

# Recipe Repo

**Luis García Martín**

Grado en ingeniería informática  
Ingeniería del software

**Oriol Martí Girona**

**Santi Caballé Llobet**

10/1/1918

Copyright © ANY EL-TEU-NOM.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

### **C) Copyright**

© (l'autor/a)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Recipe Repo</i>
<b>Nombre del autor:</b>	<i>Luis García Martín</i>
<b>Nombre del consultor/a:</b>	<i>Oriol Martí Girona</i>
<b>Nombre del PRA:</b>	Santi Caballé Llobet
<b>Fecha de entrega (mm/aaaa):</b>	<i>01/2018</i>
<b>Titulación o programa:</b>	<i>Grado en ingeniería informática</i>
<b>Área del trabajo final:</b>	<i>Trabajo final de grado en el área de ingeniería del software.</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave:</b>	<i>Recetas, salud, dieta</i>
<b>Resumen del trabajo</b>	
<p>La obesidad en España y en otros países se ha convertido en una alarma social. Por este motivo surge la idea de crear una aplicación que ayude a las personas a mantener una vida más saludable.</p> <p>Esta herramienta une los conceptos, cocina y salud. Así pues, no está sólo enfocada a mostrar información de recetas. La herramienta ofrece un buscador de recetas mediante filtros, permite crear una planificación de recetas según unos valores nutricionales y permite generar una dieta según el peso, edad, altura y sexo del usuario. Este generador de dietas mostrará unas pautas a seguir y un listado de recetas acordes con el usuario para llevar a cabo su dieta.</p> <p>Esta aplicación se ha desarrollado de cero ya que no se ha encontrado ninguna posibilidad de hacerlo a partir de otra. Respecto a la metodología escogida, se ha optado por seguir la <i>Relationship Management Methodology</i> que se explica al detalle en el apartado 1.3 enfoque y métodos seguidos.</p> <p>El resultado obtenido han sido una aplicación web accesible desde Internet, pero no se ha podido cumplir con todos los objetivos planificados. Así pues, las funciones planificar o generar una dieta han quedado a medio desarrollo. No obstante, existe documentación y diagramas para finalizarlas correctamente.</p> <p>Por último, en las conclusiones del trabajo se comentan aspectos sobre lecciones aprendidas, objetivos logrados, análisis sobre la planificación, metodologías y líneas de trabajo futuras.</p>	

## **Abstract**

The obesity in Spain and other countries has become a social alarm. For this reason, appears the idea developing an application that helps people be careful your health.

This tool joins the concepts cooking and Health. So that, this application offers more than cooking information. This tool allows search recipes through a filter, create a recipe planning about nutrition values and generate a diet according to weight, age, height and sex. This dietary generator will show a guidelines and recipes list according to user to carry out his diet.

This application has been developed from scratch as no possibility was found to make it starting from other. Respect the chosen methodology, has opted to follow the Relationship Management Methodology that is explained in the section 1.3 *enfoque y métodos seguidos*.

The result obtain has been a web application accessible from Internet, but hasn't been fulfil with all objectives. So that, the planning and diet generator functions are not finished. However, exist documentation and diagrams for finish them correctly.

Finally, the conclusions where aspects are comment about lessons learned, objectives achieved, analysis planning, methodology and lines of work.



# Índice

<b>1. Introducción</b> .....	1
1.1 Contexto y justificación del trabajo .....	1
1.2 Objetivos de trabajo .....	3
1.3 Enfoque y método seguido .....	4
1.4 Planificación del trabajo .....	6
1.5 Breve resumen de los productos obtenidos .....	9
1.6 Breve descripción de los otros capítulos de la memoria .....	10
<b>2. Resto de capítulos</b> .....	11
2.1 Análisis .....	11
2.1.1 Análisis de requisitos .....	11
2.1.2 Historias de usuario .....	13
2.1.3 Diagrama de datos .....	15
2.1.4 Diagramas de la arquitectura de la aplicación .....	16
2.1.5 Diseño lógico de la base de datos .....	18
2.1.6 Tecnologías utilizadas .....	19
2.2 Diseño .....	20
2.2.1 Casos de uso .....	20
2.2.2 Correcciones .....	22
2.2.2.1 Diagrama de datos .....	22
2.2.2.2 Diseño lógico .....	23
2.2.2.3 Arquitectura de la aplicación .....	24
2.2.3 Diseño físico de la base de datos .....	28
2.2.4 Diseño de pantallas .....	29
2.2.5 Código .....	35
2.2.6 Diagrama de secuencias .....	41
2.2.7 Configuración del entorno local y real .....	43
2.3 Valoraciones económicas .....	44
2.4 Riesgos .....	45
2.5 Mantenimiento .....	46
2.6 Ampliaciones futuras .....	47
<b>3. Conclusiones</b> .....	50
<b>4. Glosario</b> .....	52
<b>5. Bibliografía</b> .....	53

# 1. Introducción

## 1.1 Contexto y justificación del trabajo

*Según datos del Estudio Nutricional de la población española más del 60% de la población sufre de obesidad. El 39,3 % de los españoles sufren sobrepeso y el 21, 6 obesidad, por lo que más de la mitad de la población (60,9%), y en comparación con datos precedentes, muestra un aumento importante de la sobrecarga ponderal, según datos del Estudio Nutricional de la Población Española (ENPE), que acaba de publicar la Revista Española de Cardiología [1].*

Pero lo más alarmante es que estos datos van aumentando cada año. Por otro lado, la sociedad no es consciente de los riesgos que conlleva la obesidad. Algunas de las consecuencias más comunes pueden llevar a que las personas sufran enfermedades cardiovasculares, trastornos del aparato locomotor y algunos tipos de cáncer.

Para solucionar este problema, lo primero es reconocerlo como tal y después contar con información sobre los alimentos y recetas culinarias para mantener una dieta equilibrada sin dejar de disfrutar de la comida. Así pues, lo que se pretende con este trabajo es que los usuarios puedan disponer de una herramienta que les ayude a mantener una alimentación más saludable.

Son muchas las aplicaciones web que permiten buscar recetas por categorías u otro criterio. En Internet se puede encontrar hogarmania, alimentación, facilísimo o el buscador de recetas Recipe View de Google por nombrar unas cuantas. Unas permiten navegar sobre una clasificación y otras introduciendo palabras clave.

Centrando la atención en Recipe View, se puede decir que esta aplicación va mucho más allá que un simple buscador. Por un lado, ofrece dos criterios de búsqueda: por ingredientes o por recetas de chefs. Por otro lado, permite refinar la búsqueda por tiempo de cocción y número de calorías. ¿Y dónde se encuentra toda la información? Todas estas recetas se obtienen de las webs que etiquetan sus recetas con el microformato hRecipe. De este modo el buscador es capaz de representar correctamente la información.

Google pensó en los amantes de la cocina y los chefs amateurs y lanzó *Recipe View*, una opción dentro de su buscador que permite rastrear recetas, consejos y fotografías para probar distintos platos en la cocina del hogar.

*"Recipe View permite restringir los resultados de búsqueda para mostrar sólo las recetas, y le ayuda a elegir la adecuada entre los resultados de búsqueda, mostrando puntuaciones, ingredientes y fotos", indica la compañía desde su blog.*

**Cómo funciona.** *El usuario debe ingresar el nombre de un ingrediente o receta, y una vez que aparecen los resultados de búsqueda, hacer clic en la opción "Recipes" en la columna de la izquierda, donde también aparecen las opciones "noticias", "imágenes" y "tiempo real", entre otras.*

*Así, el buscador filtrará las opciones para que el usuario sólo encuentre recetas. También está la opción de depurar el rastreo por ingrediente, tiempo de cocción y calorías.*

*Recipe View también permite que el usuario conozca la valoración de las recetas por el sistema de evaluación con estrellas.*

*"Recipe View es parte de nuestros continuos esfuerzos para enriquecer la experiencia de búsqueda a partir de datos estructurados, y este lanzamiento es un hito emocionante para nuestro equipo técnico (...)", indicó Google [2].*

No obstante, no existe en la actualidad aplicaciones dentro del ámbito culinario que ofrezcan funcionalidades para mejorar la salud de las personas. La idea de poder unir recetas de cocina con salud de un modo más amplio no está tan generalizada. Así pues, la idea surgió de la necesidad de poder unir los dos conceptos, cocina y salud. Por lo tanto, esta aplicación no será un simple buscador de recetas. La herramienta permitirá entre otras cosas, crear una planificación de recetas según unos valores nutricionales o generar una dieta según el peso, edad, altura y sexo del usuario.

## **1.2 Objetivos del trabajo**

Estos son los objetivos que se pretenden conseguir con este trabajo:

### **Un buscador de recetas mediante filtros**

Esta funcionalidad estará disponible para todos los usuarios estén o no registrados. El usuario podrá obtener un listado de recetas según unos criterios de búsqueda. Así pues, se podrá escoger la categoría: carnes, pescados y mariscos, pastas, arroces, etc. y, por otro lado, la información nutricional: porcentaje de grasas, hidratos de carbono, proteínas y sal. Esta función mostrará una lista de recetas acorde con los criterios de búsqueda.

### **Un planificador de recetas**

Si el usuario está registrado podrá generar un planificador de recetas. El planificador permitirá al usuario obtener una planificación de recetas diferentes durante un intervalo de tiempo con tres recetas por día (desayuno, comida y cena). Deberá escoger el tipo de alimentación: alimentación variada, alimentación variada y saludable y alimentación proteica y el máximo de días (máximo dos semanas).

### **Un generador de dietas**

Si el usuario está registrado también tendrá acceso a generar una dieta adelgazante o para deportistas. Deberá añadir los valores siguientes: peso, edad, altura y sexo y el sistema generará la dieta. Además, mostrará una lista de recetas relacionadas con la dieta.

### **Administración de la aplicación**

Respecto a las funciones de mantenimiento de los datos de la aplicación habrá un área de gestión (backend) que ofrecerá funciones como crear, editar y eliminar usuarios, recetas e ingredientes.

### 1.3 Enfoque y métodos seguidos

Una vez realizado un estudio sobre las aplicaciones existentes que nos ofrecen listados de recetas por categorías u otros criterios he llegado a la conclusión de que la mejor estrategia era comenzar de cero.

Actualmente hay muchas aplicaciones en Internet relacionadas con la gastronomía. No obstante, al ser aplicaciones webs y no aplicaciones de código abierto, la opción de desarrollar un producto a partir de otro ya existente se ha descartado. Si se hubiera encontrado algún sistema que por ejemplo contara con la funcionalidad de mostrar recetas hubiese ayudado mucho para poder centrar el proyecto en otras funciones como el planificador o el generador de dietas.

Por otro lado, hay otro motivo que también ha influido en la decisión de desarrollar el proyecto de cero. Este ha sido la necesidad de crear un *framework* sencillo para aprovechar las ventajas que ofrece como más facilidad para desarrollar y ayuda en el mantenimiento de la aplicación. Así mismo, al ser un marco de trabajo sencillo, se ha tenido que invertir menos tiempo que si se hubiera escogido la opción de desarrollar la aplicación con el *framework* de Symfony por poner un ejemplo.

El último factor que también ha influido en la estrategia escogida ha sido desarrollar una aplicación siguiendo la descripción del patrón modelo, vista y controlador (MVC). Las ventajas que ofrece este patrón arquitectónico son la reutilización de código, ayuda en el desarrollo y mantenimiento lo que hacen que esta solución sea muy interesante y práctica. Por lo tanto, era muy poco probable encontrar un sistema de software que reuniera todas estas características que se han comentado y así poder partir de una base ya implementada.

Respecto a la metodología seguida se ha optado por la RMM *Relationship Management Methodology*.

*RMM Relationship Management Methodology: se define como un proceso de análisis, diseño y desarrollo o de aplicaciones hipermedia. Esta metodología es apropiada para dominios con estructuras regulares, es decir, con clases de objetos bien definidas, y con claras relaciones entre esas clases. Por ejemplo, catálogos o bases de datos tradicionales. El modelo propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipermedia de la aplicación. Los objetos del dominio se definen con la ayuda de entidades, atributos y relaciones asociativas, sus principales características son:*

- *Aproximación para el diseño de sitios web, bajo una aproximación centrada en la información.*
- *Lenguaje de modelado de sitios web a nivel lógico (dominio de información + estructuras de navegación + elementos de presentación)*
- *Integrado en una metodología de desarrollo*

- *Facilitar la estructuración de páginas web complejas que contienen elementos de distintas entidades (vistas múltiples)*
- *Permitir la reutilización de elementos en el diseño (vistas jerárquicas)*
- *Diseño de enlaces más potentes y versátiles*
- *Mantener el contexto durante la navegación [3]*

## **1.4 Planificación del trabajo**

Los recursos necesarios para realizar el proyecto se clasificarán en recursos humanos y primarios.

### **Recursos humanos**

En primer lugar, han sido necesarias 40 horas para realizar tareas de planificación, estudio de alternativas, análisis de requisitos y otras tareas necesarias. En segundo lugar, en la fase de análisis, se ha requerido de unas 144 horas para cumplir con las tareas siguientes: historias de usuario, diagramas de clases y arquitectura de la aplicación y diseño lógico de la base de datos. En tercer lugar, han sido necesarias unas 208 horas para llevar a cabo tareas de diseño y desarrollo. Por último, en la fase final, se han dedicado unas 44 horas para elaborar la memoria y la presentación en PowerPoint.

### **Recursos primarios**

Los recursos primarios que se han necesitado para el proyecto son los siguientes: Un ordenador personal con conexión a Internet, varios programas de software para realizar tareas de análisis, diseño y desarrollo y contratación de servicios de hosting y dominio.

### **Diagrama de Gantt**

Ha habido cambios importantes en la planificación que se hizo inicialmente y que han provocado retrasos en la entrega de la fase de diseño (PAC 3) y tareas de esta misma fase que se han descartado por falta de tiempo. Este contratiempo ha provocado a su vez que la fase final (PAC 4) tuviera que volverse a planificar con menos tiempo del que se había planificado al inicio. A continuación, se muestra el diagrama de Gantt inicial y el final para ver esos cambios.

