

Comparte tu **cocina**



Trabajo final de curso
Jhonatan Ruiz

1–Naturaleza del proyecto	3
2- Definición de los objetivos	4
3- Recursos	4
4- Tareas	4
5- Capítulos de desarrollo	5
6- Análisis	12
7- Mapa de navegación .	19
8- Funcionalidades y características	20
9- Conclusiones y propuestas	32
10- Bibliografía y referencias	33

1–Naturaleza del proyecto

Descripción del proyecto

El proyecto consiste en el desarrollo y despliegue de una aplicación web en la que un usuario puede buscar una receta de cocina a través del título, o el ingrediente principal; y en la gestión de usuarios que pueden publicar nuevas recetas.

Esta aplicación web se base en una petición a un api de recetas de cocina llamada Edaman que tiene alrededor de 700.000 recetas de cocina publicadas en diferentes sitios web.

Esta petición nos devuelve unos datos en formato json que luego se muestran de una forma legible y bonita al usuario.

La gestión de usuarios que pueden publicar una receta de cocina se base en la comprobación de que dicho usuario exista en la base de datos del sitio web, que se hace por medio de una base de datos en mysql.

El objetivo principal es compartir recetas de cocinas, y mostrar a los usuarios diferentes recetas que están publicadas en diferentes sitios de internet o aquellas que los propios usuarios han publicado. De esta forma se fomenta la gastronomía de diferentes usuarios.

Justificación/fundamentación.

A muchos jóvenes inexpertos en la cocina no se les ocurre con frecuencia qué plato preparar cuando viene la hora de comer, y muchas veces acaban recurriendo a lo de siempre: filete con arroz, macarrones, huevo frito, patatas fritas, etc. Platos sencillos que todo el mundo sabe hacer. Por este motivo pensé en hacer una aplicación que te ofrezca diferentes recetas de cocina.

Por otra parte siempre me han gustado las distintas formas en las que se preparan una misma receta, y conocer gastronomía de diferentes países, y por ello se me ocurrió que los usuarios de esta aplicación web puedan publicar sus recetas, con su forma y trucos de hacerla.

Esta aplicación web está creada para buscar recetas que otros usuarios hayan publicado en internet en distintas páginas, a través del api usado y para ver las recetas que publican los usuarios registrados.

Tiene como fundamentación diseñar un portal web amigable y adaptable a cualquier dispositivo móvil, es decir, con un diseño responsive.

Beneficiarios del proyecto

Los beneficiarios del proyecto son todos aquellos usuarios que accedan a la aplicación web para buscar una receta, ya que conocerán diferentes formas de hacer una misma receta gracias al api usado y también a los diferentes usuarios que publican sus recetas de cocinas y sus trucos culinarios para que sean más sabrosas.

Localización del proyecto: dónde se llevará a cabo

Se va a desplegar la aplicación web en una plataforma de hosting gratuita bajo el dominio : <http://www.compartetureceta.byethost7.com/web/>

Dicha plataforma de hosting gratuito es: <https://byet.host/>

Para poder subir los archivos de esta aplicación se ha utilizado el programa FileZilla Client, que es un gestor FTP para administrar archivos en un servidor.

Como se ha subido a Internet tiene un alcance global de tal forma que cualquiera que tenga acceso a Internet podrá acceder a la aplicación web y realizar consultas; y si está registrado, podrá publicar nuevas recetas.

2- Definición de los objetivos

Características de los objetivos

El objetivo principal del proyecto es el diseño y despliegue de un sitio web en el que se consulten recetas de cocinas de todos los sitios, tanto de los usuarios de la aplicación como de las recetas que estén publicadas en otro sitio web. Para realizar este objetivo serán necesarios los siguientes objetivos secundarios:

- Permitir al usuario la creación y consulta de recetas.
- Permitir al usuario la consulta de recetas mediante la api de alimentos utilizada.
- Permitir el inicio de sesión del usuario a la plataforma.
- Permitir el registro del usuario a la plataforma.
- Permitir la eliminación de las recetas.
- Diseñar una interfaz amigable y adaptable a cualquier dispositivo móvil.
- Proporcionar recetas a jóvenes inexpertos en la cocina.

Por otra parte se pretende llegar a demostrar la capacidad del autor para desarrollar una aplicación web, basada en los conocimientos adquiridos durante todo el curso. Y refrescar y obtener nuevos conocimientos, adquiridos durante el desarrollo del proyecto.

3- Recursos

Para el desarrollo de este proyecto se ha utilizado una máquina electrónica, el ordenador, principal herramienta para el desarrollo y programación de todas las partes. Para el despliegue se ha usado un hosting gratuito. Para la gestión de los datos tanto de los usuarios, como de las recetas se ha usado una base de datos en mysql. Se han consultado los apuntes dados durante todo este año.

4- Tareas

Las distintas tareas que se realizan en esta aplicación son :

- El libre registro de los usuarios: Cualquier persona podrá acceder a la aplicación.
- El control de los usuarios : Los usuarios podrán acceder a la aplicación ingresando su email y contraseña.
- El control de las recetas publicadas.
- La eliminación de las recetas.
- La búsqueda de recetas publicadas en otros sitios web.
- El cierre de sesión de los usuarios.

5- Capítulos de desarrollo

Tecnologías para el desarrollo

A la hora de desarrollar un proyecto tecnológico una de las primeras preguntas que surgen son :

- ¿ Qué herramientas tecnológicas emplear ?
- ¿ Con qué lenguaje desarrollarlo ?
- ¿ Qué páginas o funcionalidades va a tener mi aplicación ?

El principal factor a tener en cuenta a la hora de elegir las herramientas tecnológicas es la popularidad que tengan dichas herramientas y su fácil aprendizaje, las distintas plataformas con las que esté implementadas y la cantidad de documentación que tengan dichas herramientas.

Partiendo de esta base se han empleado para el desarrollo de este aplicación web las siguientes herramientas :

Para el desarrollo web en local de todo el proyecto se a empleado XAMPP.

XAMPP, consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre de XAMPP es un acrónimo : **X**(cualquier sistema operativo)**A**pache, **M**ariaDB/**M**ySQL,**P**HP,**P**erl, se distribuye con la licencia GNU y actúa como un servidor web fácil de usar y capaz de interpretar páginas dinámicas.

Como editores de texto y código se han usado las siguiente herramientas:

Notepad++: Es un programa para editar código fuente de cualquier lenguaje de programación. Tiene soporte para una gran cantidad de lenguajes

Es de estos editores que ofrecen ayudas muy útiles para "tirar líneas de código", como resaltado de colores, posibilidad de editar varios documentos a la vez, menús contextuales, auto-completar código, etc. Todo un regalo para los programadores, ya que además es gratuito.

Brackets: editor de código diseñado por Adobe y de código abierto destinado a desarrolladores web y con una interfaz y funciones que contribuyen a evitar distracciones. Involucra por completo HTML, CSS, Javascript y mejora la productividad.

Es un editor de código liviano y potente, y al mismo tiempo incluye herramientas visuales permitiendo obtener la ayuda que se necesite. Un moderno editor, de código abierto de texto que comprende el diseño web.

Como lenguaje de programación utilizado para el desarrollo de la parte del servidor se ha usado PHP. Lenguaje impartido durante todo este curso, y que gran parte de los sitios web usan para el desarrollo de sus sitios web. Una búsqueda en <https://w3techs.com/> nos muestra los lenguajes de programación más populares en la parte del servidor.

Server-side Programming Languages

Most popular server-side programming languages

© W3Techs.com	usage	change since 1 April 2019
1. PHP	79.0%	
2. ASP.NET	11.2%	-0.2%
3. Java	4.0%	
4. Ruby	2.5%	
5. static files	2.1%	

percentages of sites

PHP acrónimo recursivo en inglés de **PHP: Hypertext Preprocessor** (preprocesador de hipertexto) es un lenguaje de programación del lado del servidor, enfocado en la creación de webs dinámicas. Sus *scripts* son interpretados por el servidor y genera código HTML. Requiere Apache o IIS con librerías de PHP. Hereda su sintaxis de C, Java y Perl.

