:



- Software informático —> Gran parte del software como Visual Studio Code, HTML o Javascript va a ser gratis y/o de código abierto. Los de pago son licencias OEM de Windows 10 u 11 (12 € cada licencia), MongoDB Dedicated (52,26 €) y Github Copilot (4 €). Total: $12 \times 10 + 52,26 + 4 = 176,26$ €.
- Hardware informático —> Servidor VPS (6,99 €), Ordenadores y Periféricos como ratón o teclado (400 € cada uno) y monitores (150 € cada uno). Total: $6,99 + 400 \times 10 + 150 \times 10 = 5506,99$ €.
- Instalaciones —> Acondicionar el lugar es importante para que los trabajadores se sientan cómodos y puedan ser productivos. En este proceso se van a añadir electricidad, dar luminosidad a la oficina, hacer funcionar el servicio, limpiar, etc. El precio suele rondar los 3000 €.
- Mobiliario —> Debemos preocuparnos por establecer un ambiente cómodo y ergonómico. Las mesas deben ser amplias, las sillas regulables por el trabajador, las luces deben ser lo suficientemente potente pero sin llegar a deslumbrar. Dependiendo de la gama del mobiliario, nos gastaríamos aproximadamente 6000 €.
- Inversión en otras empresas —> Se trata de realizar inversiones en empresas que tengan potencial para aportar beneficios extra y crecer más rápido. Va a estar compuesto por empresas que se prevé que puedan crecer en el futuro. Por ello, muchas pertenecen al mismo sector que nuestra empresa, ya que se confía en que esta crezca. Se destinarían unos 2000 €.

Total inversiones —> $176,26 + 5506,99 + 3000 + 6000 + 2000 = 16.683,25$ €

7.2.- Financiación

En este caso voy a optar por las fuentes de financiación ajenas para las fuentes de financiación principales. La primera de ellas es el préstamo bancario. Esta fuente se usará para pagar las necesidades a largo plazo. Esto incluye principalmente los gastos de varios años. Ya sabemos que estos gastos van a ser una cantidad fija o casi fija, y esa cantidad es la que dispondremos en el préstamo.

Por otro lado, usaremos el crédito bancario para financiar las inversiones. Este es un gasto que hemos calculado pero que puede cambiar en función de varios factores, por lo que dispondremos de una gran cantidad en la cuenta por si acaso pero sólo sacaremos lo necesario para las inversiones. Además, es a corto plazo y las inversiones sólo habrá que pagarlas al inicio.

Finalmente, añadiremos una fuente de financiación propia y externa, que son los Business Angels. Se intentará que haya inversores individuales que ofrezcan capital, contactos y conocimientos técnicos, ya que confían en el producto desde el principio (start-up).

7.3.- Viabilidad económico-financiera

Plan de tesorería (Basado en los datos aportados al principio del plan financiero, variando ligeramente la cantidad en función del mes, ya que los precios van cambiando)

	Enero	Febrero	Marzo	Abril	Mayo	Junio
Saldo inicial	5000	277185	542535	758975	925305	1041610
Cobros / Entradas	300000	300000	250000	200000	150000	100000
Ventas mercaderías	200000	200000	170000	150000	80000	50000



Easydishes

Publicidad	100000	100000	80000	50000	70000	50000
Pagos / Salidas	27185	34.650	33.560	33670	33695	33820
Alquiler	7000	7100	7100	7200	7200	7300
Sueldos	18900	20000	18900	18900	18900	18900
Seguridad Social	5670	5670	5670	5670	5670	5670
Material ofimático	180	140	150	160	180	200
Suministros(Luz, agua, fibra óptica y teléfono)	135	140	140	140	145	150
Autónomos	400	400	400	400	400	400
Abogado	1200	1200	1200	1200	1200	1200
Saldo final mes	277815	542535	758975	925305	1041610	1107790

	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Saldo inicial	1107790	1183990	1260170	1315150	1381210	1447070
Cobros / Entradas	110000	110000	90000	100000	100000	100000
Ventas mercaderías	50000	50000	30000	35000	40000	40000
Publicidad	60000	60000	60000	65000	60000	60000
Pagos / Salidas		33820	35020	33940	34140	34150
Alquiler	7300	7300	7400	7400	7400	7400
Sueldos	18900	18900	20000	18900	18900	18900
Seguridad Social	5670	5670	5670	5670	5670	5670
Material ofimático	180	190	190	200	190	200
Suministros(Luz, agua, fibra óptica y teléfono)	150	160	160	170	180	180
Autónomos	400	400	400	400	400	400
Abogado	1200	1200	1200	1200	1400	1400
Saldo final mes	1183990	1260170	1315150	1381210	1447070	1512920

Cálculo de perdidas y ganancias. Estos son los gastos e inversiones que llevaremos a cabo en la empresa (Vistos previamente y añadidos algunos):

- Alquiler — 7000 €
- Sueldos — 18900 €
- Seguridad social — 5670 €
- Material ofimático — 200 €
- Luz — 70 €
- Agua — 25 €
- Teléfono — 10 €
- Fibra óptica — 30 €
- Autónomos — 400 €



– Abogados — 1200 €

Los inmovilizados que se amortizarán son:

Ordenadores — 3000 €(26%)

Instalaciones — 2000 €(10%)