# Inicial

Listo	Nombre de la tarea	Fecha de Inicio	Fecha final	Asignado a	% Cumplimiento	Duración
<input type="checkbox"/>	<b>- Plan de trabajo (PAC1)</b>	<b>21/09/17</b>	<b>28/09/17</b>			6d
<input type="checkbox"/>	Contexto de la aplicación	21/09/17	21/09/17			1d
<input type="checkbox"/>	Estudio de alternativas existentes	22/09/17	22/09/17			1d
<input type="checkbox"/>	<b>- Análisis de requisitos</b>	23/09/17	24/09/17			1d
<input type="checkbox"/>	funcionales	23/09/17	23/09/17			1d
<input type="checkbox"/>	no funcionales	24/09/17	24/09/17			1d
<input type="checkbox"/>	Metodologías utilizadas	25/09/17	25/09/17			1d
<input type="checkbox"/>	Propuesta de tecnologías	26/09/17	26/09/17			1d
<input type="checkbox"/>	Diagrama de gantt	28/09/17	28/09/17			1d
<input type="checkbox"/>	<b>- Análisis (PAC2)</b>	<b>03/10/17</b>	<b>03/11/17</b>			24d
<input type="checkbox"/>	Historias de usuario	03/10/17	09/10/17			5d
<input type="checkbox"/>	Diagrama de la arquitectura	10/10/17	13/10/17			4d
<input type="checkbox"/>	Diagrama de clases	14/10/17	25/10/17			9d
<input type="checkbox"/>	Diseño lógico de la base de datos	26/10/17	03/11/17			7d
<input type="checkbox"/>	<b>- Diseño (PAC3)</b>	<b>07/11/17</b>	<b>20/12/17</b>			32d
<input type="checkbox"/>	Diseño físico de la base de datos	07/11/17	10/11/17			4d
<input type="checkbox"/>	Interficie gráfica de usuario	11/11/17	15/11/17			4d
<input type="checkbox"/>	Gestión de usuarios	16/11/17	20/11/17			3d
<input type="checkbox"/>	Gestión de recetas e ingredientes	21/11/17	24/11/17			4d
<input type="checkbox"/>	Buscador de recetas	25/11/17	04/12/17			7d
<input type="checkbox"/>	Planificador de recetas	05/12/17	13/12/17			7d
<input type="checkbox"/>	Dietas para adelgazar/deporte	14/12/17	20/12/17			5d
<input type="checkbox"/>	<b>- Entrega final (PAC4)</b>	<b>26/09/17</b>	<b>20/01/18</b>			84d
<input type="checkbox"/>	<b>- Pruebas</b>	26/09/17	29/12/17			69d
<input type="checkbox"/>	Diseño de las pruebas	22/12/17	25/12/17			2d
<input type="checkbox"/>	Generación de los casos de prueba	26/09/17	26/09/17			1d
<input type="checkbox"/>	Definición de los procedimientos de prueba	27/09/17	27/09/17			1d
<input type="checkbox"/>	Ejecución de las pruebas	28/09/17	28/09/17			1d
<input type="checkbox"/>	Documentación de las pruebas	29/12/17	29/12/17			1d
<input type="checkbox"/>	<b>- Memoria</b>	30/09/17	17/01/18			79d
<input type="checkbox"/>	Resumen del trabajo	30/09/17	31/10/17			23d
<input type="checkbox"/>	Abstract	01/11/17	03/11/17			3d
<input type="checkbox"/>	Introducción	04/01/18	04/01/18			1d
<input type="checkbox"/>	Análisis	05/01/18	08/01/18			2d
<input type="checkbox"/>	Diseño	09/01/18	11/01/18			3d
<input type="checkbox"/>	Valoraciones económicas	11/01/18	11/01/18			1d
<input type="checkbox"/>	Riesgos	12/01/18	12/01/18			1d
<input type="checkbox"/>	Mantenimiento	13/01/18	13/01/18			1d
<input type="checkbox"/>	Ampliaciones futuras	14/01/18	14/01/18			1d
<input type="checkbox"/>	Conclusiones	15/01/18	15/01/18			1d
<input type="checkbox"/>	Glosario	16/01/18	16/01/18			1d
<input type="checkbox"/>	Bibliografía	17/01/18	17/01/18			1d
<input type="checkbox"/>	Presentación en video	18/01/18	19/01/18			2d
<input type="checkbox"/>	Entrega de la aplicación o prototipo	20/01/18	20/01/18			1d



# Final

1		[-]	<b>Plan de trabajo (PAC1)</b>	21/09/17	01/10/17	100%	11d
2			Contexto de la aplicación	21/09/17	22/09/17	100%	2d
3			Estudio de alternativas existentes	23/09/17	24/09/17	100%	2d
4		[-]	<b>Análisis de requisitos</b>	25/09/17	26/09/17	100%	2d
5			funcionales	25/09/17	25/09/17	100%	1d
6			no funcionales	26/09/17	26/09/17	100%	1d
7			Metodologías utilizadas	27/09/17	28/09/17	100%	2d
8			Propuesta de tecnologías	29/09/17	30/09/17	100%	2d
9			Diagrama de gantt	01/10/17	01/10/17	100%	1d
10		[-]	<b>Análisis (PAC2)</b>	02/10/17	06/11/17	100%	36d
11			Historias de usuario	02/10/17	07/10/17	100%	6d
12			Diagrama de clases	08/10/17	20/10/17	100%	13d
13			Diagramas de la arquitectura de la aplicación	21/10/17	31/10/17	100%	11d
14			Diseño lógico de la base de datos	01/11/17	06/11/17	100%	6d
15		[-]	<b>Diseño (PAC3)</b>	07/11/17	28/12/17	100%	52d
16			Casos de uso	07/11/17	07/11/17	100%	1d
17		[-]	<b>Correcciones sobre errores detectados de la fase de análisis</b>	08/11/17	10/11/17	100%	3d
18			Diagrama de datos	08/11/17	08/11/17	100%	1d
19			Diagrama de la arquitectura de la aplicación	09/11/17	09/11/17	100%	1d
20			Diseño lógico de la base de datos	10/11/17	10/11/17	100%	1d
21			Diseño físico de la base de datos	11/11/17	11/11/17	100%	1d
22			Configuración del entorno de desarrollo local	12/11/17	12/11/17	100%	1d
23		[-]	<b>Desarrollo de la aplicación</b>	13/11/17	26/12/17	100%	44d
24			Interficie gráfica de usuario	13/11/17	17/11/17	100%	5d
25			Inicio de sesión y registro	18/11/17	22/11/17	100%	5d
26			Gestión de usuarios	23/11/17	28/11/17	100%	6d
27			Gestión de recetas	29/11/17	09/12/17	100%	11d
28			Gestión de ingredientes	10/12/17	14/12/17	100%	5d
29			Buscador de recetas	15/12/17	22/12/17	100%	8d
30			Multidioma	23/12/17	26/12/17	100%	4d
31			Planificador de recetas			0%	0d
32			Generador de dietas			0%	0d
33			Selección y contratación del hosting y dominio	27/12/17	27/12/17	100%	1d
34			Configuración del entorno de desarrollo real	28/12/17	28/12/17	100%	1d
35		[-]	<b>Entrega final (PAC4)</b>	29/12/17	13/01/18	100%	16d
36		[-]	<b>Memoria</b>	29/12/17	11/01/18	100%	14d
37			Resumen del trabajo y abstract	29/12/17	29/12/17	100%	1d
38		[-]	<b>Introducción</b>	30/12/17	02/01/18	100%	4d
39			Contexto y justificación del trabajo y objetivos de trabajo	30/12/17	31/12/17	100%	1d
40			Enfoque y métodos seguido y planificación de trabajo	01/01/18	01/01/18	100%	1d
41			Breve resumen de los productos obtenidos y descripción de los otros capítulos de la memoria	02/01/18	02/02/18	100%	1d
42		[-]	<b>Resto de capítulos</b>	03/01/18	08/01/18	100%	6d
43			Análisis y diseño	03/01/18	05/01/18	100%	3d
44			Desarrollo y valoraciones económicas	06/01/18	06/01/18	100%	1d
45			Riesgos y mantenimiento	07/01/18	07/01/18	100%	1d
46			Ampliaciones futuras	08/01/18	08/01/18	100%	1d
47			Conclusiones	09/01/18	10/01/18	100%	2d
48			Glosario, Bibliografía y anexos	11/01/18	11/01/18	100%	1d
49			Presentación en powerpoint	12/01/18	13/01/18	100%	2d

## **1.5 Breve sumario de productos obtenidos**

### **Memoria**

Este documento que presenta los apartados siguientes:

- Introducción
- Resto de capítulos
- Conclusiones
- Glosario
- Bibliografía

### **Presentación**

Una presentación del trabajo en PowerPoint de unas veinte páginas con audio que será un resumen del contenido de la memoria.

### **Implementación**

El código completo de la aplicación que se entrega en un archivo .Zip y la aplicación web que se puede ver en este enlace:  
<http://www.reciperepo.net/PAC3/>

## **1.6 Breve descripción de los otros capítulos de la memoria**

### **2.1 Análisis**

El primero apartado de este capítulo muestra el análisis de requisitos y el segundo, las historias de usuario obtenidas de este análisis. En el siguiente se expone una primera aproximación del diagrama de clases y de los diagramas que representarán la arquitectura de la aplicación. A continuación, se muestra una primera versión del diseño lógico de la base de datos. Por último, se indica las tecnologías utilizadas en el proyecto.

### **2.2 Diseño**

En el primer apartado se presentan los casos de uso que se han elaborado siguiendo los requisitos y las historias de usuario. En el segundo apartado se indican las correcciones que ha habido sobre el diagrama de datos, los diagramas de la arquitectura y el diseño lógico. En el siguiente se muestra el diseño físico de la base de datos. Los siguientes apartados muestran las pantallas y el código de la aplicación. El penúltimo apartado muestra un diagrama de secuencia para reflejar la arquitectura de la aplicación. Por último, se explican algunos detalles sobre la configuración de los entornos en local y real.

### **2.3 Valoraciones económicas**

Se muestra un análisis económico del proyecto.

### **2.4 Riesgos**

Se muestra un análisis sobre los riesgos a los que se ve expuesto el proyecto.

### **2.5 Mantenimiento**

Explicación de las tareas a seguir para un buen mantenimiento de una aplicación web.

### **2.6 Ampliaciones futuras**

Se exponen algunas funcionalidades interesantes que podrían implementarse en el futuro.

## 2. Resto de capítulos

### 2.1 Análisis

En la PAC 2, se han descrito las historias de usuario con la ayuda de los requisitos funcionales obtenidos en la fase de planificación. También se ha construido una primera aproximación del diagrama de clases y de los diagramas para representar la arquitectura de la aplicación que se refinarán en la fase de diseño (PAC 3). Por último, se ha realizado el diseño lógico de la base de datos, incluyendo las tablas, relaciones, claves, etc y como en el caso de los diagramas, se revisará para detectar errores e incorporar mejoras.

#### 2.1.1 Análisis de requisitos

##### Funcionales

##### i. Gestión de usuarios (sólo administrador)

1. Permitirá crear, ver, editar y eliminar un usuario en la base de datos.

##### ii. Gestión de ingredientes y recetas (sólo administrador)

1. Permitirá crear, ver, editar y eliminar ingredientes y recetas en la base de datos.

##### iii. Buscador de recetas

1. El usuario podrá realizar una búsqueda de recetas mediante filtros como el tiempo de preparación, la categoría: carnes; pescados y mariscos; legumbres, hortalizas y verduras; pastas; arroces y ensaladas. También se podrá filtrar por la información nutricional indicando el valor máximo o mínimo para: grasas, hidratos de carbono, proteínas y sal. Una vez añadidas las preferencias, la aplicación mostrará una lista de recetas acorde con los criterios introducidos.

1. Se podrá acceder a la ficha de cada receta. La ficha nos mostrará una imagen de la receta, los ingredientes para dos comensales (con opción a ser modificada para 1, 2, 3 y 4), el tiempo de preparación, la elaboración y la información nutricional.

##### iv. Planificador de recetas

1. Se podrá añadir el intervalo de tiempo que será como máximo de dos semanas. Podremos escoger el tipo de alimentación: alimentación variada, alimentación variada y saludable y alimentación proteica. Una vez aceptemos nos mostrará una lista de recetas según el criterio escogido, dos recetas por día (la cena contendrá menos grasas y aporte energético). Un usuario sólo tendrá en memoria un planificador y se podrá eliminar y generar otro en cualquier momento.

#### **v. Dietas para adelgazar/deporte**

1. El usuario podrá crear una dieta escogiendo entre dieta de adelgazamiento o dieta de deportista. Añadiendo los valores siguientes: peso, edad, sexo, altura y el tipo de dieta, la aplicación mostrará un informe de la dieta a seguir. Un usuario sólo tendrá en memoria una dieta y se podrá eliminarla y generar otra en cualquier momento.
2. También mostrará una lista de recetas relacionadas con su dieta.

#### **No funcionales**

##### **i. Requisitos de usabilidad**

1. La aplicación tendrá interfaces gráficas bien creadas. - Para garantizar una visualización adecuada en diferentes dispositivos la aplicación tendrá un diseño Responsive. - Mostrará mensajes de error para orientar e informar al usuario en sus acciones con el sistema.

##### **ii. Requisitos de mantenimiento**

1. Los usuarios podrán enviar consultas, sugerencias y avisos de error y el administrador las atenderá.
- 2.

##### **iii. Requisitos de rendimiento**

1. La aplicación deberá soportar cincuenta usuarios simultáneamente. - La aplicación debe estar preparada para almacenar un máximo de 10Gb de datos: información personal de usuarios, dietas guardadas, planificaciones, recetas, etc.

##### **iv. Requisitos de seguridad**

1. Los datos de acceso de los usuarios (usuario y contraseña) estarán cifrados mediante el método sha1.
2. Un administrador será el único que tenga acceso a los datos de los usuarios registrados.

##### **v. Requisitos legales**

1. La aplicación deberá cumplir la ley de protección de datos de carácter personal (LOPD).

### **2.1.2 Historias de usuario**

Para la documentación de requisitos se ha optado por las historias de usuario por ser el método más rápido que permite tener una aproximación de lo que se requiere. En este punto, el proyecto está en una fase muy temprana y he considerado que no era aconsejable consumir mucho tiempo y recursos en documentarla con más detalle. En la fase de diseño, se ha utilizado la técnica de los casos de uso para ofrecer un grado más alto de detalle y formalismo que las historias de usuario no permiten.

Estas son las historias de usuario de los dos tipos de actores que intervienen en la aplicación Recipe Repo.

#### **Administrador**

##### Gestión de usuarios

Como administrador del sistema quiero ver un listado con todos los usuarios, consultar y editar su ficha de usuario, registrar un usuario nuevo y eliminarlo.

##### Gestión de recetas

Como administrador del sistema repo quiero ver un listado con todas las recetas, consultar y editar las fichas de recetas, eliminar y registrar una receta nueva.

##### Gestión de ingredientes

Como administrador del sistema quiero ver un listado con todos los ingredientes disponibles, consultar y editar las fichas de ingredientes, registrar un ingrediente nuevo y poder eliminarlos.

## **Usuario**

### Búsqueda de recetas

Como usuario quiero consultar un listado de recetas:

- Quiero que el sistema me muestre por defecto una lista con todas las recetas disponibles.
- Quiero poder realizar una búsqueda por categoría, tiempo de cocción, por número de calorías, grasas, proteínas, sal, o periodo del día (almuerzo, comida y cena).
- Quiero poder acceder a la ficha de una receta del listado para obtener más información.

### Planificación

Como usuario quiero consultar mi planificación creada por mí.

- Quiero que el sistema muestre un listado de días (máximo 14) con tres recetas por día según la configuración escogida: tipo de nutrición y número de días.
- Quiero poder acceder a las fichas de las recetas del listado para obtener más información sobre la receta.

-

Como usuario quiero realizar cambios sobre mi planificación.

Como usuario quiero poder eliminarla.

Como usuario quiero crear una nueva planificación.

### Dieta

Como usuario quiero consultar mi dieta creada por mí.

- Quiero que el sistema me calcule las kcalorias que debo consumir después de introducir los datos siguientes: sexo, edad, altura, peso y tipo de actividad física.
- Quiero que el sistema muestre un listado de siete días con tres recetas cada uno.
- Quiero poder acceder a las fichas de las recetas del listado para obtener más información sobre la receta.

-

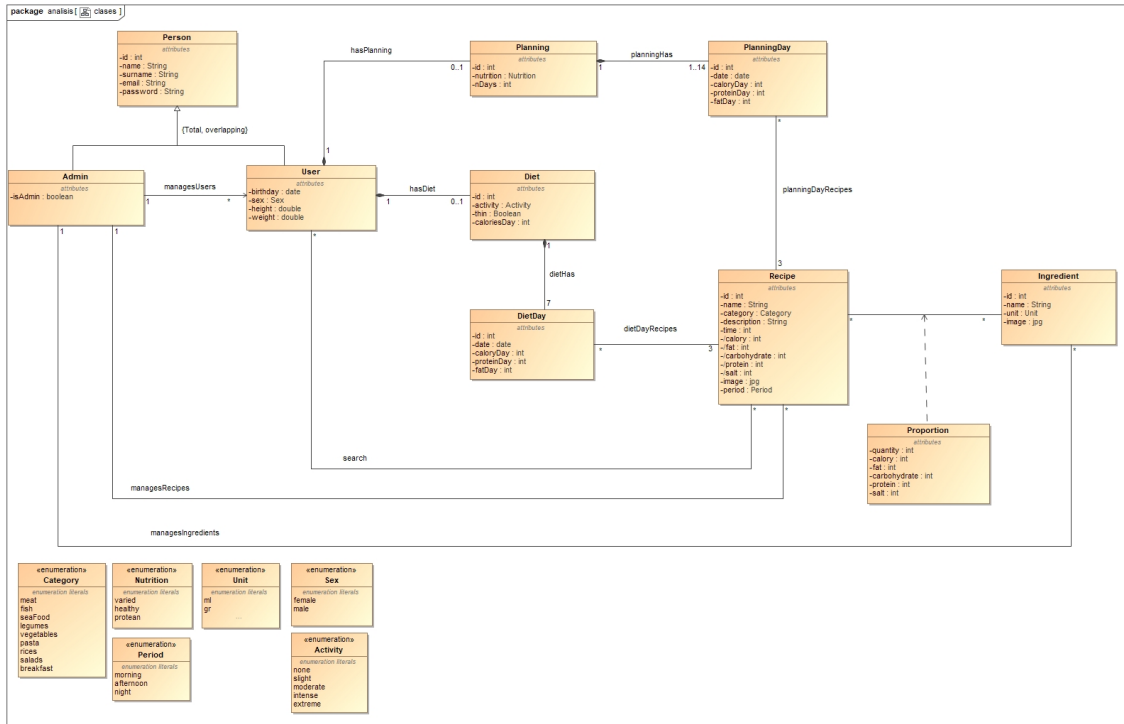
Como usuario quiero realizar cambios sobre mi dieta.