Como principales ventajas, hemos de decir que es un lenguaje fácil de aprender y muy rápido. Soporta la orientación a objetos y utiliza un lenguaje multiplataforma. Además, puede conectarse con una gran cantidad de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server... No necesita que se definan los tipos de variables. Uno de sus aspectos más llamativos es que está diseñado con el fin de ser un lenguaje muy seguro para escribir CGI, más que Perl o C.

Es el lenguaje base que utilizan la mayoría de CMS o gestores de contenidos más extendidos como WordPress, PrestaShop, Drupal o Joomla!

Para el diseño de la parte del cliente (front-end), la parte que visualiza el usuario, se ha usado HTML, CSS, Javascript y algunos framework y librerías como Bootstrap y JQuery, Fonts icons (fontawesome), que han facilitado el desarrollo del diseño y la interacción del usuario con el sitio web.

HTML, cuyas siglas en inglés son **HyperText Markup Language** (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica

y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web es el que todos los navegadores actuales han adoptado.

CSS (siglas en inglés de **C**ascading **S**tyle **S**heets), en español "Hojas de estilo en cascada", es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

Los css marca la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

JavaScript (abreviado comúnmente **JS**) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos basado en prototipos, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (front-end), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas dinámicas, aunque hoy en día, el uso de Javascript se ha extendido a la parte del servidor e incluso a la parte de las bases de datos, gracias al *Server-side JavaScript* o *SSJS*.

Junto con HTML, Javascript y CSS son una tecnologías usadas por muchos sitios web, para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web.

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript tradicionales. A

diferencia de muchos frameworks web, que solo se ocupa del desarrollo front-end. Bootstrap es el segundo proyecto más destacado en GitHub.

JQuery es una biblioteca multiplataforma de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. De acuerdo a un análisis de la Web (realizado en 2017) JQuery es la biblioteca de JavaScript más utilizada. Ofrece una serie de funcionalidades basadas en JavaScript.

Font Awesome es una herramienta de fuente e icono que se basa en CSS, LESS y SASS. Fue creado por Dave Gandy para usar con Bootstrap, y luego se incorporó a BootstrapCDN. Font Awesome tiene una participación de mercado del 20% de los sitios que utilizan scripts de fuentes de terceros en su plataforma, lo que lo coloca en segundo lugar después de Google Fonts. Font Awesome 5 se lanzó el 7 de diciembre de 2017.

Organización del proyecto.

Para el desarrollo software tener una buena organización del proyecto es esencial, ya que te permite tener todo los recursos empleados bajo control, por ello es muy importante seguir una metodología de trabajo, la cual se definirá más adelante. El uso de una metodología de trabajo conduce a realizar las partes del proyecto de una forma más ordenada y estructurada; el conocimiento del trabajo que se está realizando, las fases por las que pasa y las que quedan por realizar para la finalización del proyecto; son algunas ventajas que nos aportan seguir una metodología de trabajo.

Para la realización de este proyecto se ha usado una metodología tradicional de desarrollo software, dividiendo en dos partes la aplicación. Por un lado el *back-end* y por el otro el *front-end*.

Back-end

El back-end de la aplicación está formado por una llamada a una api de alimentos y a una base de datos en mysql.

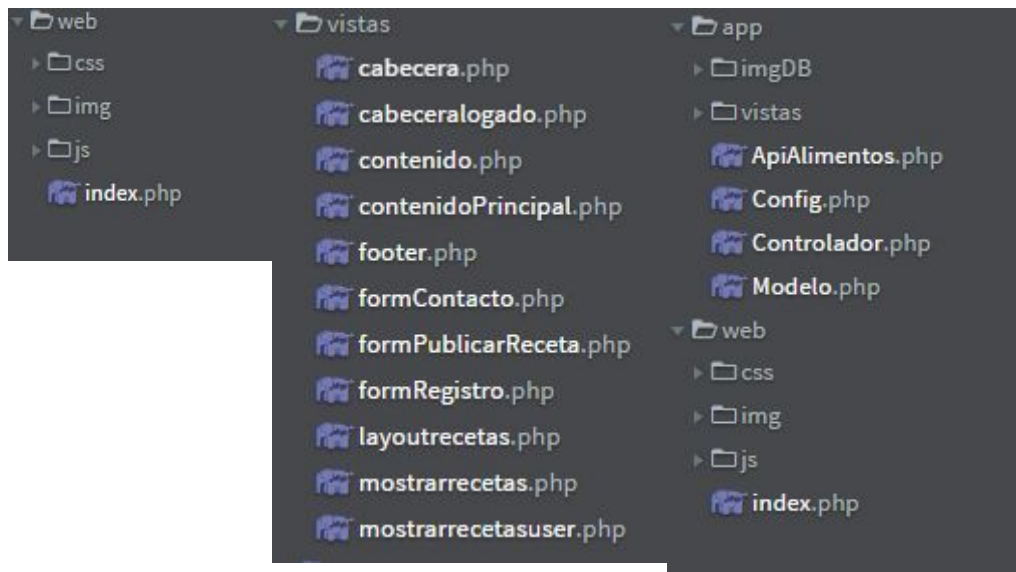
Front-end

El front-end de este proyecto está formado por un conjunto de vistas, un controlador, y unos frameworks y librerías, como bootstrap y JQuery.

Los distintos componentes de la aplicación se han estructurado en capas, siguiendo un patrón de diseño, llamado Modelo, Vista, Controlador (MVC), que consiste en separar

en tres capas la totalidad del proyecto: la base de datos, la lógica de negocio y la interfaz.

A continuación muestro cómo están estructurados los componentes de este proyecto.



Las diferentes capas que forman el MVC consisten en:

MODELO

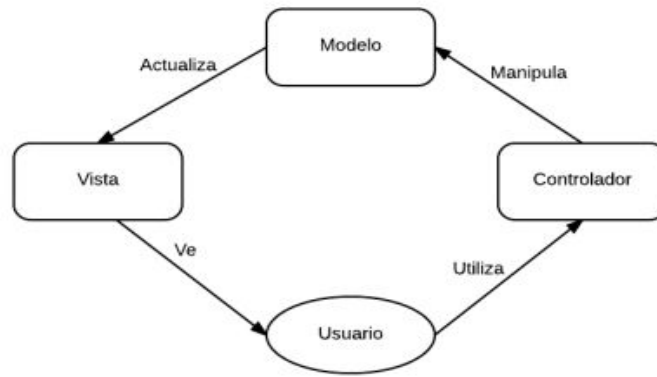
Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. Esta puede acceder a los datos sin importar la forma en la que estén guardados.

CONTROLADOR

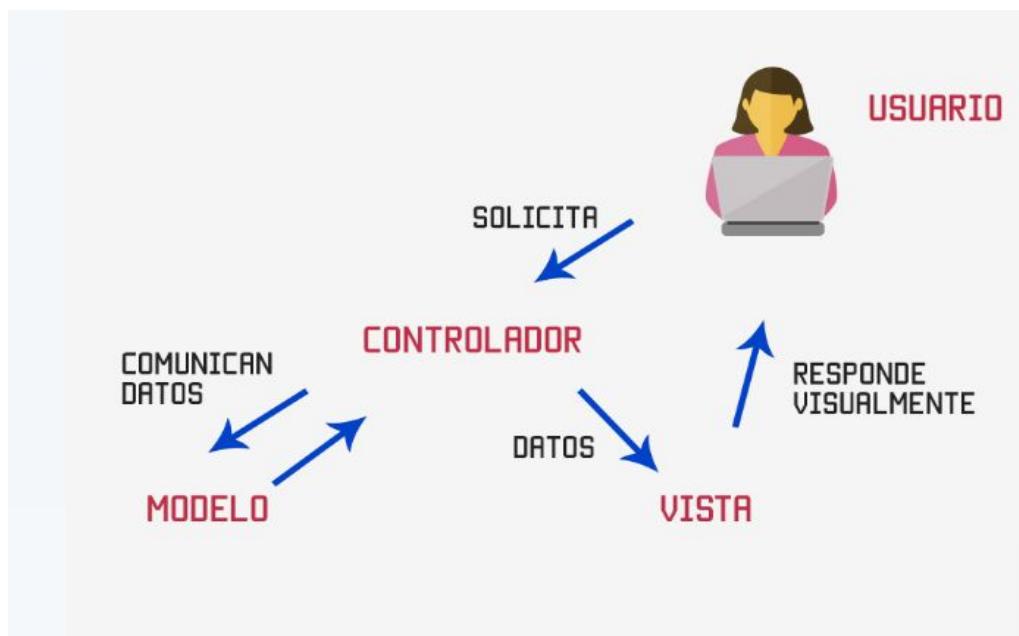
Se encarga de controlar, recibe las órdenes del usuario y las diferentes funcionalidades de la aplicación. Solicitar los datos al modelo y enviárselos a la vista.