Ahora dispondremos los gastos e ingresos en la tabla

INGRESOS DE EXPLOTACIÓN	Publicidad (+100000) Ventas del producto tangible(50000) e intangible (+150000)
GASTOS DE EXPLOTACIÓN	Alquiler(-7000) Sueldos(-18900) Seguridad social(-5670) Material ofimático (-200) Luz(-70) Agua(-25) Teléfono(-10) Fibra óptica(-30) Autónomos(-400) Abogados(-1200) Amortización Ordenadores (-780) Amortización Instalaciones (-200)
RESULTADO DE EXPLOTACIÓN	565515
INGRESOS FINANCIEROS	
GASTOS FINANCIEROS	Intereses préstamo(-6000)
RESULTADO FINANCIERO	-6000
RESULTADO ANTES DE IMPUESTOS	559.515
RESULTADO DEL EJERCICIO (25%)	419.636,25 €

El siguiente cálculo será el del balance de situación:

Activo:

Activo no corriente	Material ofimático Servidores, ordenadores y otros equipos informáticos	– 200 – 7000	7200
Activo Corriente: existencias	Mercaderías (Producto Físico)	– 50000	50000
Activo Corriente: realizable	Deudas de cliente IVA Soportado Para calcular el IVA necesitamos el precio del producto (10 € versión lite): $10 + 0,21 * 10 * 200000 = 420010$	– 1000 421010	
Activo Corriente: disponible		1512920 (Ver plan de tesorería)	1512920
Total activo			1991130

Pasivo y Patrimonio Neto

Patrimonio Neto	Capital inicial del empresario	– 5000 € – 302584	727220,25
-----------------	-----------------------------------	----------------------	-----------



	individual Reservas (20% del resultado de tesorería) Resultado del ejercicio	- 419636,25	
Pasivo no corriente	No hay préstamos		
Pasivo corriente	Préstamo IVA Repercutido	6000 € 420010 €	426010
Total pasivo + patrimonio neto			1153230,25

H. CONCLUSIÓN

Se puede ver que dentro de las herramientas usadas, las más baratas son las de software. Esto es así porque dichas herramientas son gratuitas e incluso algunas de código abierto. La elección de ellas es correcta, ya que son tecnologías con mucho soporte, robustas, compatibles y bastante escalables. Además, son bastante accesibles para cualquier trabajador que las use. Tal vez el problema sea que se necesita una formación previa sobre ellas y, por tanto, si alguien no sabe puede influir en la productividad de la empresa. El uso de IDEs es fundamental, permitiendo al usuario realizar acciones nuevas rápidamente. La creación del producto ha sido satisfactoria pero ha habido distintos retos dentro del desarrollo que ha habido que ir solucionando, ya que me he encontrado varias cosas que cambiar con respecto a la propuesta inicial del proyecto.

La viabilidad económica en mi opinión es correcta. Está pensada para una empresa de desarrollo que pueda obtener grandes beneficios ya que la demanda actual es abundante, por lo que la inversión inicial es grande pero como se puede ver, los ingresos también. Si se atiende al análisis de mercado, se puede ver que aunque hay gran demanda de páginas de recetas, hay un nicho con respecto a la funcionalidad de la lista de la compra y los favoritos. Por otro lado, los sueldos y el tipo de contrato de los trabajadores es correcto y busca ser justo para todos los empleados.

BIBLIOGRAFÍA / WEBGRAFÍA

- <https://www.tiobe.com/tiobe-index/>
- <https://es.statista.com/map/europa/espana/tecnologia-y-telecomunicaciones>
- <https://www.w3.org/WAI/standards-guidelines/es>
- <https://www.atlassian.com/es/agile/project-management/gantt-chart>
- Diagrama de casos de uso – Universidad Politécnica de Valencia:
<https://riunet.upv.es/bitstre>
<am/handle/10251/174393/Moreira%20-%20Desarroll>
<o%20de%20una%20aplicacion%20web%20para%20la%20gestion%20de%20recetas%20de%20cocina.pdf?sequence=1>
<https://fonts.google.com/specimen/Roboto>
<https://balsamiq.com/>
<https://getbootstrap.com/>
<https://www.youtube.com/watch?v=i5c6oREVRFo>
- Para hacer el mapa de navegación:

EasyDishes

Easydishes



CENTRO INTEGRADO DE
FORMACIÓN PROFESIONAL
Nº 1 - CUENCA

<https://www.canva.com/>

Para hacer el diagrama E-R:

<http://dia-installer.de/shapes/ER/index.html.es>

<https://www.w3schools.com/MySQL/default.asp>

Para otros recursos gráficos:

<https://www.getpaint.net/>

<https://wave.webaim.org/>

<http://www.wammi.com/>

<https://www.youtube.com/watch?v=f7unUpshmpA>

Información del hosting:

<https://www.loading.es/blog/>

Libro Empresa E Iniciativa Emprendedora

<https://miro.com/es/graficos/que-es-organigrama/>

<https://www.sepe.es/HomeSepe/que-es-el-sepe/comunicacion-institucional/publicaciones/publicaciones-oficiales/listado-pub-empleo/guia-contratos.html>

<https://www.sepe.es/HomeSepe/que-es-el-sepe/comunicacion-institucional/publicaciones/publicaciones-oficiales/listado-pub-empleo/bonificaciones-reducciones-contratacion-laboral.html>

<https://www.prolaboral.com/es/blog/prevencion-de-riesgos-laborales-en-oficinas.html>

<https://www.accessibilitychecker.org/>

<https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>