Como usuario quiero poder eliminarla.

Como usuario quiero crear una nueva dieta.

### 2.1.3 Diagrama del modelo de datos

Se ha utilizado el lenguaje unificado de modelado (UML) para representar el modelo de datos. Las razones principales por las que se ha escogido esta alternativa son por su facilidad de comprensión y que permite representar muchos de los conceptos más importantes en el desarrollo de aplicaciones orientadas a objetos. Esta es la primera versión del diagrama del modelo de datos



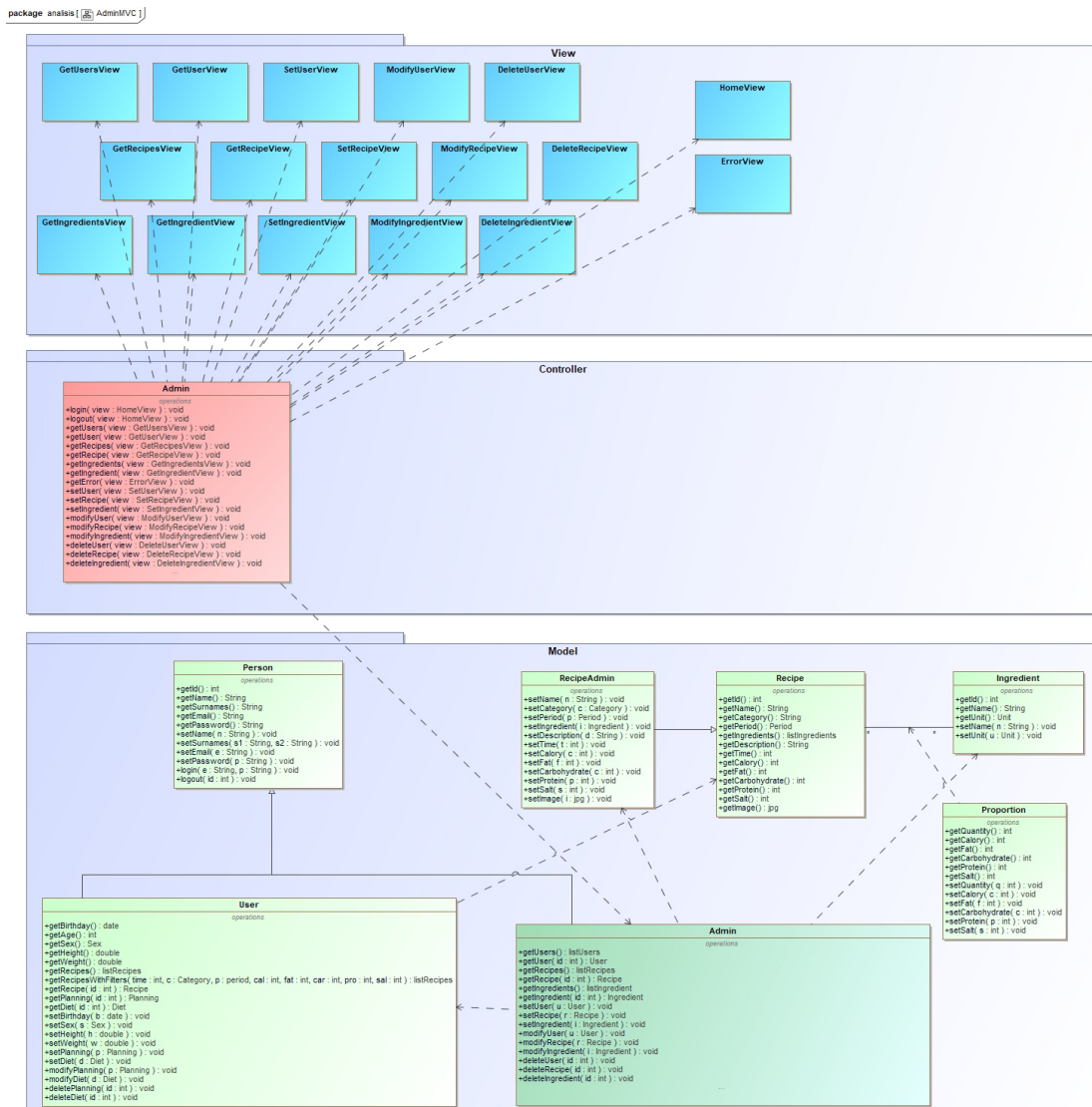
Nombre	Descripción	Clave	Relaciones
Person	Información personal de usuarios	id	
Admin	Usuarios administradores de la aplicación	person	User, Recipe e Ingredient
User	Usuarios de la aplicación	person	Recipe, Planning y Diet
Planning	Guarda información de las planificaciones de los usuarios	user + id	PlanningDay
PlanningDay	Mantiene la información sobre un día de una planificación	planning + id	Recipe
Diet	Guarda información de las dietas de los usuarios	user + id	DietDay
DietDay	Mantiene la información sobre un día de una dieta	diet + id	Recipe
Recipe	Guarda la información de las recetas	id	PlanningDay, DietDay, Ingredient y Proportion
Ingredient	Guarda la información de los ingredientes	id	Recipe y Proportion
Proportion	Mantiene la información de un ingrediente respecto a una receta en concreto.	recipe + ingredient	Recipe e Ingredient



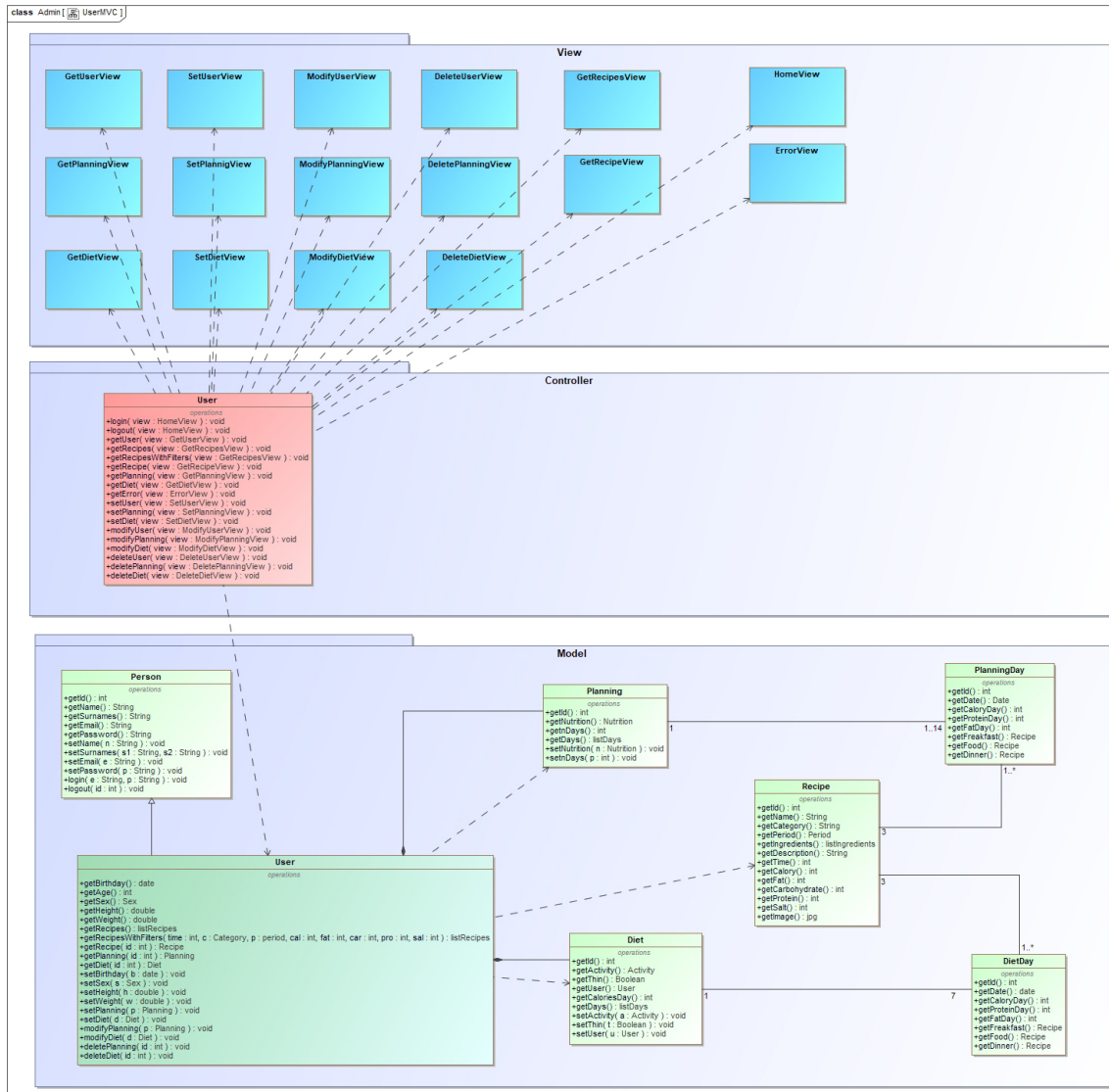
## 2.1.4 Diagramas de la arquitectura de la aplicación

También se ha utilizado UML para representar la arquitectura de la aplicación por el mismo motivo que con el diagrama de datos. El diseño de la aplicación utiliza el patrón arquitectónico modelo vista controlador (MVC) para permitir la reutilización de código y mejorar la incorporación de nuevas funcionalidades y su mantenimiento. A continuación, se muestra una primera versión del diagrama de la arquitectura de la aplicación.

### Controlador: AdminController



# Controlador: UserController



## 2.1.5 Diseño lógico de la base de datos

En esta fase se ha comenzado con el diseño lógico de la base de datos elaborando una primera versión. Esta versión parte del diagrama del modelo del dominio inicial y muestra la transformación del diagrama estático en elementos del modelo relacional que más tarde se utilizarán para el diseño físico de la base de datos. A continuación, se muestra la primera versión del diseño lógico de la base de datos

Notación utilizada:

- Para indicar un atributo como clave primaria se subrayará.
- Para indicar un atributo como clave foránea se escribirá en rojo.

Tabla	
Admin	Admin( <u>id</u> , name, surname, email, password)
User	User( <u>id</u> , name, surname, email, password, <b>admin</b> )
Search	Search( <b>user</b> , <b>recipe</b> )
Planning	Planning( <u>id</u> , <b>user</b> , nutrition, period)
PlanningDay	PlanningDay( <u>id</u> , date, caloryDate, proteinDay, fatDay, <b>planning</b> )
planningDayRecipes	planningDayRecipes( <b>planningDay</b> , <b>recipe</b> )
Diet	Diet( <u>id</u> , <b>user</b> , activity, thin)
DietDay	DietDay( <u>id</u> , date, caloryDay, proteinDay, fatDay, <b>diet</b> )
dietDayRecipes	dietDayRecipes( <b>dietDay</b> , <b>recipe</b> )
Recipe	Recipe( <u>id</u> , name, category, description, time, calory, fat, carbohydrate, protein, salt, image, period, <b>admin</b> )
Proportion	Proportion( <b>recipe</b> , <b>ingredient</b> , quantity, calory, carbohydrate, protein, salt)
Ingredient	Ingredient( <u>id</u> , name, unit, <b>admin</b> )

### 2.1.6 Tecnologías utilizadas

Tecnologías	
Modelado de datos UML	Magic Draw
Servidor web entorno local	WAMP Server (servidor web apache)
Servidor web entorno real	Servidor web Apache
IDE	Netbeans
Base de datos en local y real	Mysql
Lenguaje de programación	PHP, JavaScript y JQuery
Lenguajes para la representación visual	HTML5 y CSS3

Una vez escogidas las tecnologías el siguiente paso será preparar el entorno de trabajo. La implementación y las pruebas de la aplicación se harán sobre un servidor web local. Se ha optado por utilizar el sistema de infraestructura de Internet WAMP Server ya que permite instalar un servidor web Apache con un servidor Mysql de forma simple.

Respecto al código, se ha optado por el IDE de NETbeans para programar con PHP ya que además de ser muy potente ofrece un gran número de módulos, es gratuito y libre sin restricciones. Para definir la presentación de las páginas se utilizará HTML junto con CSS en sus últimas versiones (HTML5 y CSS3). También será necesario utilizar la librería jQuery para el manejo de eventos, animaciones, etc.

Para el modelado de los diagramas se ha optado por la herramienta MagicDraw por ser una herramienta completa, fácil de usar y además por ser proporcionada por la UOC. Esta herramienta permite crear distintos diagramas como el de datos o diagramas para representar la arquitectura de la aplicación. Respecto al diagrama de datos, con esta herramienta se podrá representas las clases, añadir atributos, métodos, restricciones e indicar las relaciones entre clases.

La aplicación estará basada en la tecnología cliente/servidor ya que ofrece ventajas como mezclar y acoplar componentes, organización centralizada, multiplataforma, etc. También se utilizará el patrón modelo, vista y controlador (MVC) para separar las capas de presentación, lógica y acceso a datos. De este modo será mucho más fácil su mantenimiento y la incorporación de nuevas funciones en un futuro.

## 2.2 Diseño

En la PAC 3, se han documentado los requisitos a partir de las historias de usuario y los diagramas obtenidos en la fase análisis. También se ha llevado a cabo una serie de modificaciones que han afectado al diagrama de datos y a los diagramas de la arquitectura de la aplicación. Por último, se muestran unas capturas de pantalla para dar un enfoque visual de la aplicación.

### 2.2.1 Casos de uso

**Caso de uso:** gestionar usuarios, recetas e ingredientes

**Actor principal:** usuario administrador

**Ámbito:** administración (backend)

**Nivel:** general

#### **Escenario principal de éxito:**

1. El usuario pide gestionar los datos de la aplicación.
2. El sistema ofrece tres tipos de información de que dispone.
3. El usuario escoge uno.
4. El sistema muestra una lista de registros sobre la elección del usuario con las opciones ver, editar y eliminar. Además, ofrece crear un nuevo registro.
5. Opcionalmente, el usuario escoge otro tipo de registros y volvemos al paso 4.
6. El usuario va reiterando sobre los pasos 3, 4 y 5 hasta que considere oportuno.

#### **Escenarios alternativos:**

1. Crear un nuevo registro
  - a. El usuario indica el tipo de registro (usuario, receta o ingrediente)
  - b. El usuario indica que quiere crear un registro.
  - c. El sistema registra el nuevo registro.
2. Editar un registro
  - a. El usuario indica el tipo de registro (usuario, receta o ingrediente)
  - b. El usuario indica que quiere editar un registro.
  - c. El sistema registra los cambios.
3. Eliminar un registro
  - a. El usuario indica el tipo de registro (usuario, receta o ingrediente)
  - b. El usuario indica que quiere eliminar un registro.
  - c. El sistema elimina el registro.

**Caso de uso:** gestionar ingredientes para una receta

**Actor principal:** usuario administrador

**Ámbito:** administración (backend)

**Nivel:** usuario

**Escenario principal de éxito:**

1. El usuario pide gestionar las recetas.
2. El sistema ofrece un listado de recetas.
3. El usuario selecciona una.
4. El sistema muestra un listado de ingredientes asociados a la receta (en caso de que ya tuviera)
5. Opcionalmente, el usuario sale de la receta seleccionada y volvemos al paso 2.
6. El usuario va reiterando sobre los pasos 2, 3, 4 y 5 hasta que considere oportuno.

**Escenarios alternativos:**

1. Añadir un nuevo ingrediente
  - a. El usuario escoge un ingrediente y añade la cantidad en gramos.
  - b. El sistema asocia el ingrediente y la cantidad para esa receta.
2. Desvincular un ingrediente
  - a. El usuario indica que quiere desvincular un ingrediente
  - b. El sistema desvincular el ingrediente para esa receta.

**Caso de uso:** ver recetas

**Actor principal:** usuario

**Ámbito:** recetas

**Nivel:** usuario

**Escenario principal de éxito:**

1. El usuario pide hacer una búsqueda de recetas
2. El usuario introduce la categoría o el tiempo, el máximo de calorías, grasas, carbohidratos, proteínas o el tipo de comida (desayuno, comida y cena) o introduce valores en todas las opciones a la vez.
3. El sistema muestra un listado con las recetas que coinciden con los criterios de la búsqueda.
4. Opcionalmente, el usuario cambia el criterio de búsqueda y volvemos al paso 3.
5. Los pasos 2, 3 y 4 se van repitiendo hasta que el usuario considere oportuno.