VISTA

Es la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista. Convierte las llamadas del usuario en llamadas a las diferentes capas y funcionalidades.



De esta forma la lógica de la aplicación está separada de la vista. Este patrón de diseño independiza los distintos componentes que constituye este proyecto, de tal forma que cuando hacemos un cambio un cambio en alguna parte de nuestro código, esto no afecte otra parte del mismo. Por ejemplo, si modificamos nuestra Base de Datos, solo deberíamos modificar el modelo que es quién se encarga de los datos y el resto de la aplicación debería permanecer intacta.



Esto respeta el principio de la responsabilidad única. Es decir, una parte de tu código no debe de saber qué es lo que hace toda la aplicación, solo debe de tener una responsabilidad.

La reutilización de código, el fácil mantenimiento y escalabilidad de la aplicación, de tal forma que se hace más fácil y cómodo incorporar nuevas funcionalidades.

Organización de las rutas.

Las rutas a las diferentes pantallas de este proyecto se han realizado de forma que desde el index.php se llame a las distintas pantallas :

```
$map = array(
    'inicio' => array('controlador' => 'Controlador','action' =>
        'inicio'),
    'formContacto'=> array('controlador' => 'Controlador','action' =>
        'mostrarFormContacto'),
    'formRegistro'=> array('controlador' => 'Controlador','action' =>
        'mostrarFormRegistro'),
    'login' => array('controlador' => 'Controlador','action' =>'login'),
    'cerrarSesion' => array('controlador' => 'Controlador','action' =>
        'salir'),
    'buscarReceta' => array('controlador' => 'Controlador','action' =>
        'getRecetaApi'),
    'publicarReceta' => array('controlador' => 'Controlador','action'
        =>'formPublicarReceta'),
    'recetasUsuarios' => array('controlador' => 'Controlador','action'
        =>'getRecetaUser'),
    'eliminarReceta' => array('controlador' => 'Controlador','action'
        =>'eliminarReceta'),
);
session_start();
```

6- Análisis

Diagrama de clases

El diagrama de clases tiene como finalidad mostrar la estructura del sistema que se pretende desarrollar, mostrando las tablas más importantes y las relaciones que hay entre ellas.

Usuario. Representa al dueño de una cuenta, aquel que podrá ingresar al sitio web y formar parte del sitio una vez registrado. Esta entidad tiene como atributos : el **id**, el **nombre** y apellido, el **nombre de usuario**, el **email**, el **password**, el **país** y la **ciudad**.

Cabe destacar que el nombre de usuario nunca podrá repetirse, ya que al registrar un usuario se comprueba que dicho nombre de usuario no esté en la base de datos.

Receta. Representa el contenido que el usuario registrado creará. Esta entidad tiene como atributos: el **id**, el **título**, una **imagen de la receta**, el **tiempo de preparación**, los **ingredientes**, los **pasos** a seguir, y el **usuario** que publicó esa receta.

Cabe destacar que solo se pueden introducir imágenes JPG, PNG y JPEG con un tamaño menor a 1 mb.

Ingredientes. Representa el grupo de ingredientes que formará parte de una receta. Esta entidad tiene como atributos: el **id**, el **nombre** del ingrediente, la **cantidad** y una referencia a la receta.

Pasos. Representa el grupo de pasos que harán falta para la elaboración de una receta. Esta entidad tiene como atributos: el **id**, la **descripción** del paso y una referencia a la receta.