**Escenarios alternativos:**

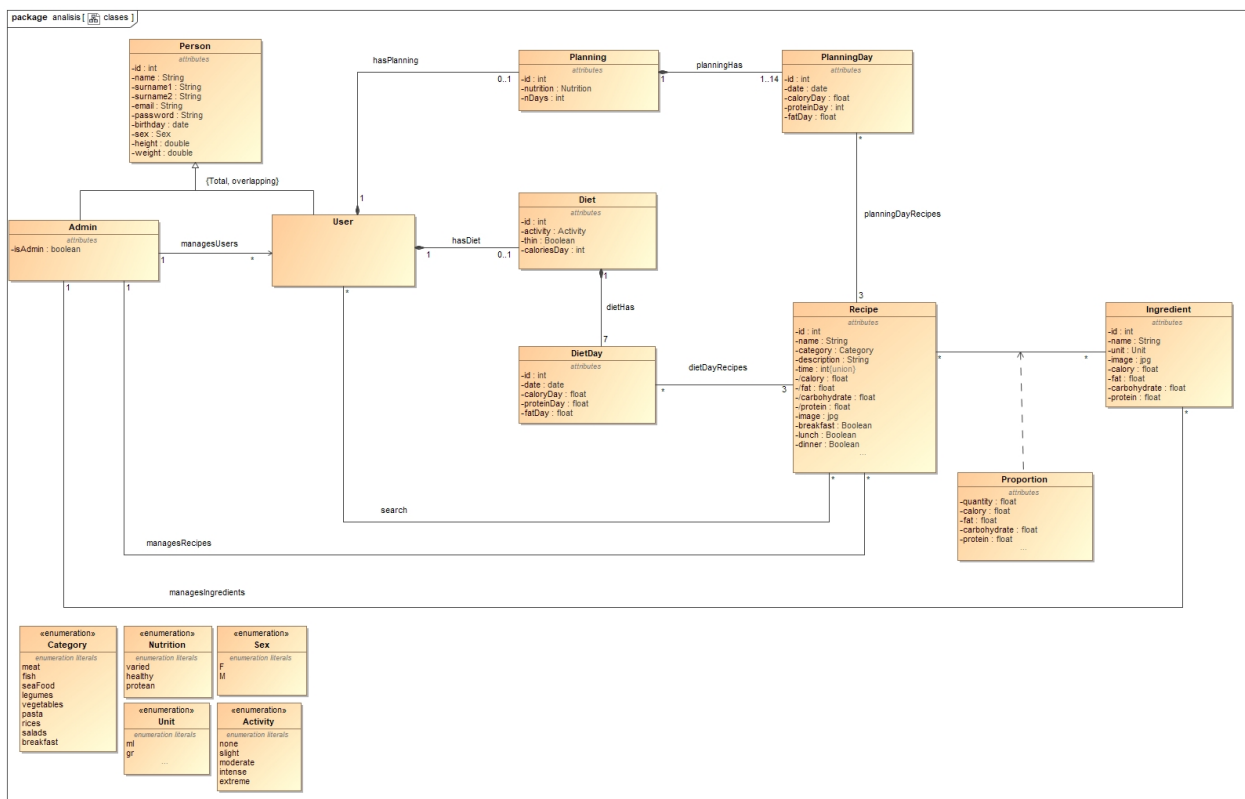
1. Ver receta
  - a. El usuario indica que quiere ver los detalles de una receta del listado mostrado por el sistema.
  - b. El sistema muestra los detalles de la receta: título, descripción, calorías, grasas, carbohidratos y proteínas calculadas para una persona, si es desayuno, comida o cena y una imagen.

## 2.2.2 Correcciones

En este apartado se muestran las correcciones y actualizaciones que han sido necesarias realizar sobre el diagrama estático, el diseño lógico y la arquitectura de la aplicación.

### 2.2.2.1 Diagrama de datos

1. Se ha añadido un campo (isAdmin) en la clase Admin para identificar al usuario administrador del resto de usuarios de la aplicación.
2. Se ha eliminado la enumeración Period y en su lugar, se ha añadido tres nuevos campos a la clase Recipe: breakfast, lunch y dinner. Esta modificación ha sido necesaria para mejorar el filtro de recetas. Ahora se puede hacer una búsqueda filtrando por estos campos.
3. La clase Ingredient ha sido modificada con los campos: calory, fat, carbohydrate y protein. Estos campos sólo indican el valor del ingrediente por gramo. Esta actualización ha sido necesaria para poder realizar los cálculos de los ingredientes de las recetas.
4. El campo salt se ha eliminado.



Nombre	Descripción	Clave	Relaciones
Person	Información personal de usuarios	id	
Admin	Usuarios administradores de la aplicación	person	User, Recipe e Ingredient
User	Usuarios de la aplicación	person	Recipe, Planning y Diet
Planning	Guarda información de las planificaciones de los usuarios	user + id	PlanningDay
PlanningDay	Mantiene la información sobre un día de una planificación	planning + id	Recipe
Diet	Guarda información de las dietas de los usuarios	user + id	DietDay
DietDay	Mantiene la información sobre un día de una dieta	diet + id	Recipe
Recipe	Guarda la información de las recetas	id	PlanningDay, DietDay, Ingredient y Proportion
Ingredient	Guarda la información de los ingredientes	id	Recipe y Proportion
Proportion	Mantiene la información de un ingrediente respecto a una receta en concreto.	recipe + ingredient	Recipe e Ingredient

### 2.2.2.2 Diseño lógico

Notación utilizada:

- Para indicar un atributo como clave primaria se subrayará.
- Para indicar un atributo como clave foránea se escribirá en **rojo**.

Tabla	
User	User( <u>id</u> , name, surname1, surname2, email, password, birthday, sex, height, weight, isadmin)
Planning	Planning( <u>id</u> , <b>user</b> , nutrition, nDays)
PlanningDay	PlanningDay( <u>id</u> , date, caloryDate, proteinDay, fatDay, <b>planning</b> )
planningDayRecipes	PlanningDayRecipes( <b>planningDay</b> , <b>recipe</b> )
Diet	Diet( <u>id</u> , <b>user</b> , activity, thin, caloriesDay)
DietDay	DietDay( <u>id</u> , date, caloryDay, proteinDay, fatDay, <b>diet</b> )
dietDayRecipes	DietDayRecipes( <b>dietDay</b> , <b>recipe</b> )
Recipe	Recipe( <u>id</u> , name, category, description, time, calory, fat, carbohydrate, protein, image, breakfast, lunch, dinner)
Proportion	Proportion( <b>recipe</b> , <b>ingredient</b> , quantity, calory, carbohydrate, protein)
Ingredient	Ingredient( <u>id</u> , name, unit, calory, carbohydrate, protein)



### **2.2.2.3 Arquitectura de la aplicación**

La arquitectura escogida es una arquitectura web formada por tres capas. La primera es la capa de presentación, esta se encarga de obtener los datos de los usuarios para enviarlos al servidor. La segunda capa es la de proceso donde se realizan las operaciones sobre estos datos y se muestran en la capa de presentación. Por último, la capa de datos realiza las operaciones para almacenarlos, actualizarlos y recuperarlos de la base de datos.

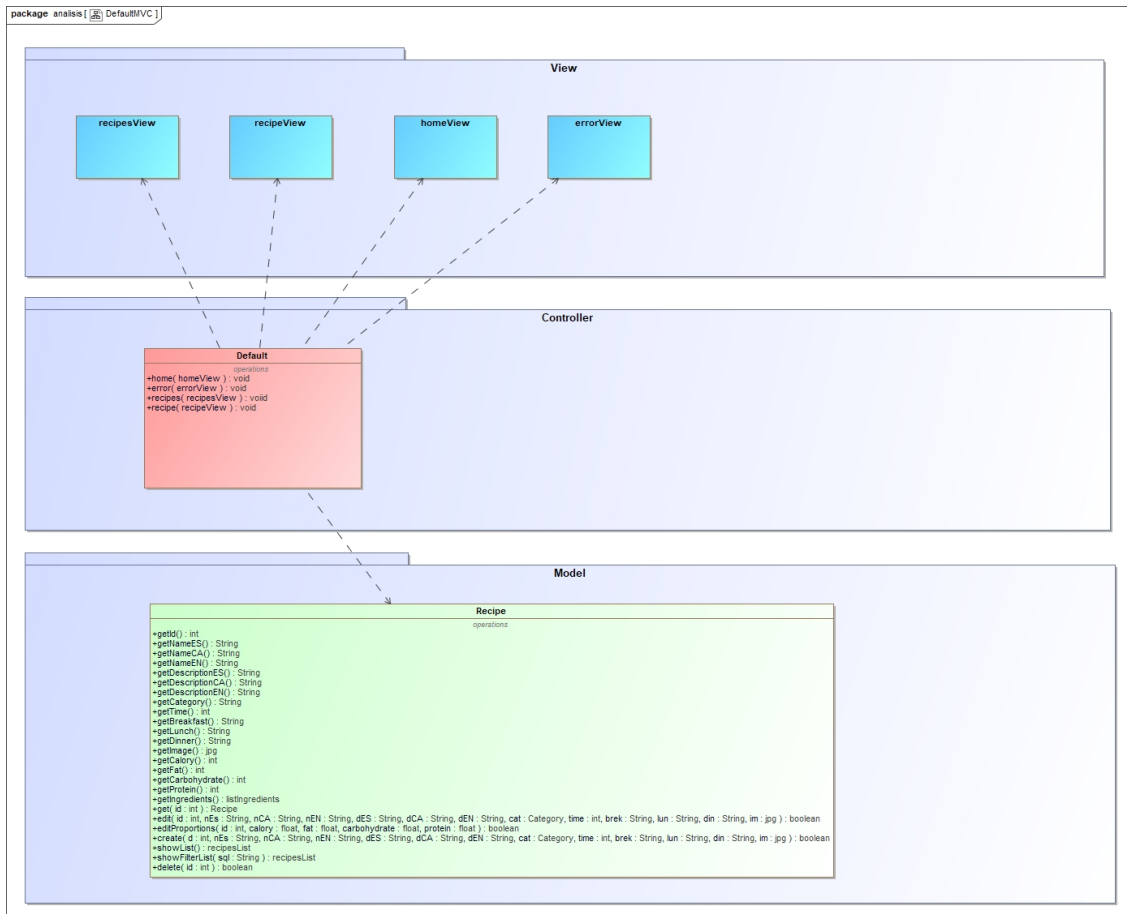
Para el desarrollo de la aplicación se ha optado por aplicar el patrón arquitectónico modelo, vista y controlador (MVC). Respecto a la versión inicial del patrón, ha surgido la necesidad de incorporar un nuevo controlador para separar conceptos que no se tuvieron en cuenta en la fase del análisis.

Este nuevo controlador llamado `DefaultController`, se encargará de aspectos básicos como: inicializar la aplicación y gestionar las funcionalidades que permite la aplicación por defecto (funciones que no son de administración ni de usuario registrado).

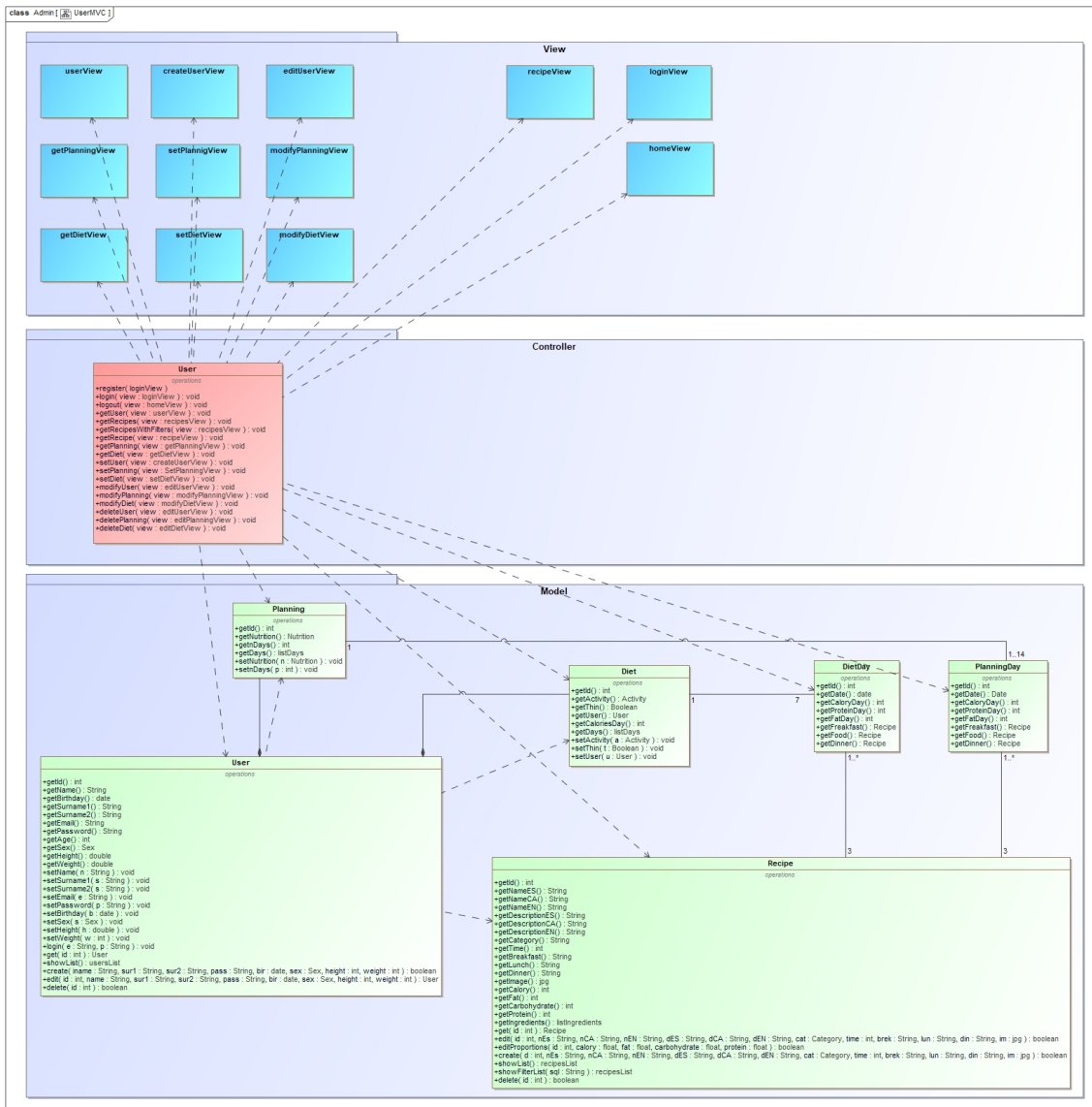
La aplicación es iniciada por `DefaultController` ya que es el primero en llamar al controlador frontal encargado de seleccionar el controlador y cargar los ficheros de configuración. Por otro lado, también se encarga de las funciones que son accesibles a cualquier usuario, esté o no registrado como por ejemplo la función filtrar recetas.

A continuación, se muestran los diagramas `DefaultController`, `AdminController` y `UserController` con las modificaciones mencionadas.

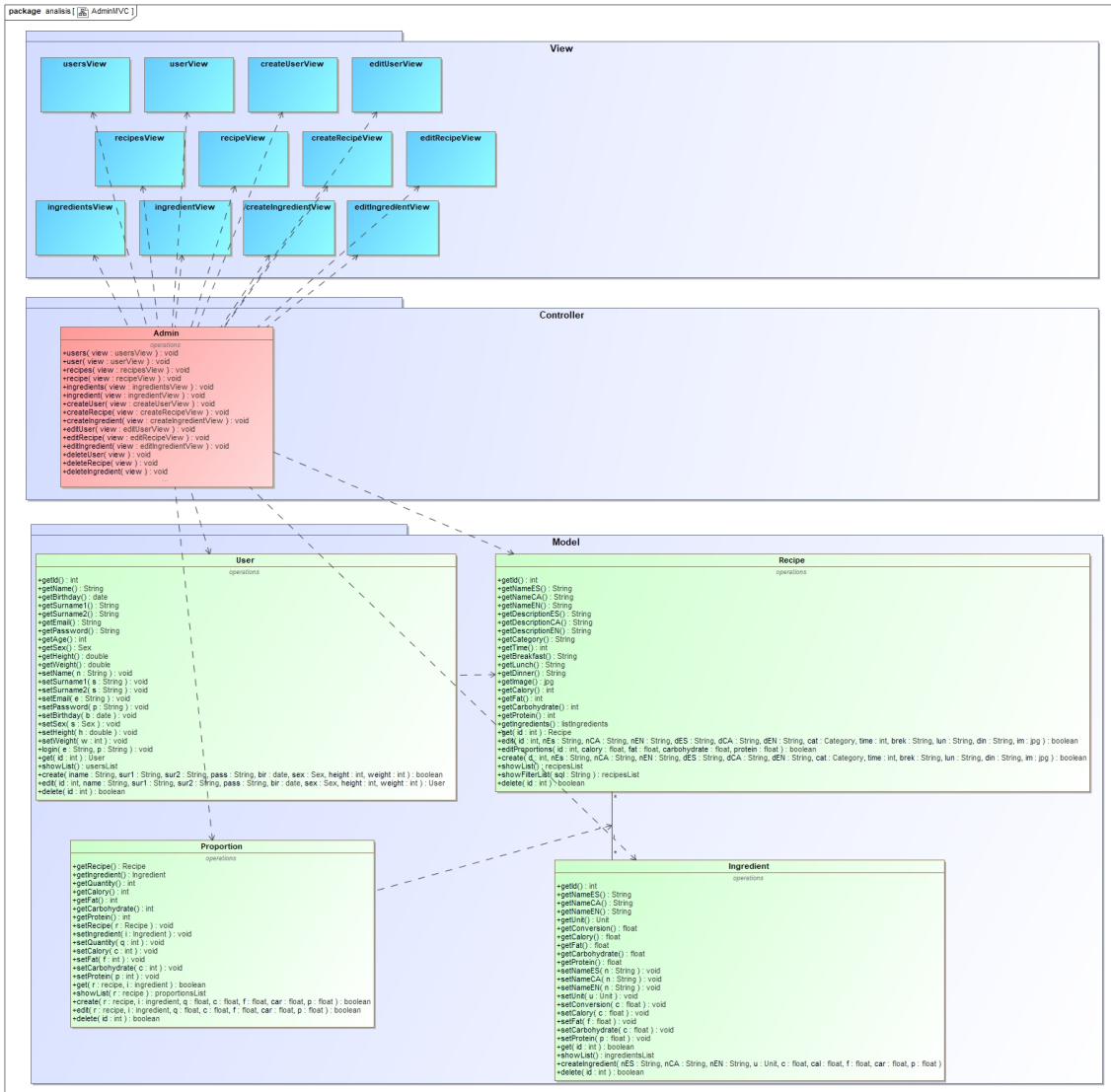
# DefaultController



# UserController

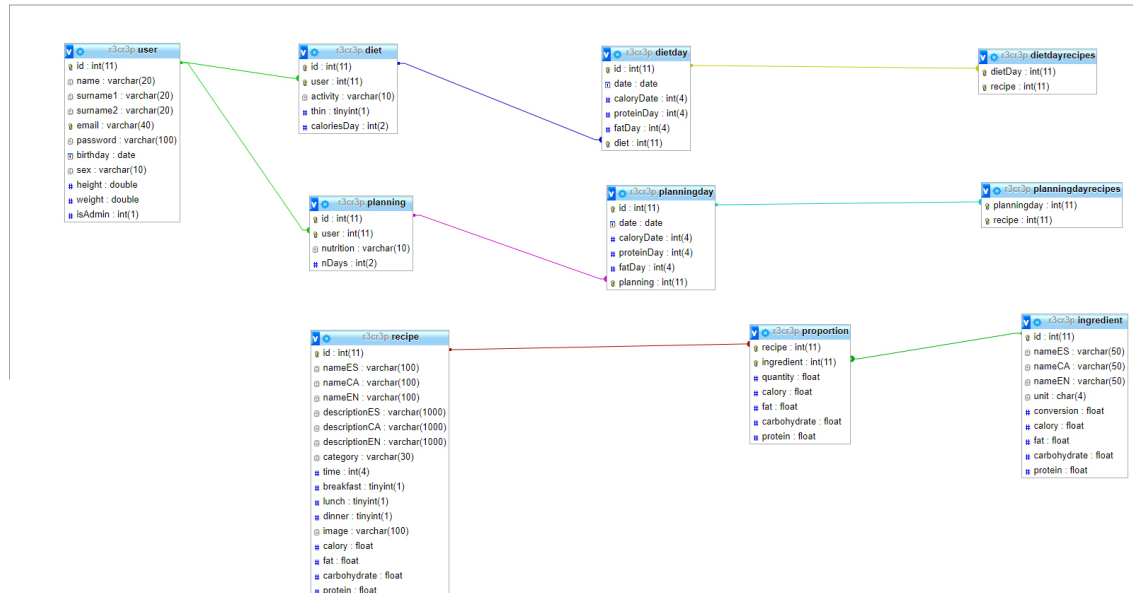


# AdminController



## 2.2.3 Diseño físico de la base de datos

Este es el diagrama de la base de datos relacional que se ha obtenido del diseño lógico de la base de datos.

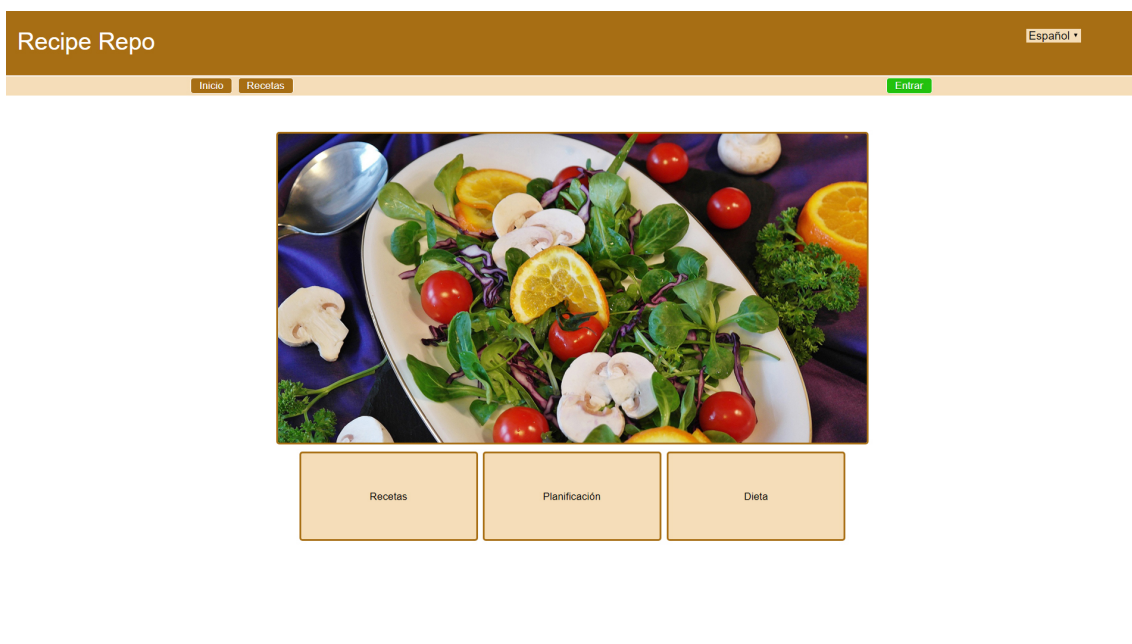


### 2.2.4 Diseño de pantallas

En primer lugar, se muestran algunas de las pantallas que forman el área de administración (backend) de Recipe repo: administración, ver, editar, crear y eliminar. Por último, se muestran las pantallas para la funcionalidad ver recetas y obtener detalle de una receta que es accesible para cualquier usuario este o no registrado (*frontend*).

### Home

Es la pantalla inicial, desde aquí se puede ir a la pantalla de inicio de sesión, registro y a la pantalla de listado de recetas.



## Iniciar sesión

En esta pantalla el usuario podrá acceder añadiendo su email y contraseña. También se podrá registrar

### Inicio de sesión

Recipe Repo Español ▾

[Inicio](#) [Recetas](#) [Entrar](#)

Email:

Contraseña:

[Entrar](#)

[Nuevo usuario](#)

## Registro

Recipe Repo Español ▾

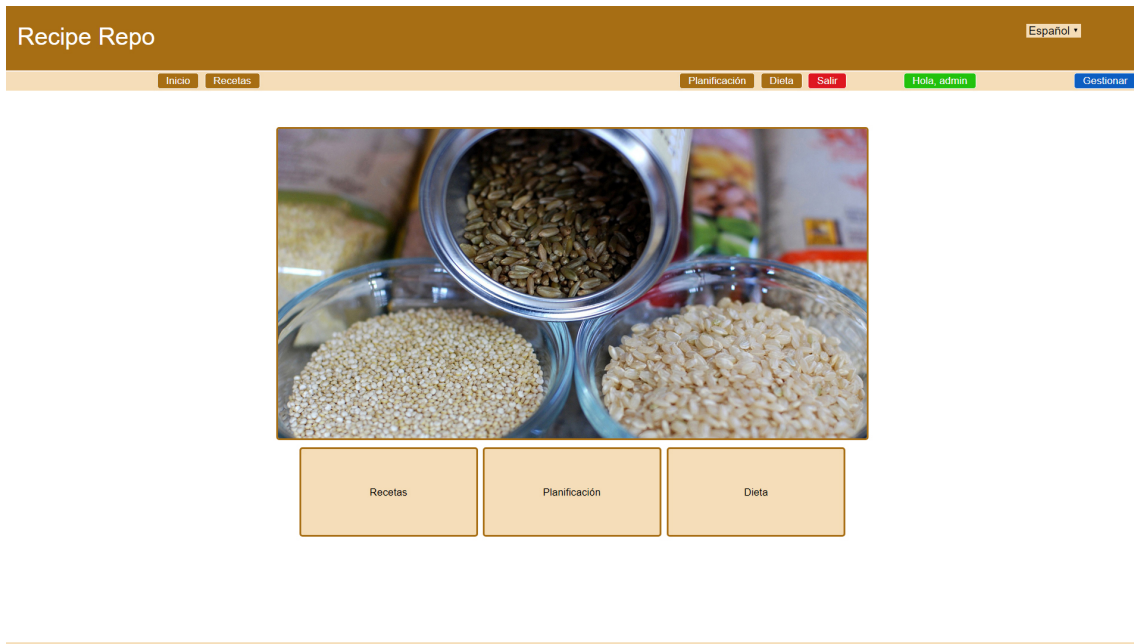
[Inicio](#) [Recetas](#) [Entrar](#)

Nombre:	Apellido 1:	Apellido 2:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Email:	Repite email:	Contraseña:
<input type="text" value="admin@reciperepo.com"/>	<input type="text"/>	<input type="password" value="*****"/>
Repite contraseña:	Cumpleaños:	Sexo:
<input type="text"/>	<input type="text" value="dd/mm/aaaa"/>	<input type="text"/>
Altura (cm):	Peso:(kg)	<a href="#">Registro</a>
<input type="text"/>	<input type="text"/>	

[Entrar](#)

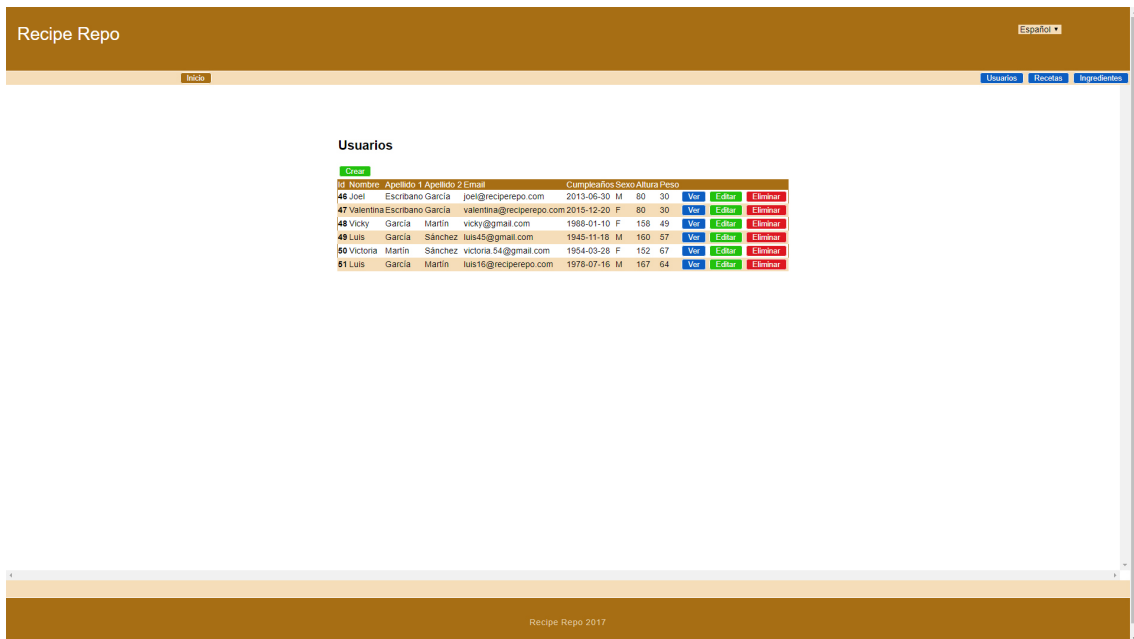
## Sesión iniciada

Una vez iniciada la sesión y dependiendo si el usuario es administrador o usuario normal tendrá acceso a unas determinadas funciones.



## Administración

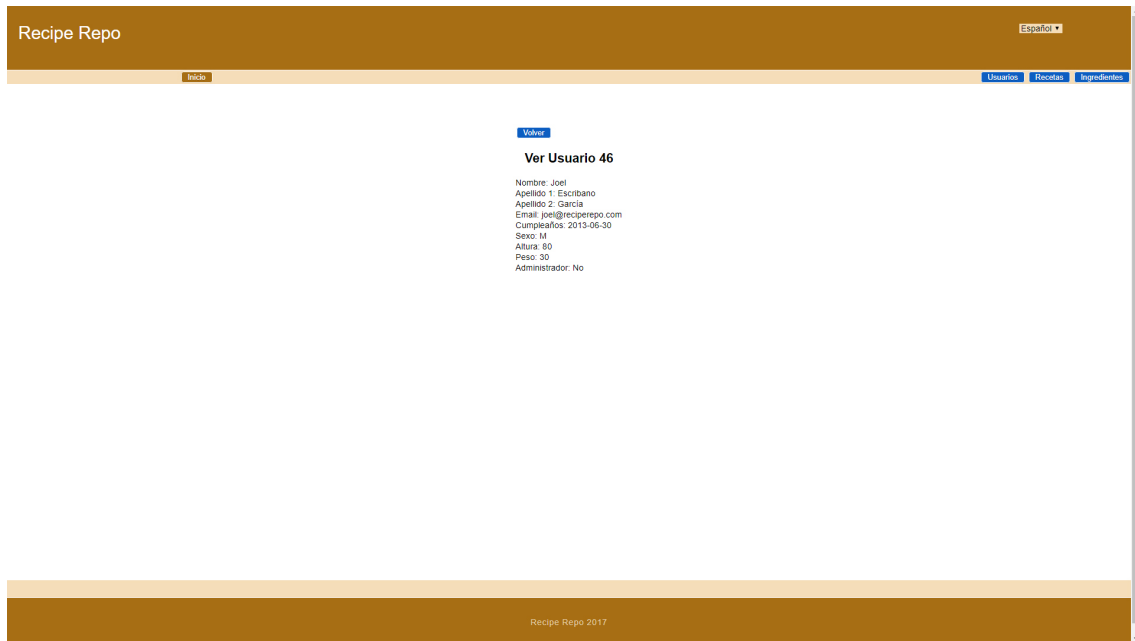
El usuario administrador al iniciar sesión en la aplicación se le mostrará un botón que le llevará al área de gestión. Podrá acceder a gestionar los usuarios, las recetas y los ingredientes mediante los botones que se muestran en azul (arriba a la derecha).