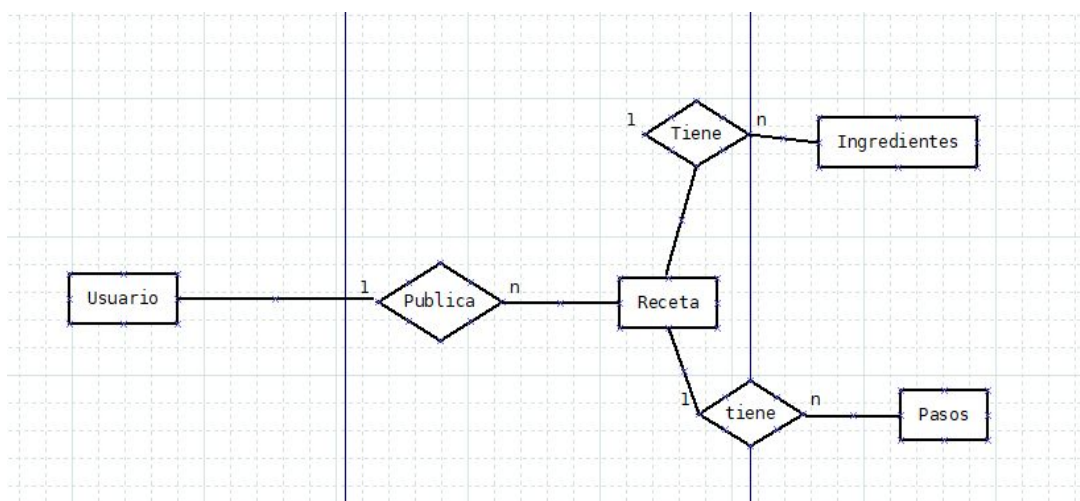
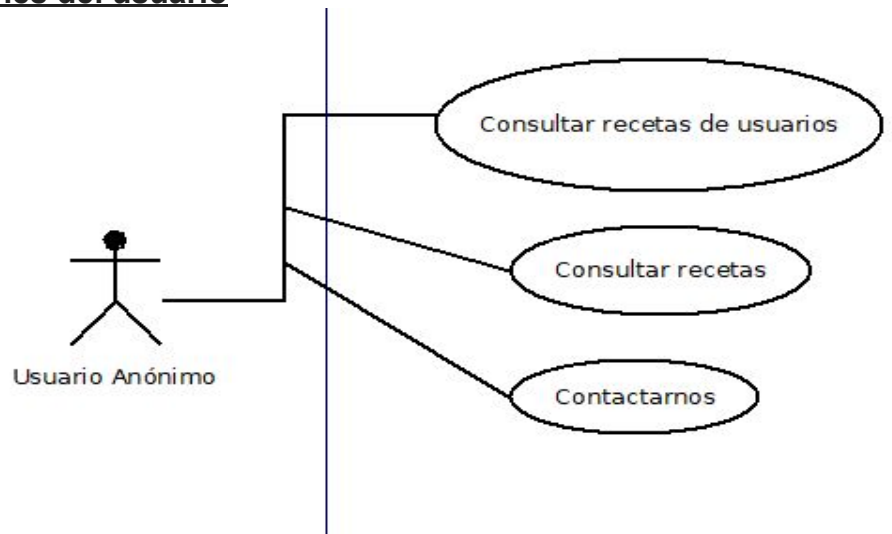
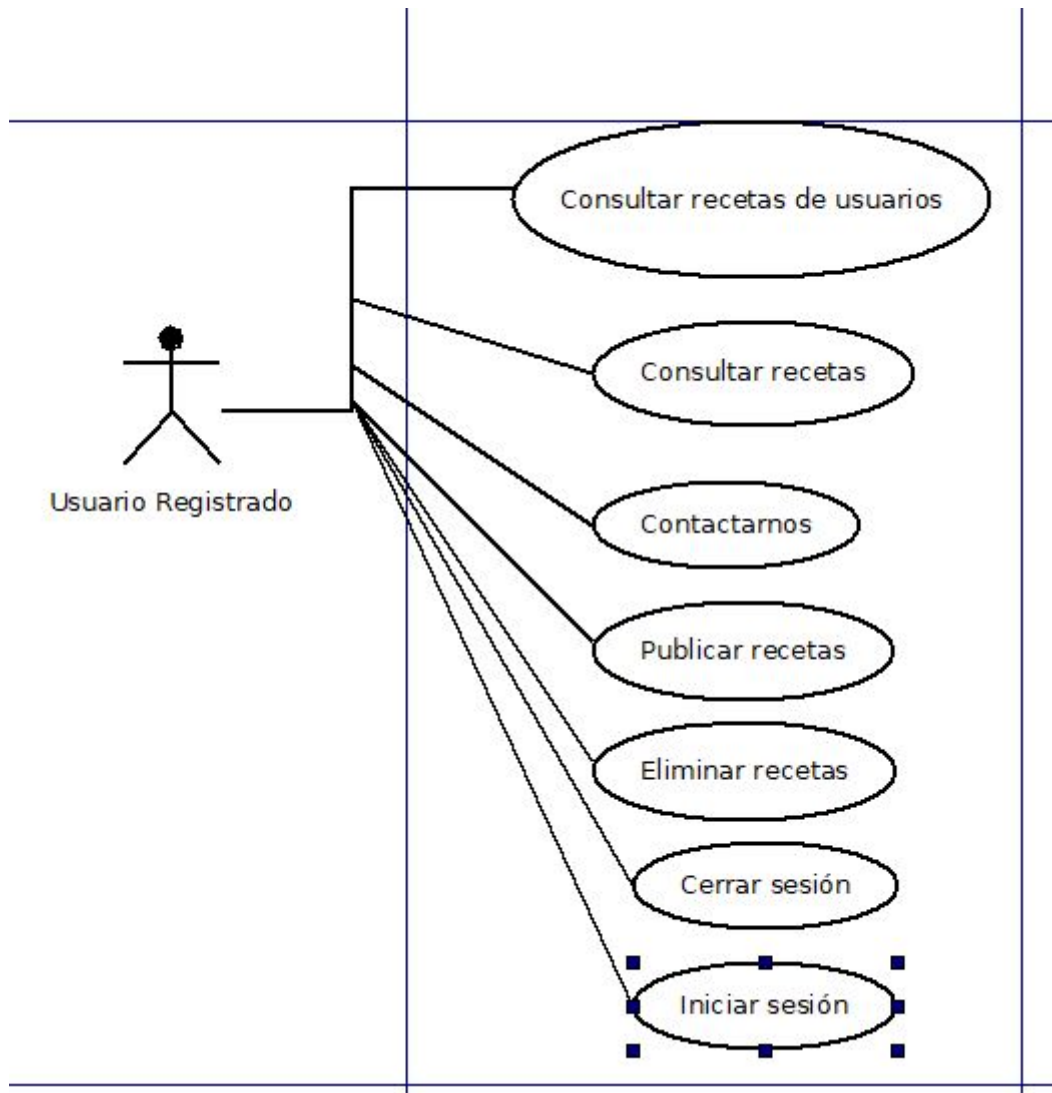


Diagrama de casos de uso

Este tipo de diagrama nos ayuda a capturar la información de cómo funciona el sistema, es decir, captura los requisitos funcionales del sistema que se pretende desarrollar. Está formado por el usuario, quien representa una entidad que interacciona con el sistema, y por los casos de uso, las distintas funcionalidades del sitio.

Funciones del usuario





A continuación se explica con más detalle los casos de uso.

Caso de uso	Registrarse
Actor	Usuario anónimo
Descripción	Crear una cuenta nueva para acceder al sitio web.
Secuencia del caso	<p>El usuario solicita registrarse.</p> <p>Se muestra un formulario para que introduzca sus datos.</p>

	<p>Se comprueba que los datos sean válidos y no estén vacíos.</p> <p>Se comprueba que el nombre del usuario no esté en la base de datos.</p> <p>Se registra al usuario en la base de datos</p>
--	--

Caso de uso	Buscar recetas
Actores	Usuario anónimo y registrado
Descripción	Buscar recetas a través del api
Secuencia del caso	<p>Se muestra una barra de búsqueda para que el usuario introduzca la receta que quiere buscar.</p> <p>Se llama al api</p> <p>Se comprueba que el api devuelve datos</p> <p>Se muestran los datos al usuario de una forma bonita y legible.</p>

Caso de uso	Contactarnos
Actor	Usuario anónimo y Usuario registrado
Descripción	Mostrar un formulario de contacto al usuario
Secuencia del caso	<p>El usuario solicita contactarnos.</p> <p>Se muestra un formulario para que introduzca sus datos.</p>

	<p>Se comprueba que los datos sean válidos y no estén vacíos.</p> <p>Y recibimos su mensaje.</p>
--	--

Caso de uso	Iniciar sesión
Actor	Usuario registrado.
Descripción	Permite al usuario acceder a su cuenta previamente creada para utilizar la aplicación.
Secuencia del caso	<p>El usuario solicita iniciar sesión</p> <p>El sistema le muestra un formulario en el que debe de introducir su email y su contraseña.</p> <p>El sistema creará una sesión y le enviará al inicio de la aplicación.</p>

Caso de uso	Cerrar sesión
Actor	Usuario registrado
Descripción	Permite a un usuario eliminar una sesión
Secuencia del caso	<p>El usuario solicita el cierre de sesión</p> <p>El sistema elimina la sesión creada cuando se logeo</p>

Caso de uso	Consultar recetas usuarios
Actor	Usuario registrado
Descripción	Muestra al usuario las recetas que han publicado los otros usuarios
Secuencia del caso	El sistema proporciona un listado de las recetas que han publicado todos los usuarios de la aplicación.

Caso de uso	Eliminar recetas
Actor	Usuario registrado
Descripción	El usuario creador de esta receta podrá eliminar la receta que haya creado
Secuencia del caso	<p>Al usuario creador de la receta se le mostrará un botón con el cual podrá eliminar la receta que haya publicado.</p> <p>El sistema borra de la base de datos la receta y la imagen que se almacenan en una carpeta del servidor.</p>

Caso de uso	Publicar recetas
--------------------	-------------------------

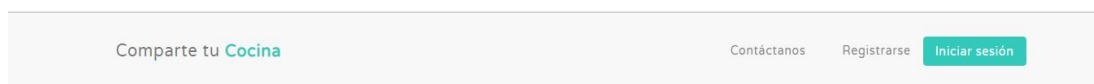
Actor	Usuario registrado
Descripción	Permite a los usuarios crear nuevas recetas
Secuencia del caso	<p>El usuario solicitará crear una receta</p> <p>Se le mostrará un formulario</p> <p>Se comprobará que los datos sean los correctos, que no estén vacíos, que el formato de la imagen sea el permitido y que el tamaño sea el adecuado.</p> <p>Tras comprobar que los datos estén correctos se publicará la receta</p> <p>Se le enviará a la vista que muestra la lista de las recetas publicadas por los usuarios.</p>

7- Mapa de navegación .



Cuando el usuario acceda al sitio web, que está alojado en el dominio : <http://www.compartetureceta.byethost7.com/web/index.php>

Se le mostrará una página principal con un menú con las siguientes opciones: Contáctanos, Registrarse, Iniciar sesión.



Si el usuario da click en Contáctanos se le mostrará el un formulario con el que podrá enviar un mensaje a los administradores del sitio web.

Si da click en Registrarse se le mostrará un formulario para que ingrese sus datos personales.

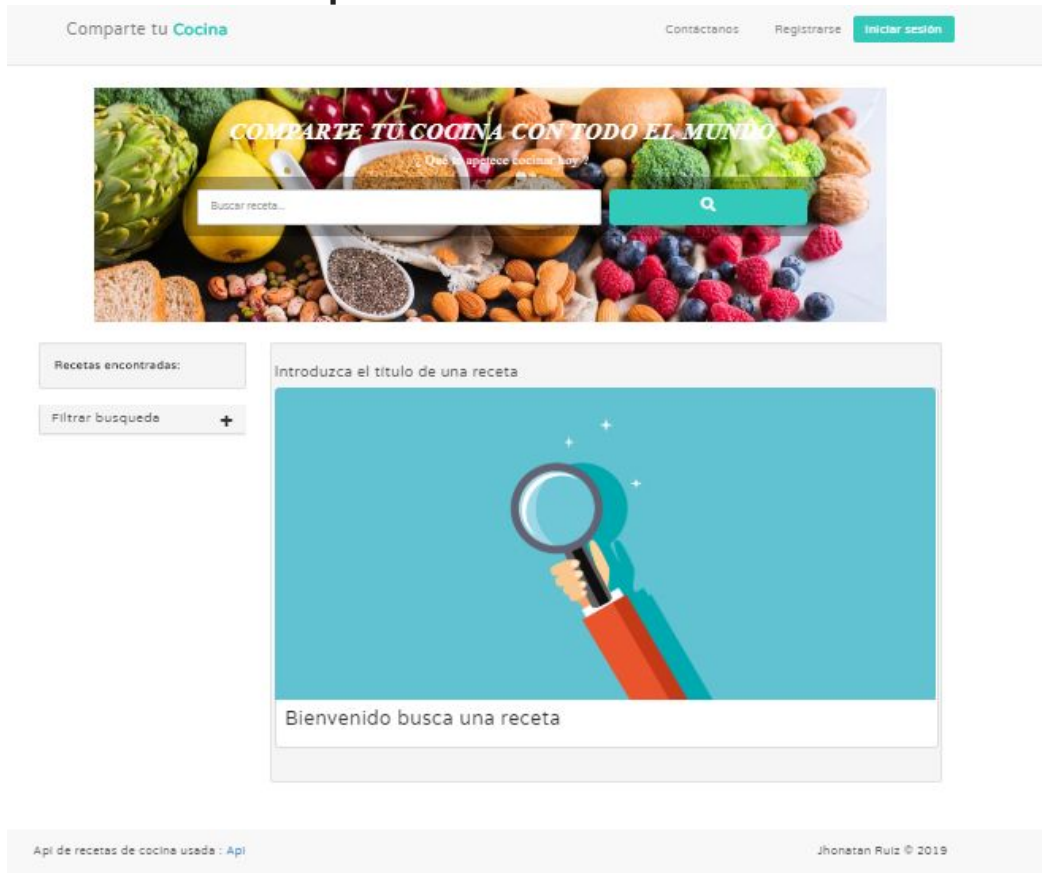
Y si pulsa el botón de iniciar sesión se le mostrará un formulario con el que podrá acceder al sitio web y a las distintas funcionalidades que esté tiene.

Cuando el usuario esté registrado y haya iniciado sesión se le mostrará un menú para que pueda ver las recetas de los usuarios que han publicado y publicar una nueva receta.

8- Funcionalidades y características

A continuación se expondrán las diferentes vistas que tiene el proyecto, cómo se ha implementado y las funcionalidades que tiene cada una.

Ventana Principal



El usuario anónimo al acceder a la aplicación verá esta página de inicio. Que le muestra una barra de búsqueda en la que podrá introducir el nombre de la receta que quiere buscar, un menú al que podrá acceder a la página de contacto, y a la página para registrarse. Y un botón para que acceda a la aplicación. Dicho botón muestra esta ventana.



Con esta ventana el usuario podrá acceder a la aplicación, ingresando su email y contraseña con la que se registro.

Esta funcionalidad comprueba que exista ese email y esa contraseña en la base de datos.

Esta implementación se hace de la siguiente manera.

El index de este proyecto recibe como dato de entrada, al pulsarse el botón de login, un parámetro llamado login, para posteriormente llamar al controlador.

```
$map = array(
    'inicio' => array('controlador' => 'Controlador','action'
=>'inicio'),
    'formContacto'=> array('controlador' => 'Controlador','action'
=>'mostrarFormContacto'),
    'formRegistro'=> array('controlador' => 'Controlador','action'
=>'mostrarFormRegistro'),
    'login' => array('controlador' => 'Controlador','action'
=>'login'),
    'cerrarSesion' => array('controlador' => 'Controlador','action'
=>'salir'),
    'buscarReceta' => array('controlador' => 'Controlador','action'
=>'getRecetaApi'),
    'publicarReceta' => array('controlador' =>
'Controlador','action' =>'formPublicarReceta'),
    'recetasUsuarios' => array('controlador' =>
'Controlador','action' =>'getRecetaUser'),
    'eliminarReceta' => array('controlador' =>
'Controlador','action' =>'eliminarReceta'),
);
```

El controlador ejecuta la siguiente función que llama al modelo.

```

public function login(){
    |
    $entrar = false;
    // Se ha enviado la identificación
    if (isset($_POST['logEmail']) && isset($_POST['logPassword']))
    {
        $exito = $this->ChequearAcceso($_POST['logEmail'],$_POST['logPassword']);
        $entrar = $exito;
        if ( !$exito ){
            $params['mensaje_error'] = ' Usuario y contraseña no validos';
        }
    }
    // Si el chequeo ha sido correcto entrar en la página de inicio
    if ($entrar ){
        $params['mensaje_ok2'] = '<script> alert(Usuario y contraseña
        validos); </script>';
        header('Location:index.php');
        $this->inicio();
    }else{
        header('Location:index.php');
    }
}

```

El modelo comprueba que el usuario exista en la base de datos con ese email y esa contraseña que se ha introducido, de la siguiente manera.

```

public function login($mail, $pass){
    $mail = htmlspecialchars($mail);
    $pass = htmlspecialchars($pass);
    $sql = "select * from usuarios where email = :email and password =
    :pass ";
    try{
        $consulta = $this->db->prepare($sql);
        $consulta->bindParam('email',$mail, PDO::PARAM_STR);
        $consulta->bindParam('pass',$pass, PDO::PARAM_STR);
        $consulta->execute();
        $row = $consulta->fetch(PDO::FETCH_ASSOC);
    }catch(PDOException $e){
        die("Error al ejecutar la orden :". $e->getMessage());
    }
    return $row;
}

```

Cabe destacar que en este proyecto no está implementado la opción de acceder con la cuenta de facebook o twitter, está pensado para una futura mejora.

Si el usuario pulsa en **Contáctanos** se le mostrará el siguiente formulario con el que podrá ponerse en contacto con los administradores del sitio web.

CONTACTA CON NOSOTROS










Escribe tu mensaje

Submit

Si pulsa en **Registrarse** se le mostrará un formulario en el que podrá introducir sus datos personales para poder crearse una cuenta en el sitio web para acceder a las diferentes funcionalidades de la aplicación.

Comparte tu **Cocina**

Contáctanos Registrarse **Iniciar sesión**

 **REGISTRO**

Nombre y apellido:

Nombre de usuario:

Correo electrónico:

Contraseña:

País:

Ciudad:

☐ Estoy de acuerdo con los términos y condiciones de uso

REGISTRARSE

Cuando el usuario introduzca todos los datos y se compruebe que el nombre de usuario no está en la base de datos, que se hace de la siguiente manera:


```

public function comprobarUsuarioExistente($nomUser) {
    $id = htmlspecialchars($nomUser);
    $sql="select * from usuarios where nombreUser = :user";
    try{
        $consulta = $this->db->prepare($sql);
        $consulta ->execute([':user'=>$nomUser]);
    }catch(PDOException $e){
        die("Error al ejecutar la consulta :". $e->getMessage());
    }
    if($consulta ->rowCount() >0){
        return true;
    }else{
        return false;
    }
}

```

Cuando se compruebe que el nombre de usuario no está en la base de datos, el usuario se podrá registrar.