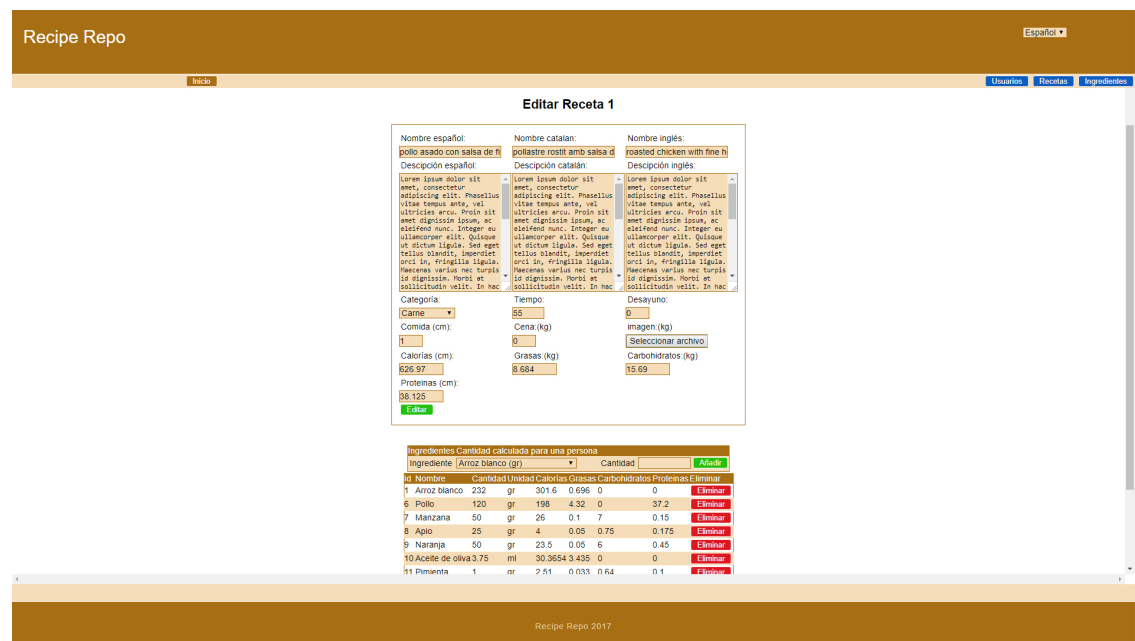
## Ver

Cada registro de usuarios, recetas e ingredientes dispone de tres botones: ver, editar y eliminar registro. Esta pantalla muestra la ficha de un registro de usuario después de haber pulsado sobre el botón ver.



## Editar

Esta pantalla muestra la ficha de un registro de receta después de haber pulsado sobre el botón editar. La pantalla editar receta es un caso especial de edición, ya que permite editar sobre los campos del registro y además permite asociar los ingredientes que formaran la receta.



## Crear

Esta pantalla permite crear un nuevo registro. Las tres secciones del área de administración (usuarios, recetas e ingredientes) disponen de esta función.

The screenshot shows the 'Crear Ingrediente' form in the Recipe Repo application. The form is titled 'Crear Ingrediente' and has a 'Volver' button at the top left. It contains several input fields for creating a new ingredient record:

- Nombre español:
- Nombre catalán:
- Nombre inglés:
- Unidad:
- Conversión gr:
- Calorías:
- Grasas:
- Carbohidratos:
- Proteínas:

A green 'Crear' button is located at the bottom left of the form. The application header shows 'Recipe Repo' and 'Español' as the selected language. Navigation links for 'Inicio', 'Usuarios', 'Recetas', and 'Ingredientes' are visible at the top right. The footer of the application displays 'Recipe Repo 2017'.

## Eliminar

Desde la pantalla principal de cada área se podrá eliminar un registro. Al eliminar un registro aparecerá un mensaje de advertencia preguntando si realmente queremos eliminar el registro.

The screenshot shows the 'Ingredientes' list in the Recipe Repo application. A confirmation dialog box is displayed over the list, asking 'reciperepo.net dice: Eliminar ingrediente' with 'Aceptar' and 'Cancelar' buttons. The list below contains the following data:

#	Nombre	Unidad	Conversión	gr	Calorías	Grasas	Carbohidratos	Proteínas			
1	Arroz blanco	gr	1	1.3	0.003	0.28	0.027		Ver	Editar	Eliminar
2	Leche	ml	1.03	0.4285	0.0103	0.0515	0.0302		Ver	Editar	Eliminar
4	Harina	ml	0.57	2.0748	0.0057	0.4332	0.057		Ver	Editar	Eliminar
5	Mantequilla	ml	0.911	6.53187	0.73791	0.000911	0.008199		Ver	Editar	Eliminar
6	Pollo	gr	1	2.39	0.125	0	0.246		Ver	Editar	Eliminar
7	Manzana	gr	1	0.52	0.002	0.14	0.003		Ver	Editar	Eliminar
8	Apio	gr	1	0.16	0.002	0.03	0.007		Ver	Editar	Eliminar
9	Naranja	gr	1	0.47	0.001	0.12	0.009		Ver	Editar	Eliminar
10	Acete de oliva	ml	0.916	8.09744	0.916	0	0		Ver	Editar	Eliminar
11	Pimienta	gr	1	2.51	0.033	0.64	0.1		Ver	Editar	Eliminar
12	Finas hierbas	gr	1	0	0	0	0		Ver	Editar	Eliminar
13	Sal	gr	1	0	0	0	0		Ver	Editar	Eliminar
14	Vino blanco	ml	1	0.82	0	0.026	0.001		Ver	Editar	Eliminar
15	Bombó	gr	1	1.38	0.06	0	0.21		Ver	Editar	Eliminar
16	Cebolla	gr	1	0.4	0.001	0.09	0.011		Ver	Editar	Eliminar
17	Pimiento verde	gr	1	0.2	0.01	0.02	0.01		Ver	Editar	Eliminar
18	Ajo	gr	1	1.2	0.002	0.244	0.043		Ver	Editar	Eliminar
19	Patatas fritas	gr	1	2.34	0.11	0.34	0.04		Ver	Editar	Eliminar
20	Patatas al horno	gr	1	1.62	0.04	0.3	0.03		Ver	Editar	Eliminar
21	Patatas cocidas	gr	1	0.77	0.002	0.16	0.017		Ver	Editar	Eliminar
22	Laurel	gr	1	0	0	0	0		Ver	Editar	Eliminar
23	Spingl	gr	1	0.36	0.008	0.06	0.03		Ver	Editar	Eliminar
24	Lentijas	gr	1	1.16	0.004	0.2	0.09		Ver	Editar	Eliminar
25	Chorizo	gr	1	4.55	0.38	0.019	0.24		Ver	Editar	Eliminar
26	Zanahoria	gr	1	0.394	0.002	0.069	0.0125		Ver	Editar	Eliminar
27	Coliflor	gr	1	0.25	0.003	0.05	0.019		Ver	Editar	Eliminar
28	Jamón curado	gr	1	2	0.13	0.004	0.2		Ver	Editar	Eliminar
29	Huevo	gr	1	1.55	0.11	0.011	0.13		Ver	Editar	Eliminar

The application header shows 'Recipe Repo' and 'Español' as the selected language. Navigation links for 'Inicio', 'Usuarios', 'Recetas', and 'Ingredientes' are visible at the top right. The footer of the application displays 'Recipe Repo 2017'.

## Ver recetas

Esta pantalla es accesible para cualquier usuario, aunque no esté registrado. Podrá acceder a todas las recetas que ofrece Recipe repo y obtener una lista de recetas filtrada por los parámetros siguientes: categoría; tiempo; máximo de calorías; máximo de grasas por gramo; máximo de carbohidratos por gramo; máximo de proteínas por gramo y tipo de comida; desayuno, comida y cena.

Recetas

Filtros:

- Categoría: Legumbres
- Grasas(igr. Max.): 40
- Carbohidratos(igr. Max.): 40
- Proteínas(igr. Max.): 50
- Tiempo(min. Max.): 40
- Calorías(cal. Max.): 50
- Desayuno:  Sí  No
- Comida:  Sí  No
- Cena:  Sí  No

Nombre	Categoría	Tiempo	Desayuno	Comida	Cena	Calorías	Grasas	Carbohidratos	Proteínas
pollo asado con salsa de finas hierbas y arroz	Pasta	55	No	Sí	No	626.97	8.684	15.69	38.125
Patatas con bonito	Ensaladas	30	No	Sí	Sí	446.622	20.703	38.17	25.5875
Lentejas con arroz	Postres	60	No	Sí	No	544.07	35.4687	26.6857	27.9656
Coliflor con huevo y patatas	Verduras	40	No	Sí	Sí	337.358	13.4055	37.4462	20.405
Salsado de setas y langostinos	Verduras	15	No	Sí	Sí	422.291	16.178	11.215	49.4375
Espagueti salsa aurosa	Pasta	25	No	Sí	No	375.132	18.6516	32.6061	18.8525
Arroz con salchicas y pimientos	Arroz	30	No	Sí	Sí	255.625	12.5555	27.2025	7.8875
Ensalada reina	Ensaladas	30	No	Sí	Sí	536.977	17.5155	50.6013	48.1875

## Ver detalle de receta

Cada receta de la lista anterior dispone de un botón que permite ver la ficha de la receta: título, descripción, imagen y una tabla con los valores nutricionales calculado para una persona.

Detalle de receta: pollo asado con salsa de finas hierbas y naranja

Descripción: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae tempus ante, vel ultricies arcu. Proin sit amet dignissim ipsum, ac eleifend nunc. Integer eu ullamcorper elit. Quisque ut dictum ligula. Sed eget tellus blandit, imperdiet orci in, fringilla ligula. Maecenas varius nec turpis id dignissim. Morbi at sollicitudin velit. In hac habitasse platea dictumst. Sed sit amet risus eget erat eleifend aliquet. Etiam tellus arcu, tincidunt at arcu at, feugiat volutpat mauris. Phasellus nec metus sit amet leo ornare posuere in at mi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean tincidunt iaculis erat in posuere. Phasellus faucibus condimentum pharetra. Aenean vel justo pulvinar enim congue elementum eu sit amet nibh. Nam vulputate ligula sit amet laoreet sodales.

Categoría: Carne | Tiempo: 55 min | Desayuno: No | Comida: No | Cena: Sí

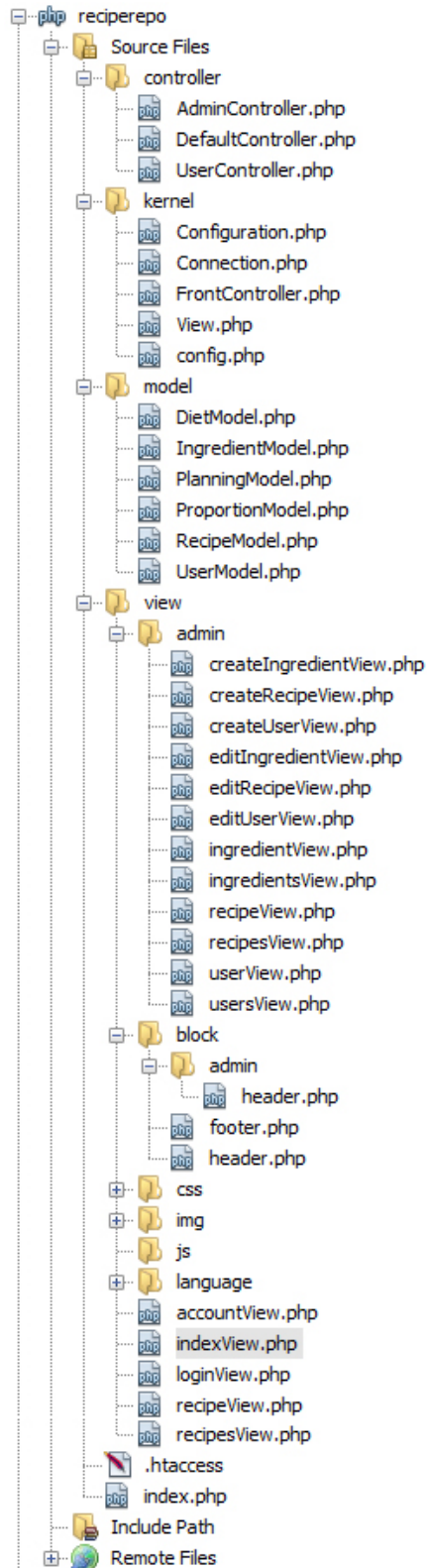
Imagen:

Cantidad calculada para una persona							
Id	Nombre	Cantidad	Unidad	Calorías	Grasas	Carbohidratos	Proteínas
1	Arroz blanco	232	gr	301.6	0.696	0	0
6	Pollo	120	gr	198	4.32	0	37.2
7	Manzana	50	gr	26	0.1	7	0.15
8	Apio	25	gr	4	0.05	0.75	0.175
9	Naranja	50	gr	23.5	0.05	6	0.45
10	Aceite de oliva	3.75	ml	30.3654	3.435	0	0
11	Pimienta	1	gr	2.51	0.033	0.64	0.1
12	Finas hierbas	1	gr	0	0	0	0
13	Sal	1	gr	0	0	0	0
14	Vino blanco	50	ml	41	0	1.3	0.05
<b>Total</b>				<b>626.97</b>	<b>8.684</b>	<b>15.69</b>	<b>38.125</b>

## 2.2.5 Código

En este apartado se muestra la estructura del proyecto y el código de algunas clases.

### Estructura del proyecto



## Archivos principales

### config.php

En este archivo están los datos para la conexión con la base de datos y constantes necesarias para el funcionamiento de los controladores, modelos y vistas.

```
8
9 $config = Configuration::singleton();
10
11 $config->set('controllersFolder', 'controller/');
12 $config->set('modelsFolder', 'model/');
13 $config->set('viewsFolder', 'view/');
14
15 $config->set('dbhost', 'localhost');
16 $config->set('dbname', 'r3cr3p');
17 $config->set('dbuser', 'ul70400617_r3c');
18 $config->set('dbpass', '4f645sawhjkfn43');
```

### Connection.php

Con esta clase se crea una única conexión a la base de datos que servirá para todas las consultas y actualizaciones que se realicen desde la aplicación. Para conseguir usar una única conexión se utiliza el patrón singleton.

```
14 class Connection extends PDO{
15
16     private static $instance = null;
17
18     public function __construct(){
19         $config = Configuration::singleton();
20         parent::__construct('mysql:host=' . $config->get('dbhost') . ';dbname=' . $config->get('dbname'),
21             $config->get('dbuser'), $config->get('dbpass'));
22     }
23
24     public static function singleton(){
25         if (self::$instance == null){
26             self::$instance = new self();
27         }
28         return self::$instance;
29     }
30 }
```

### View.php

Esta clase hará la función de motor de vistas, todas las vistas son gestionadas desde esta clase.

```
14 class View{
15
16     function __construct(){
17
18     }
19
20     public function show($view, $object = array()){
21         $config = Configuration::singleton();
22         $path = $config->get('viewsFolder') . $view . "View.php";
23         if (file_exists($path) == false){
24             trigger_error ('Template ' . $path . ' does not exist.', E_USER_NOTICE);
25             return false;
26         }
27         include($path);
28     }
29
30     public function show2($view, $object = array(), $object2 = array()){
31         $config = Configuration::singleton();
32         $path = $config->get('viewsFolder') . $view . "View.php";
33         if (file_exists($path) == false){
34             trigger_error ('Template ' . $path . ' does not exist.', E_USER_NOTICE);
35             return false;
36         }
37         include($path);
38     }
39
40     public function show3($view, $object = array(), $object2 = array(), $object3 = array()){
41         $config = Configuration::singleton();
42         $path = $config->get('viewsFolder') . $view . "View.php";
43         if (file_exists($path) == false){
44             trigger_error ('Template ' . $path . ' does not exist.', E_USER_NOTICE);
45             return false;
46         }
47         include($path);
48     }
49 }
```

## Frontcontroller.php

Frontcontroller es la clase encargada de gestionar las peticiones que vienen de las vistas. Se encarga de preparar el controlador y la acción necesarias para cada petición del usuario.

```
class FrontController {

    static function main()

        require 'Configuration.php';
        require 'Connection.php';
        require 'View.php';
        require 'XXXXXX-XXXX';

        //Comenzamos el nombre del Controlador o en su defecto, tomamos que es el DefaultController
        if (!isset($_POST['controller'])) {
            if (!empty($_GET['controller'])) {
                $controllerName = $_GET['controller'];
            }
            else {
                $controllerName = "DefaultController";
            }
        }
        else {
            if (!empty($_POST['controller'])) {
                $controllerName = $_POST['controller'];
            }
            else {
                $controllerName = "DefaultController";
            }
        }

        //Lo mismo sucede con las acciones, si no hay acción, tomamos home como acción
        if (!isset($_POST['action'])) {
            if (!empty($_GET['action'])) {
                $actionName = $_GET['action'];
            }
            else {
                $actionName = "home";
            }
        }
        $controllerPath = $config->get('controllersFolder').$controllerName.'.php';
    }
    else {
        if (!empty($_POST['action'])) {
            $actionName = $_POST['action'];
        }
        else {
            $actionName = "home";
        }
        $controllerPath = $config->get('controllersFolder').$controllerName.'.php';
    }

    //Incluimos el fichero que contiene nuestra clase controladora solicitada
    if (is_file($controllerPath)) {
        require $controllerPath;
    }
    else {
        die("THE CONTROLLER: ' . $controllerName . ' NOT FOUND - ERROR 404");
    }

    //Si no existe la clase que buscamos y en acción, mostramos un error 404
    if (is_callable(array($controllerName, $actionName)) == false) {
        //trigger_error($controllerName.' -> '.$actionName.' not found ', E_USER_NOTICE);
        die("THE ACTION: ' . $actionName . ' OF THE CONTROLLER: ' . $controllerName . ' NOT FOUND - ERROR 404");
        //return false;
    }

    if (!isset($_SESSION['controller'])) $_SESSION['controller'] = "DefaultController";
    if (!isset($_SESSION['language'])) $_SESSION['language'] = "es";
    if (!isset($_SESSION['page'])) $_SESSION['page'] = "index";
    if (!isset($_SESSION['login'])) $_SESSION['login'] = "visibility: visible";
    if (!isset($_SESSION['myAccount'])) $_SESSION['myAccount'] = "visibility: hidden";
    if (!isset($_SESSION['manage'])) $_SESSION['manage'] = "visibility: hidden";
    if (!isset($_SESSION['logout'])) $_SESSION['logout'] = "visibility: hidden";

    $controller = new $controllerName();
    $controller->$actionName();
}
}
```





# Modelo

Estas clases son las encargadas de realizar las consultas y actualizaciones a la base de datos. Las clases controladoras obtienen los datos de estas clases.

## UserModel

```
14 class userModel {
15     protected $db;
16     private $id;
17     private $name;
18     private $surname1;
19     private $surname2;
20     private $email;
21     private $password;
22     private $birthday;
23     private $sex;
24     private $weight;
25     private $height;
26     private $isAdmin;
27
28     public function __construct(){
29         $this->db = Connection::getSingleton();
30     }
31
32     function getDb() {
33         return $this->db;
34     }
35
36     function getId() {
37         return $this->id;
38     }
39
40     function getName() {
41         return $this->name;
42     }
43
44     function getSurname1() {
45         return $this->surname1;
46     }
47
48     function getSurname2() {
49         return $this->surname2;
50     }
51
52     function getEmail() {
53         return $this->email;
54     }
55
56     function getPassword() {
57         return $this->password;
58     }
59
60     function getBirthday() {
61         return $this->birthday;
62     }
63
64     function getSex() {
65         return $this->sex;
66     }
67
68     function getHeight() {
69         return $this->height;
70     }
71
72     function getWeight() {
73         return $this->weight;
74     }
75
76     function getIsAdmin() {
77         return $this->isAdmin;
78     }
79
80     function setId($id) {
81         $this->id = $id;
82     }
83
84     function setName($name) {
85         $this->name = $name;
86     }
87
88     function setSurname1($surname1) {
89         $this->surname1 = $surname1;
90     }
91
92     function setSurname2($surname2) {
93         $this->surname2 = $surname2;
94     }
95
96     function setEmail($email) {
97         $this->email = $email;
98     }
99
100    function setPassword($password) {
101        $this->password = $password;
102    }
103
104    function setBirthday($birthday) {
105        $this->birthday = $birthday;
106    }
107
108    function setSex($sex) {
109        $this->sex = $sex;
110    }
111
112    function setHeight($height) {
113        $this->height = $height;
114    }
115
116    function setWeight($weight) {
117        $this->weight = $weight;
118    }
119
120    function setIsAdmin($isAdmin) {
121        $this->isAdmin = $isAdmin;
122    }
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



## Vista

Las vistas se encargan de mostrar los datos que vienen de las clases controladoras con una estructura formada con HTML y CSS que puede ser visualizada desde cualquier navegador web.

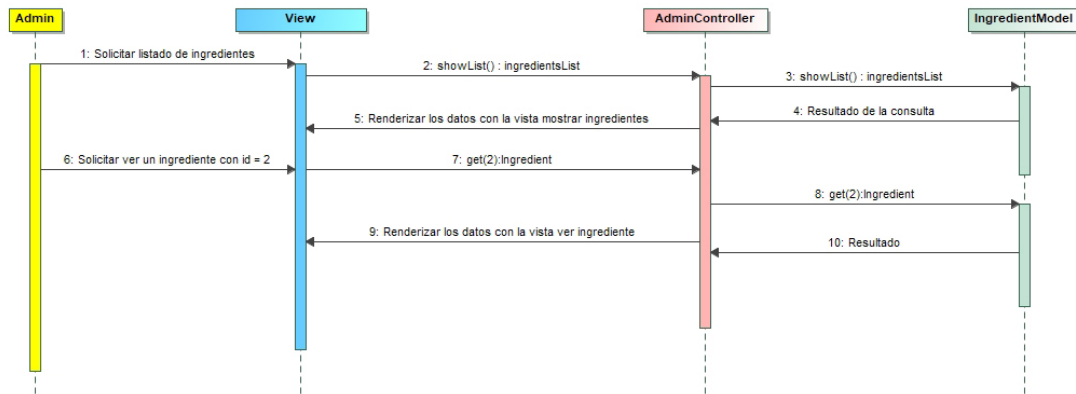
## UsersView

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>
5 To change this license header, choose License Headers in Project Properties.
6 To change this template file, choose Tools | Templates
7 and open the template in the editor.
8 -->
9 </title>
10 </head>
11 <body>
12 <div class="container" style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: 0 auto; text-align: center;">
13 <div class="form-group" style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: 0 auto; text-align: center;">
14 <input type="text" value="" style="width: 100%; border: none; border-bottom: 1px solid #ccc; margin-bottom: 5px;" />
15 <input type="button" value="Crear Usuario" class="btn btn-success" style="width: 100%; margin-top: 5px;" />
16 </div>
17 <div class="form-group" style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: 0 auto; text-align: center;">
18 <input type="text" value="" style="width: 100%; border: none; border-bottom: 1px solid #ccc; margin-bottom: 5px;" />
19 <input type="button" value="Actualizar Usuario" class="btn btn-success" style="width: 100%; margin-top: 5px;" />
20 </div>
21 </div>
22 <div class="table" style="width: 100%; border-collapse: collapse; margin-top: 20px;">
23 <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;">
24 <thead>
25 <tr>
26 <th style="width: 10%;>ID</th>
27 <th style="width: 30%;>Nombre</th>
28 <th style="width: 15%;>Apellido</th>
29 <th style="width: 15%;>Edad</th>
30 <th style="width: 10%;>Sexo</th>
31 <th style="width: 10%;>Acciones</th>
32 </tr>
33 </thead>
34 <tbody>
35 <tr>
36 <td>1</td>
37 <td>Juan</td>
38 <td>Perez</td>
39 <td>25</td>
40 <td>M</td>
41 <td><input type="button" value="Ver" class="btn btn-info" /> <input type="button" value="Editar" class="btn btn-warning" /> <input type="button" value="Eliminar" class="btn btn-danger" /></td>
42 </tr>
43 <tr>
44 <td>2</td>
45 <td>Maria</td>
46 <td>Gomez</td>
47 <td>30</td>
48 <td>F</td>
49 <td><input type="button" value="Ver" class="btn btn-info" /> <input type="button" value="Editar" class="btn btn-warning" /> <input type="button" value="Eliminar" class="btn btn-danger" /></td>
50 </tr>
51 <tr>
52 <td>3</td>
53 <td>Carlos</td>
54 <td>Rodriguez</td>
55 <td>28</td>
56 <td>M</td>
57 <td><input type="button" value="Ver" class="btn btn-info" /> <input type="button" value="Editar" class="btn btn-warning" /> <input type="button" value="Eliminar" class="btn btn-danger" /></td>
58 </tr>
59 <tr>
60 <td>4</td>
61 <td>Ana</td>
62 <td>Martinez</td>
63 <td>22</td>
64 <td>F</td>
65 <td><input type="button" value="Ver" class="btn btn-info" /> <input type="button" value="Editar" class="btn btn-warning" /> <input type="button" value="Eliminar" class="btn btn-danger" /></td>
66 </tr>
67 <tr>
68 <td>5</td>
69 <td>Pedro</td>
70 <td>Lopez</td>
71 <td>35</td>
72 <td>M</td>
73 <td><input type="button" value="Ver" class="btn btn-info" /> <input type="button" value="Editar" class="btn btn-warning" /> <input type="button" value="Eliminar" class="btn btn-danger" /></td>
74 </tr>
75 <tr>
76 <td>6</td>
77 <td>Sofia</td>
78 <td>Garcia</td>
79 <td>20</td>
80 <td>F</td>
81 <td><input type="button" value="Ver" class="btn btn-info" /> <input type="button" value="Editar" class="btn btn-warning" /> <input type="button" value="Eliminar" class="btn btn-danger" /></td>
82 </tr>
83 </tbody>
84 </table>
85 </div>
86 </body>
87 </html>
```

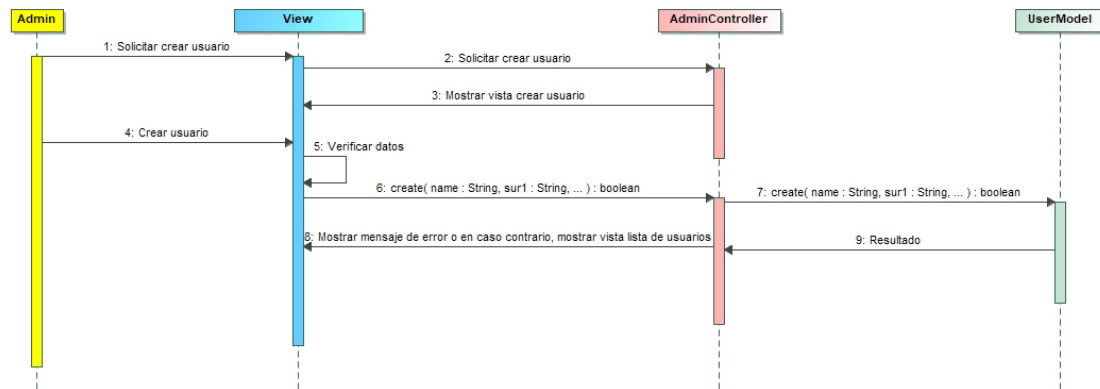
## 2.2.6 Diagrama de secuencias

A continuación, se muestran los diagramas de secuencia para las operaciones crear, editar, ver y eliminar del área de gestión. Debido a que prácticamente estas operaciones son iguales para los objetos usuario, receta e ingrediente se ha optado por escoger un objeto diferente para cada operación. También se ha representado el diagrama para la operación añadir ingredientes a una receta.

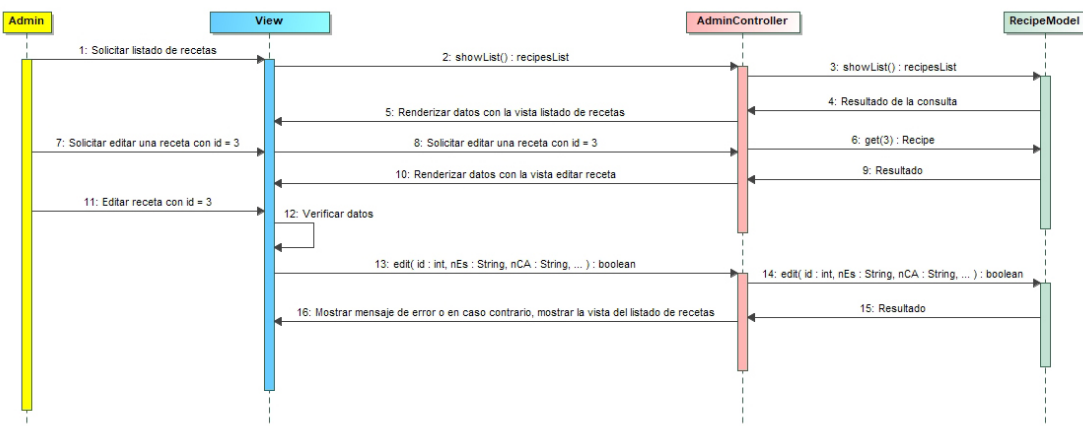
### Ver ingrediente



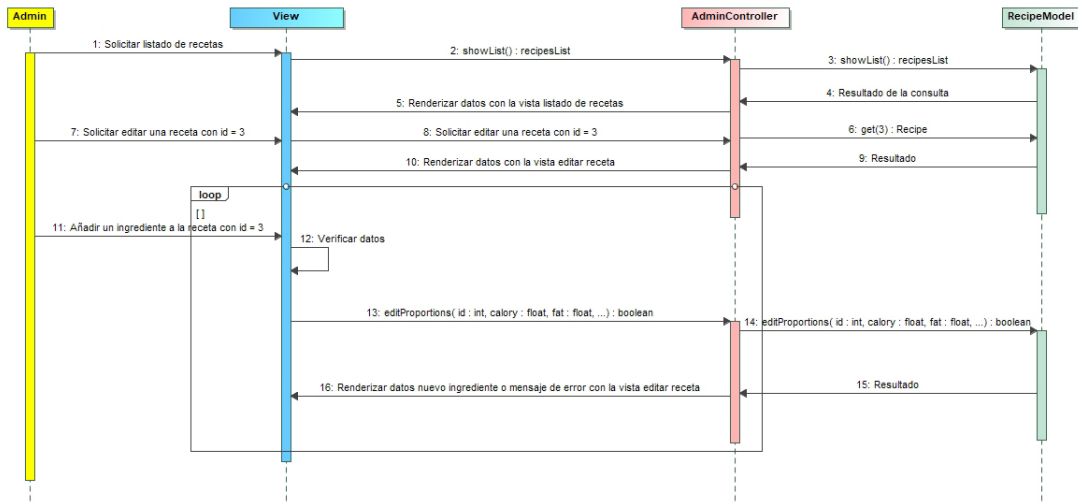
### Crear usuario



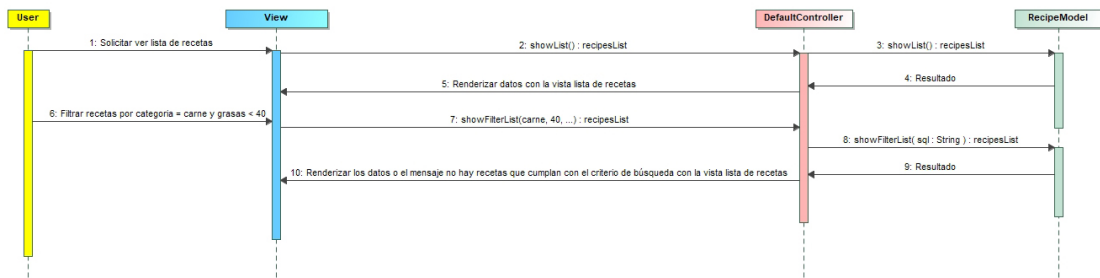
### Editar receta



## Añadir ingredientes a una receta

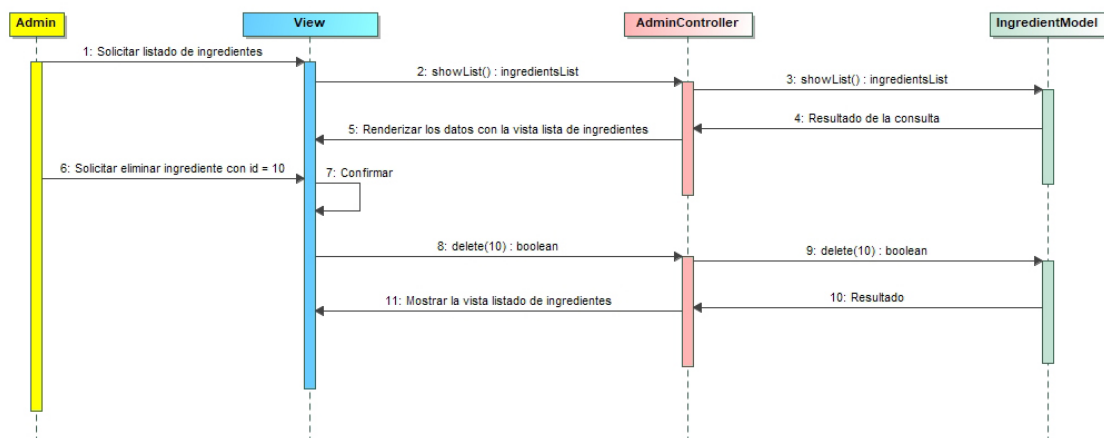


## Eliminar ingrediente



Este diagrama de secuencia representa la operación que realiza un usuario esté o no registrado para buscar recetas mediante filtro.