El registro de usuario se hace de la siguiente manera.

Cuando el usuario pulsa en Registrarse se le pasa al index un parámetro llamado from Registro que luego llama al controlador que ejecuta una función del modelo, y comprueba que los datos sean válidos. Si todos los datos son correctos se ingresan los datos del usuario en la base de datos.

Index

```

$map = array(
    'inicio' => array('controlador' => 'Controlador','action' =>
    'inicio'),
    'formContacto'=> array('controlador' => 'Controlador','action' =>
    'mostrarFormContacto'),
    'formRegistro'=> array('controlador' => 'Controlador','action' =>
    'mostrarFormRegistro'),
    'login' => array('controlador' => 'Controlador','action' =>'login'),
    'cerrarSesion' => array('controlador' => 'Controlador','action' =>
    'salir'),
    'buscarReceta' => array('controlador' => 'Controlador','action' =>
    'getRecetaApi'),
    'publicarReceta' => array('controlador' => 'Controlador','action'
    =>'formPublicarReceta'),
    'recetasUsuarios' => array('controlador' => 'Controlador','action'
    =>'getRecetaUser'),
    'eliminarReceta' => array('controlador' => 'Controlador','action'
    =>'eliminarReceta'),
);

```

Controlador

```
public function mostrarFormRegistro(){
    $n= ModeloSingleton::singleton();
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        // comprobar campos formulario
        if ($n->validarDatos($_POST['r_nombreUser'], $_POST['r_username'],
            $_POST['r_email'], $_POST['r_pass'], $_POST['r_pais'],
            $_POST['r_city'])) {
            if (!$n->comprobarUsuarioExistente($_POST['r_username'])) {
                $n->registrarUsuarios($_POST['r_nombreUser'], $_POST['r_username'],
                    $_POST['r_email'], $_POST['r_pass'], $_POST['r_pais'],
                    $_POST['r_city']);
                $params['mensajeok'] = '<script type="text/javascript">alert("Registrado
                    correctamente !")</script>';
                $_SESSION['userName']=$_POST['r_username'];
                $_SESSION['tiempo']=time();
                //$_SESSION['expire'] = $_SESSION['tiempo'] + (5* 60);
                header('Location:index.php');
                $this->inicio();
            }else{
                $params['mensaje'] = '<script type="text/javascript">alert("Ese nombre de
                    usuario ya existe !")</script>';
                $params['mensajeError']="<div class='container'>El nombre de usuario
                    ($_POST['r_username']) ya existe en la base de datos </div> ";
            }
        } else {
            $n->validarDatos($_POST['r_nombreUser'], $_POST['r_username'],
                $_POST['r_email'], $_POST['r_pass'], $_POST['r_pais'],
                $_POST['r_city']);
            $params['mensaje'] = 'No se ha podido registrar';
        }
    }
    require __DIR__ . '/vistas/formRegistro.php';
}
```

Modelo

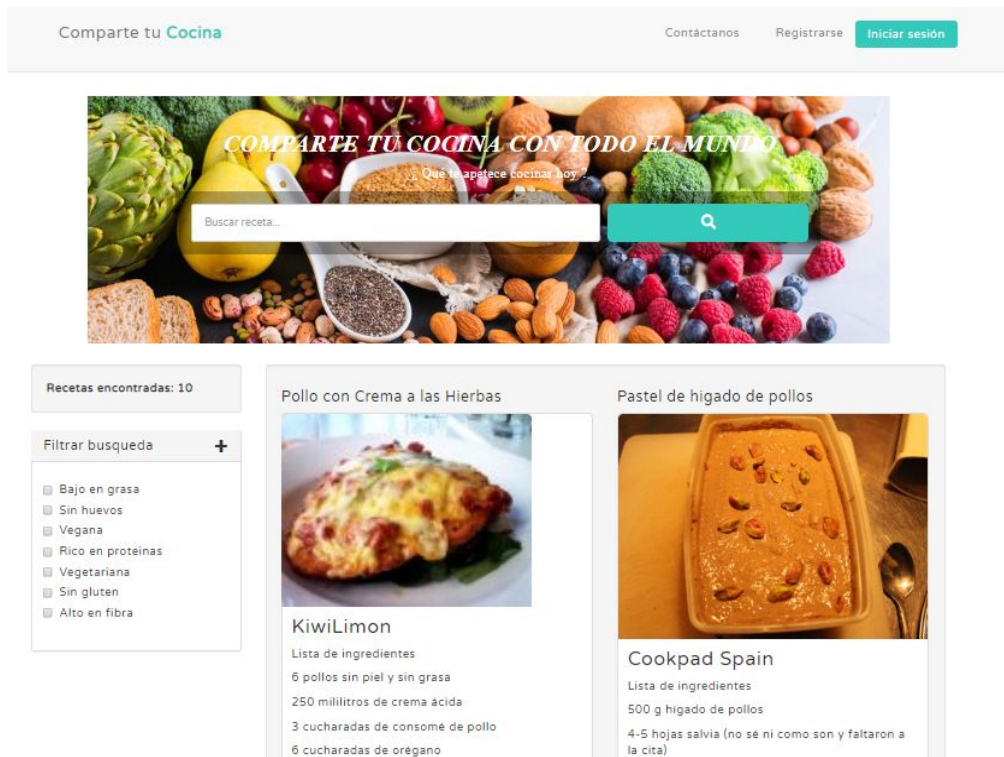
```
public function registrarUsuarios($nomApe, $nomUser,$email,$password,$pais,$ciudad){
    $nomApe = htmlspecialchars($nomApe);
    $nomUser = htmlspecialchars($nomUser);
    $email = htmlspecialchars($email);
    $password = htmlspecialchars($password);
    $pais = htmlspecialchars($pais);
    $ciudad = htmlspecialchars($ciudad);

    $sql= "insert into usuarios (nombreApe, nombreUser, email, password, pais, ciudad)
        values(?, ?, ?, ?, ?, ?)";

    try{
        $consulta = $this->db->prepare($sql);

        $consulta->bindParam(1,$nomApe, PDO::PARAM_STR);
        $consulta->bindParam(2,$nomUser, PDO::PARAM_STR);
        $consulta->bindParam(3,$email,PDO::PARAM_STR);
        $consulta->bindParam(4,$password,PDO::PARAM_STR);
        $consulta->bindParam(5,$pais, PDO::PARAM_STR);
        $consulta->bindParam(6,$ciudad, PDO::PARAM_STR);
        $consulta->execute();
    }catch(PDOException $e){
        die("Error al ejecutar la consulta : " . $e->getMessage());
    }
    return $consulta;
}
```

Si realiza una consulta se le mostrará la siguiente pantalla.



Esta ventana muestra los datos obtenidos del api de alimentos usada en este proyecto. La implementación de esta funcionalidad se hace de la siguiente manera:

Se envía un parámetro al index para que llame al controlador. Y el controlador llama a una función del modelo, que se encarga de obtener los datos del api de alimentos, además el controlador muestra dichos datos de una forma bonita al usuario.