## Búsqueda de recetas con filtro



## **2.2.7 Configuración del entorno local y real**

### **Entorno local**

Para realizar tareas de programación ha sido necesario instalar un servidor web local. Se ha optado por utilizar el sistema de infraestructura de Internet WAMP Server, ya que permite instalar un servidor web Apache con un servidor Mysql de forma simple.

Por otro lado, se ha escogido el IDE Netbeans para programación en PHP por ser un IDE con el que ya estaba bastante familiarizado.

Para probar la aplicación en el entorno local, se han utiliza los navegadores siguientes: Chrome, Internet Explorer, Mozilla Firefox, Opera y Microsoft Edge. También ha sido necesario probar y adaptar el aspecto visual de la aplicación para que se mostrara correctamente en todos los dispositivos: pc, tableta y móviles de diferentes resoluciones.

### **Entorno real**

Por un lado, ha sido necesario llevar a cabo un pequeño estudio para seleccionar el proveedor de servicios de hosting y dominio que se iba a contratar. Los principales factores que se han tenido en cuenta a la hora de valorar el proveedor más idóneo han sido los siguientes:

- Tipo de base de datos (MySQL)
- Compatibilidad con múltiples sistemas
- Ubicación del servidor
- Velocidad de subida y bajada para gestionar la información
- Ancho de banda
- Un buen servicio de atención al cliente
- Precio

Después de realizar el análisis entre los siguientes hostings: Bluehost, iPage, HostGator, Hostinger, SiteBuilder, 1&1, GoDaddy, JustHost, Host1Plus y MediaTemple se ha escogido Hostinger por ser uno de los más adecuados respecto a las características del proyecto. La empresa Hostinger tiene ubicados los servidores en España, ofrece compatibilidad con múltiples sistemas, tiene buen ancho de banda y el precio es el más asequible comparado con la competencia.

## **2.3 Valoraciones económicas**

### **Gastos asociados al desarrollo**

No ha habido apenas gastos desde el inicio hasta el final del proyecto si no contamos con las horas de trabajo en todas las fases del proyecto. Por otro lado, no ha sido necesario la compra de ningún activo como herramienta para diseñar, programar o para otro tipo de tareas. No obstante, ha sido necesario la compra del dominio y el alquiler del hosting. Así pues, los gastos totales asociados al desarrollo han sido unos 11,95 €.

### **Gastos asociados al mantenimiento**

Si calculamos los gastos al año obtenemos unos gastos totales de 47,88 €. Este gasto resulta de multiplicar 11,95 € (gastos de hosting y dominio trimestral) por 4.

### **Beneficios obtenidos**

No se puede estimar los beneficios debido a que depende de muchos factores. Estos son algunos de ellos:

- El tiempo que se requerirá para que el sitio sea conocido por muchos usuarios
- Qué tipo de ingresos se escogerá como por ejemplo cobrar al usuario por los servicios prestados u obtener los ingresos por publicidad, etc.
- Cuánto tiempo se dispondrá para invertir en su mantenimiento.

### **Análisis de la viabilidad del proyecto**

No se puede realizar un análisis por falta de datos per pienso que puede ser viable a medio o largo plazo, Además, carece de riesgos ya que su mantenimiento conlleva poco gasto.

## **2.4 Riesgos**

En este apartado se comentan los riesgos que podrían afectar negativamente al proyecto clasificándolos en riesgos de proyecto, técnicos y de negocio. También se exponen algunas medidas para prevenirlos en el subapartado gestión de riesgos.

### **Riesgos de proyecto**

Estos riesgos pueden poner en peligro todo el proyecto, por lo tanto, exigen más atención que el resto de riesgos.

El riesgo que se ha detectado tiene que ver con la planificación, ya que ha habido una desviación que se ha generado en la fase de diseño. Esta desviación ha provocado que dos tareas de esta fase no pudieran desarrollarse y también ha puesto en peligro la fase PAC4.

También ha habido un riesgo respecto a la falta de especialización en cuanto a PHP y JavaScript que ha provocado retrasos y tareas sin terminar en la fase PAC3.

### **Riesgos técnicos**

Estos riesgos ponen en riesgo la calidad del proyecto.

Por un lado, se han detectado riesgos de implementación ya que para realizar todas las tareas planificadas de la fase PAC3 se requería más tiempo.

Por otro lado, pueden surgir riesgos que tengan que ver con la usabilidad y con la detección de errores por no haber realizado test de pruebas.

### **Riesgos de negocio**

Respecto a esta categoría se podría considerar el riesgo de que la idea del proyecto no cubra una necesidad real y por lo tanto no sea útil.

### **Gestión de riesgos**

Las dos estrategias a seguir para gestionar los riesgos son: estrategia reactiva y estrategia proactiva. En la proactiva se realiza un estudio al comienzo del proyecto y la reactiva se van solucionando según se van detectando. En este caso se optará por la reactiva, ya que no se ha tenido en cuenta los riesgos al comienzo del proyecto.

## **2.5 Mantenimiento**

Se muestran las tareas habituales que se llevaran a cabo periódicamente para asegurar el buen funcionamiento de la aplicación durante todo su ciclo de vida.

### **Revisión periódica de su correcto funcionamiento**

Una vez por semana se deberá verificar que todos los enlaces, páginas y funcionalidades funcionan correctamente. En caso contrario se tomarán medidas correctivas. Se les indicará los problemas una prioridad alta, media o baja según su nivel de criticidad.

### **Eliminación de cuellos de botella**

Realizar pruebas para comprobar que no existen cuellos de botella que hagan que la web se ralentice debido al aumento de usuarios. Las pruebas se podrían realizar llegado a un porcentaje de usuarios. La prioridad en este caso será media.

### **Detección de posibles ataques**

Detectar y bloquear los posibles ataques que sufra la web. También se tomarán medidas correctivas para evitar problemas de seguridad. La prioridad en este caso será alta.

### **Corrección de problemas no detectados en la fase de diseño e implementación**

Estos problemas serán solucionados en cuanto se tenga conocimiento de ellos. Dependiendo de lo que puedan afectar al sistema se les podrá asignar una prioridad alta, media o baja.

### **Mejoras de rendimiento**

Según se vaya necesitando se podrá incrementar el rendimiento y la velocidad de la página web. La prioridad será baja, pero se deberá llevar un seguimiento continuo.

## **2.6 Ampliaciones futuras**

### **Funcionalidades en fase de desarrollo**

Breve descripción de las funcionalidades que no se han podido presentar y constaban en la planificación. Actualmente, estas funcionalidades se encuentran en fase de desarrollo.

#### **Planificador de recetas**

Si el usuario está registrado podrá generar un planificador de recetas. El planificador permitirá al usuario obtener una planificación de recetas diferentes durante un intervalo de tiempo con tres recetas por día (desayuno, comida y cena). Deberá escoger el tipo de alimentación: alimentación variada, alimentación variada y saludable y alimentación proteica y el máximo de días (máximo dos semanas).

#### **Generador de dietas**

Si el usuario está registrado también tendrá acceso a generar una dieta adelgazante o para deportistas. Deberá añadir los valores siguientes: peso, edad, altura y sexo y el sistema generará la dieta. Además, mostrará una lista de recetas relacionadas con la dieta.

#### **Área de usuario**

Espacio donde el usuario podrá modificar sus datos personales como nombre, apellidos, email, contraseña y datos necesarios para otras funciones como sexo, edad, altura y peso. También permitirá ver sus recetas publicadas, sus comentarios a otras recetas y comentarios sobre sus publicaciones. Por último, desde aquí el usuario podrá eliminar su cuenta cuando lo desee.

### **Funcionalidades futuras**

Se presentan algunas de las funcionalidades que podrían implementarse en un futuro.

#### **Los usuarios registrados podrán editar recetas**

Permitir a los usuarios registrados crear, editar y eliminar sus propias recetas. Las recetas incluirán el autor de la receta.

#### **Los usuarios registrados podrán votar recetas**

Permitir a los usuarios registrados votar las recetas publicadas por RecipeRepo y las publicadas por otros usuarios. Las recetas mostrarán la puntuación en una escala del 1 al 5.



## Los usuarios podrán comentar recetas

Los usuarios registrados podrán hacer comentarios sobre las recetas de otros usuarios y las recetas propias de RecipeRepo.

## Nueva sección del *backend* para añadir imágenes y popups

La implementación de una nueva sección del *backend* que permitirá desde la aplicación añadir y modificar las imágenes de toda la aplicación. También se podrá añadir y configurar un *popup* para que sea visible desde cualquier parte de la web o sólo en unas determinadas páginas.

## Adaptación a los microformatos de hRecipe

Uno de los pilares más importantes de la web semántica son los microformatos que permiten que el contenido de las webs sea comprendido no sólo por humanos, sino también por entes como buscadores o navegadores. La ventaja más destacable del uso de los microformatos es que se obtiene mejor posicionamiento web porque permite a los buscadores representar mejor la información.

hRecipe es un microformato para describir recetas utilizado por el buscador de recetas de Google y que podría aportar grandes beneficios a RecipeRepo respecto a posicionamiento. Actualmente, las etiquetas hRecipe se están usando en muchas webs dedicadas a la temática de las recetas.

### *¿Que es hRecipe?*

*Es un proyecto de microformato que esta en la versión 0.22. que lo define como: simple, abierto, en formato distribuido, muy valido para incluir información de recetas de cocina en (X)HTML, Atom, RSS, y XML arbitrario [4].*

A continuación, se expone el esquema a seguir y se comenta cada propiedad para describir una receta con el uso del microformato hRecipe.

## hrecipe

**fn.** requerido. texto. el nombre de la receta.

**ingredient.** requerido. 1 o más. texto opcionalmente con código (x)html.

**value** y **type.** opcional. [experimental]

**yield.** opcional. texto.

**instructions.** opcional. texto opcionalmente con código (x)html.

**duration.** opcional. 1 o más. texto.

**photo.** opcional. 1 o más. usando algún enlace URL, a una imagen.  
[experimental]

**summary.** opcional. texto. [experimental]

**author.** opcional. 1 o más. [experimental]

**published.** opcional. [experimental]

**nutrition.** opcional. 1 o más. [experimental]

**value** y **type.** opcional. [experimental]

**tag.** opcional. 1 o más. [experimental]

**hrecipe:** permite describir e identificar valores y metadatos sobre recetas.

**fn:** Título de la receta

**ingredients:** Describe uno o más ingredientes usado por la receta.

**yield:** Especifica la cantidad, cuantas personas pueden comer o cuantas unidades podemos obtener.

**instructions:** El proceso de la receta.

**duration:** Tiempo de preparación

**photo:** Imagen de la receta

**summary:** Introducción o declaración sobre la receta.

**author:** Autor de la receta.

**published:** Fecha de publicación de la receta.

**nutrition:** Información nutricional.

**tag:** Una palabra clave que indica una característica importante de la receta.

## 3. Conclusiones

### Lecciones aprendidas

En la fase de análisis (PAC2) se ha sabido extraer y clasificar los requisitos en funcionales y no funcionales. Después, estos requisitos se han documentado con el método historias de usuario y se ha terminado de perfilar con los casos de uso. También se ha aprendido a aplicar algunos de los diagramas que ofrece el lenguaje de modelado UML. Estos han sido los siguientes: diagrama de clases para la representación del dominio de la aplicación, diagramas de paquetes para representar la estructura o arquitectura de la aplicación. Para la representación de los datos se ha aprendido a utilizar técnicas para el diseño lógico de la base de datos.

En la fase de diseño (PAC3), se ha comprendido que los productos obtenidos en la fase anterior eran susceptibles de experimentar cambios ya fuese por errores detectados o mejoras consideradas en esta fase. Se ha comprendido que a pesar de estar en la fase de diseño se sigue realizando algunas tareas de análisis como el diagrama de secuencias para modelizar el proceso de algunas funciones de la aplicación. También se ha aprendido a omitir representaciones del diseño lógico que no son relevantes para crear el diseño físico de la base de datos. Respecto al desarrollo de la aplicación, se ha aprendido a utilizar el patrón MVC y se ha implementado con éxito.

### Logro de los objetivos planteados inicialmente

Todo el objetivo planteado inicialmente se ha logrado excepto las tareas planificación y generador de dietas de la fase de diseño y que correspondían al desarrollo de la aplicación. El motivo ha sido por falta de tiempo, no se ha sabido prever el tiempo real que supondría realizar estas tareas.

### Análisis crítico sobre el seguimiento de la planificación y metodología

En la fase de planificación (PAC1) y la fase de análisis (PAC2) se ha seguido tal y como estaban marcadas las fechas para cada tarea. Sin embargo, ha habido una importante desviación en la planificación en la fase de diseño (PAC3). Esta desviación ha provocado que algunas de las tareas de desarrollo que estaban previstas no pudieran llegar a ser completadas. Pero no sólo se ha visto afectada esta fase, sino que también ha sufrido cambios temporales la fase (PAC4) pero que se han podido solventar.

Respecto a la alternativa utilizada se puede decir que en este caso ha sido la adecuada, ya que la posibilidad de partir de una base ya creada no ha sido posible, pero se han aprovechado conceptos y técnicas ya existentes. También ha sido acertada la metodología escogida, ya que con la RMM se ha podido llevar un proceso correcto en las fases de análisis, diseño y desarrollo.

## **Líneas de trabajo futuro**

Por un lado, han quedado pendientes el desarrollo de funciones como poder mostrar una planificación de recetas y generar un seguimiento para hacer una dieta alimenticia. Estas funciones son muy interesantes y formaban parte del atractivo del trabajo. Por otro lado, implementar los microformatos hRecipe es una línea de trabajo muy interesante que aportaría muchos beneficios como un mayor posicionamiento en poco tiempo y, además, la aplicación estaría preparada para el nuevo salto de la web, la web semántica.

## 4. Glosario

**ENPE:** Estudio Nutricional de la Población Española.

**Microformato:** *Según el W3C, son conjuntos de formatos de datos abiertos y simples, desarrollados sobre estándares ya existentes, ampliamente adoptados, incluyendo XHTML (HTML y XML) y CSS [5].*

**Backend:** Es la capa o área de una aplicación web que se encarga principalmente de la gestión de los datos. Pero se le puede atribuir otras funciones que tengan que ver con el diseño como gestionar las imágenes de la web.

**Framework:** *Un framework, entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar [6].*

**MVC:** Es un patrón arquitectónico que permite separar la presentación de la lógica y los datos de una aplicación.

**Hosting:** Servicio de alojamiento para páginas y aplicaciones web en Internet.

**Dominio:** La dirección única que identifica un sitio en Internet.

**Responsive:** Técnica de diseño web para la correcta visualización de una página en distintos dispositivos.

**Sha1:** Algoritmo de cifrado que se utiliza para encriptar las contraseñas.

**LOPD:** Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal.

**UML:** Es un lenguaje de modelado para diseñar sistemas de software.

**PHP:** Es un lenguaje de código abierto muy extendido en el desarrollo de páginas web dinámicas.

**JavaScript:** Es un lenguaje de programación interpretado que se utiliza para controlar los eventos en una página web.

**JQuery:** Biblioteca multiplataforma de JavaScript.

**HTML5:** Es la quinta versión del lenguaje básico de la web.

**CSS3:** Es un lenguaje para definir la presentación visual de un documento estructurado con HTML.

**Singleton:** Es un patrón de diseño de programación que permite restringir una clase de código para que sólo se pueda instanciar una vez.

## 5. Bibliografía

[1]

<http://www.efesalud.com/estudio-enpe-mas-del-60-de-los-espanoles-sufre-sobrepeso-y-obesidad/>

20/09/2017

[2]

<http://www.lavoz.com.ar/noticias/tecnologia/google-lanza-recipe-view-opcion-para-buscar-recetas>

24/09/2017

[3]

<https://programacionwebisc.wordpress.com/2-5-metodologias-para-el-desarrollo-de-aplicaciones-web/>

12/10/2017

[4]

<https://csanchezmonte.wordpress.com/2012/02/12/google-acepta-hrecipe-como-microformato-para-las-recetas/>

2/11/2017

[5]

<https://es.wikipedia.org/wiki/Microformato>

10/01/2018

[6]

<https://es.wikipedia.org/wiki/Framework>

10/01/2018