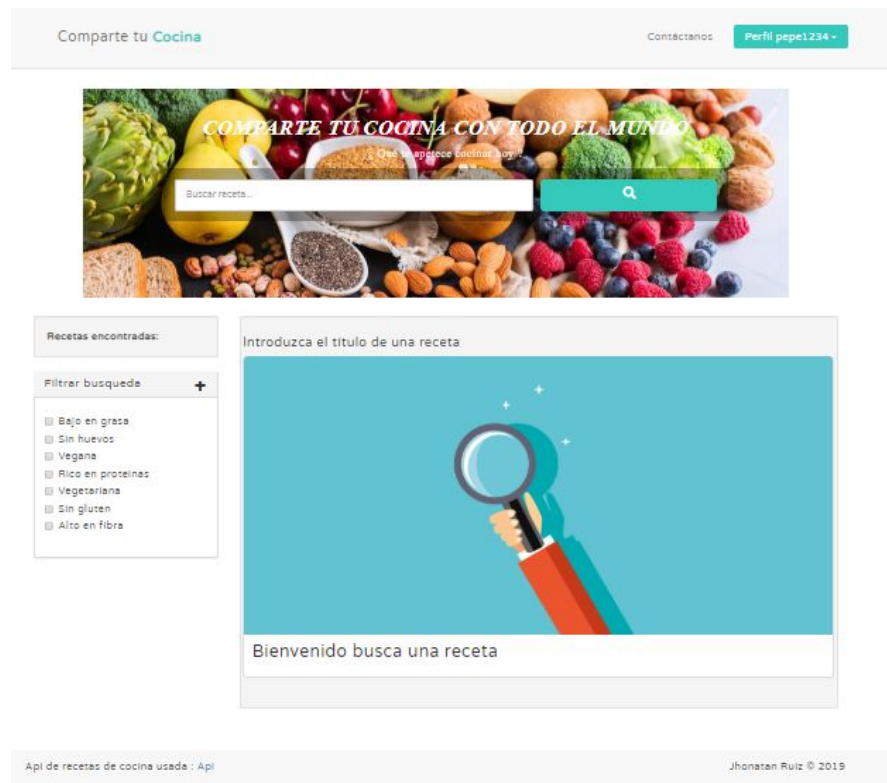
```
public function getRecetaApi() {
    $n= ModeloSingleton::singleton();
    if (isset($_GET['recetaBuscar'])) {
        $resultado = $n->getReceta($_GET['recetaBuscar']);
        require __DIR__ . '/vistas/mostrarrecetas.php';
    } else {
        header('Location:index.php');
        $this->inicio();
    }
}
```

```
public function getReceta($recetaBuscar){
    $URLAPI = "https://test-es.edamam.com/search?
    q=".urlencode($recetaBuscar)."&app_id=$APPID&app_key=$APIKEY";
    $urlapiJson=file_get_contents($URLAPI);
    $urlapi_array= json_decode($urlapiJson,true);

    //$resultado=$urlapi_array['count']; numero de recetas
    //echo "$resultado";
    //$resultado=$urlapi_array['to'] return
    $recetas=$urlapi_array['hits'];
    return $resultado = $urlapi_array;
}
```

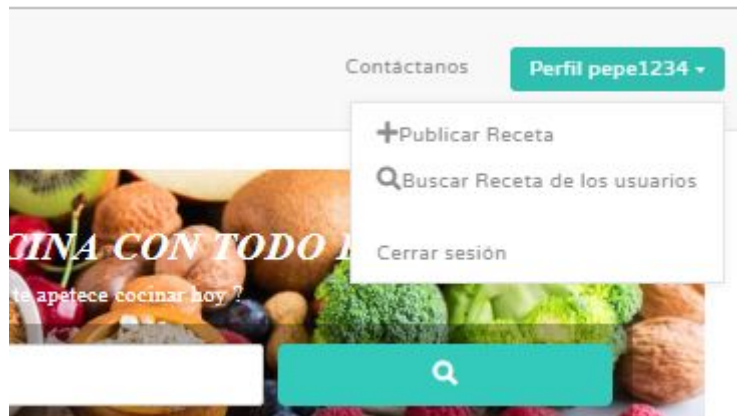
Los datos de la receta que nos muestra son: El título de la receta, la imagen de la receta, el sitio web en el que está publicada, y los ingredientes de la receta. Si le da click el usuario a la imagen de la receta le llevará al sitio web donde está publicada y podrá tener más detalles sobre esa receta.

Cuando el usuario inicia sesión se le mostrará la siguiente ventana.



La funcionalidad de inicio de sesión se realiza de la siguiente manera, el controlador llama a una función del modelo que se encarga de comprobar que dicho usuario esté en la base de datos. Y si el usuario está en la base de datos se mostrará la ventana anterior.

En esta ventana se le muestra el nombre del usuario y si pulsa en el botón con su nombre de usuario se le mostrará un menú con las siguientes opciones:



En este menú el usuario podrá publicar nuevas recetas o ver las todas recetas que los otros usuarios han publicado.

Cuando pulse en publicar nuevas recetas se le mostrará un formulario en el que podrá ingresar los datos de una nueva receta.

A screenshot of a web application showing a form to publish a new recipe. The form is titled 'PUBLICAR NUEVA RECETA' with a pencil icon. It contains several input fields: 'Título Receta' (with placeholder 'Título de la receta'), 'Imagen Receta' (with a 'Seleccionar archivo' button and placeholder 'No se eligió archivo'), 'Tiempo de preparación', 'Ingredientes de la receta', and 'Pasos a seguir'. At the bottom of the form are two buttons: 'PUBLICAR RECETA' and 'CANCELAR'. The page header includes 'Comparte tu Cocina', 'Contáctanos', and 'Perfil pepe1234'. The footer contains the text 'Api de recetas de cocina usada : Api' and 'Jhonatan Ruiz © 2019'.

Tras introducir los datos de la receta y tras comprobar que los datos sean los correctos, que la imagen sea menor que 1 mb, que el formato sea png, jpg o jpeg, y que ningún campo esté vacío, se publicará la nueva receta creada.

La implementación de esta funcionalidad se realiza de la siguiente manera. Primero se llama al controlador quien es el encargado de llamar al modelo y de pasarlo los datos de la receta para que luego se introduzcan en la base de datos. El controlador además se encarga de comprobar que dicha imagen pese menos de 1mb, de almacenarla en una carpeta del servidor con un nombre aleatorio y de comprobar que el formato de la imagen sea el adecuado (JPG,PNG, JPEG).

Controlador

```
public function formPublicarReceta(){
    $n= ModeloSingleton::singleton();
    if(isset($_POST['btnPublicarReceta']))){
        // $tituloR, $imgR, $tiempoPreR, $ingredientesR, $pasosR
        $validation = $n->validarDatosReceta($_POST['r_TituloR'], $_FILES['r_ImgR'], $_POST['r_tiempoR'], $_POST['r_ingreR'], $_POST['r_pasosR']);

        $imgFile = $_FILES['r_ImgR']['name'];
        $tmp_dir = $_FILES['r_ImgR']['tmp_name'];
        $imgSize = $_FILES['r_ImgR']['size'];
        if($validation == null){
            //Cargamos directorio para guardar imagenes en DB
            $carga_dirImag = __DIR__ . '/imgDB/';
            //obtenemos la extension img
            $imgExt = strtolower(pathinfo($imgFile, PATHINFO_EXTENSION));
            //comprobamos que se acepten solo jpg png jpeg
            $valid_extensions = array('jpeg', 'jpg', 'png');
            //renombramos la imagen cargada
            $userpic = rand(1000,1000000).".". $imgExt;
            //comprobamos el formato
            if(in_array($imgExt, $valid_extensions)){
                // comprobamos el tamaño
                if($imgSize < 1000000){
                    move_uploaded_file($tmp_dir, $carga_dirImag.$userpic);
                    $n->publicarReceta($_POST['r_TituloR'], $userpic, $_POST['r_tiempoR'], $_POST['r_ingreR'], $_POST['r_pasosR'], $_SESSION['userName']);
                }else{
                    echo '<script type="text/javascript">alert("Su archivo es muy grande!")</script>';
                }
            }else{
                echo '<script type="text/javascript">alert("Solo archivos JPG, JPEG, PNG & GIF son permitidos!")</script>';
            }
        }else{
            echo '<script type="text/javascript">alert("Compruebe que haya introducido todos los datos!")</script>';
        }
    }
    if(isset($_params['mensajeErrorReceta'])){
        // $tituloR, $imgR, $tiempoPreR, $ingredientesR, $pasosR, $usuario
        if($n){
            echo '<script type="text/javascript">alert("Receta insertada correctamente!")</script>';
            header('Refresh:3; url=index.php?ctl=recetasUsuarios');
        }
    }
}
require __DIR__ . '/vistas/formPublicarReceta.php';
}
```

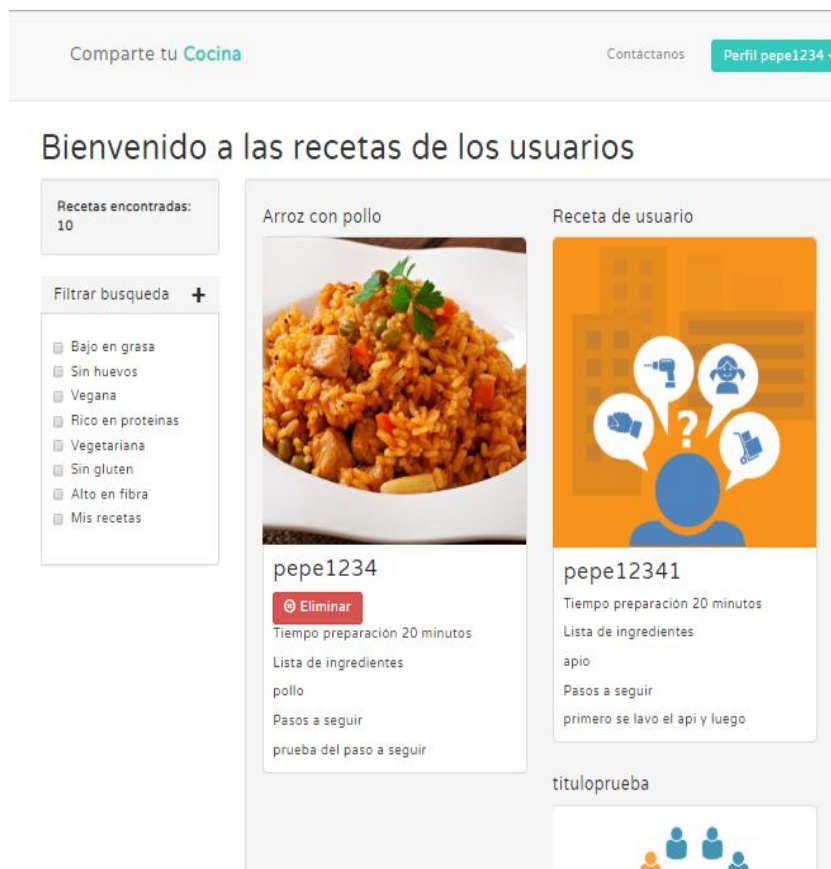
Modelo

```
public function publicarReceta($tituloR,$imgR,$tiempoPreR,$ingredientesR,$pasosR,$usuario){
    $tituloR = htmlspecialchars($tituloR);
    $tiempoPreR = htmlspecialchars($tiempoPreR);
    $ingredientesR = htmlspecialchars($ingredientesR);
    $pasosR = htmlspecialchars($pasosR);
    $sql= "insert into recetas (titulo, imgReceta, tiempoPreparacion, ingredientes,
    pasos,usuario) values(?,?,?,?,?,?)";
    try{
        $consulta = $this->db->prepare($sql);

        $consulta->bindParam(1,$tituloR);
        $consulta->bindParam(2,$imgR);
        $consulta->bindParam(3,$tiempoPreR);
        $consulta->bindParam(4,$ingredientesR);
        $consulta->bindParam(5,$pasosR);
        $consulta->bindParam(6,$usuario);

        $consulta->execute();
    }catch(PDOException $e){
        die("Error al ejecutar la consulta :". $e->getMessage());
    }
    return $consulta;
}
```

Una vez que todo haya ocurrido con éxito nos enviará al listado de todas las recetas que los usuarios han publicado.



Si el usuario que publicó la receta está logado le aparecerá el botón para eliminarla.

La eliminación de la receta se hace la siguiente manera:

Se manda al controlador cuando se da click en el botón de eliminar el id de la receta y el modelo ejecutar la función adecuada para eliminarla

Controlador

```
public function eliminarReceta(){
    if(isset($_GET['idRecetaDelete'])){
        $n= ModeloSingleton::singleton();
        $idReceta=$_GET['idRecetaDelete'];
        $receta = $n->getNombreImgBorarIMGDB($idReceta);
        if($receta['recetaId'] == $idReceta){
            $nombreRecetaAeliminarIMGDB=$receta['imgReceta'];
            echo $nombreRecetaAeliminarIMGDB;
            unlink( __DIR__ . "/imgDB/".$nombreRecetaAeliminarIMGDB);
        }
        $n->eliminarRecetaUser($idReceta);
    }
    if($eliminado = $n->eliminarRecetaUser($idReceta)){
        header("Refresh:3; url=index.php?ctl=recetasUsuarios");
    }
}
```

Modelo

```
public function eliminarRecetaUser($idReceta){
    $idReceta = htmlspecialchars($idReceta);

    $sqlEliminaReceta='delete from recetas where recetaId= :id';
    try{
        $consulta = $this->db->prepare($sqlEliminaReceta);
        $consulta->bindParam('id',$idReceta,PDO::PARAM_INT);
        $consulta->execute();
    }catch(PDOException $e){
        die("Error al ejecutar la orden : " . $e->getMessage());
    }
    return $consulta;
}
```


9- Conclusiones y propuestas

Conclusiones

Se ha desarrollado una aplicación web funcional y accesible desde cualquier dispositivo móvil, a través de un navegador. Dicho proyecto da a conocer a gente inexpertas en el mundo de la cocina, nuevas recetas. Y pone a disposición de los usuarios registrados una herramienta para difundir sus recetas de cocinas, con sus trucos culinarios que las hacen más ricas.

Además, se ha conseguido hacer un sitio web responsive, con algunos fallos de diseño que se mejorarán en futuras actualizaciones.

Como resultados del proyecto realizado podemos destacar los siguientes puntos:

Se ha conseguido fomentar la forma de cocinar de diferentes usuarios.

Se ha usado una api de alimentos que pone a disposición de todos más de 700.000 recetas de cocina publicadas en diferentes sitios web.

Se ha desarrollado una lógica capaz de soportar las funcionalidades del sitio web.

Se han diseñado interfaces sencillas y amigables adaptables a distintos dispositivos móviles.

Se ha desplegado el proyecto en un hosting gratuito que garantiza su funcionamiento; y el acceso de cualquier usuarios a la aplicación mediante cualquier navegador bajo el dominio: <http://www.compartetureceta.byethost7.com/web/>

Futuras mejoras

Corregir fallos de diseño, por ejemplo, las imágenes de la recetas tengan un tamaño fijo.

Implementar funciones para filtrar las recetas de cocina según, la cantidad de proteínas, si no tienen huevos, si son vegetarianas, o vengas, etc...

Implementar un sistema de red social, en el que se pueden versionar una receta de cocina, se puedan dejar comentarios en las recetas publicadas, y valorar las recetas.

Valoración personal

Personalmente el proyecto no tiene todas las funcionalidades que me hubieran gustado hacer, pero por falta de motivación, tiempo, y problemas personales, no se han podido desarrollar.

Pero aun así las distintas funcionalidades que tiene esta aplicación me han servido para poner en práctica los conocimientos adquiridos durante todo este curso y para adquirir nuevos que he ido investigando durante el desarrollo de este proyecto, también me ha servido para refrescar conceptos olvidados.

Por otra parte quisiera destacar que el desarrollo de un proyecto implica profundizar en diferentes áreas de conocimiento, estar al día de las nuevas herramientas que vayan surgiendo.

10- Bibliografía y referencias

Se han empleado los apuntes dados durante todo el año

Para la parte de diseño se ha empleado bootstrap 3.3.7

Para la interacción del navegador (la página web), con el usuario se ha usado JQuery

La api de recetas usada es <https://www.edamam.com/>