



Escuela Técnica Superior de Ingeniería  
Informática y Telecomunicaciones



Tecnologías Web  
Curso 2019 - 2020

---

## Proyecto Web : Recetario

---

Jesús Ruiz Castellano - 76439001L  
Víctor Peralta Cámara - 76438074N

## Contents

<b>1</b>	<b>Usuarios del sistema</b>	<b>2</b>
<b>2</b>	<b>Fichero de restauración de la BBDD</b>	<b>2</b>
<b>3</b>	<b>Diseño de la BBDD</b>	<b>3</b>
3.1	Modelo Entidad-Relación . . . . .	3
3.2	Modelo Relacional . . . . .	3
<b>4</b>	<b>Explicaciones técnicas</b>	<b>4</b>
<b>5</b>	<b>Ítems Opcionales</b>	<b>6</b>

## 1 Usuarios del sistema

### 1. Rol - Administradores :

- **Usuario :** victor@gmail.com  
**Clave :** 1234
- **Usuario :** jesus@gmail.com  
**Clave :** 1234

### 2. Rol - Colaborador :

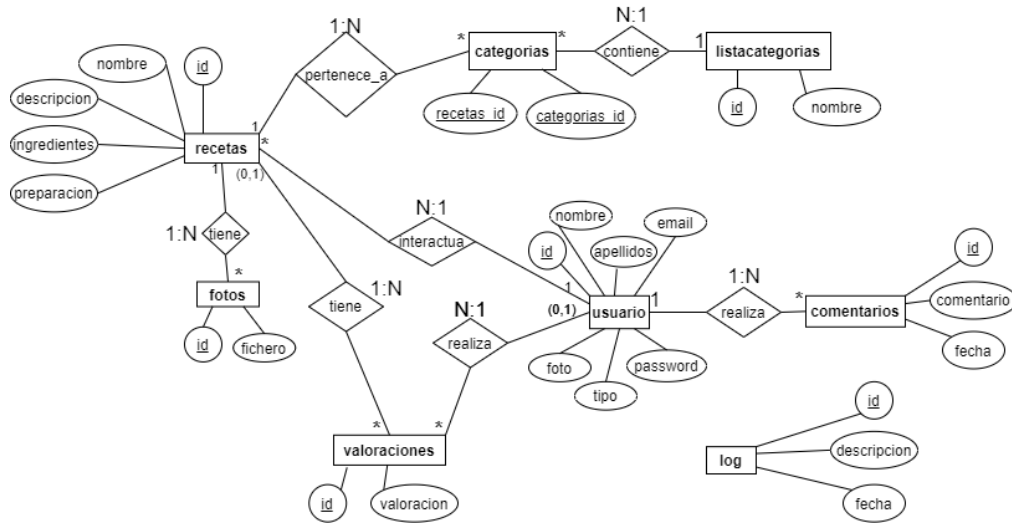
- **Usuario :** colaborador@gmail.com  
**Clave :** 1234

## 2 Fichero de restauración de la BBDD

Este fichero se encuentra en el directorio *public\_html/proyecto/export*, con el nombre *tw.sql*.

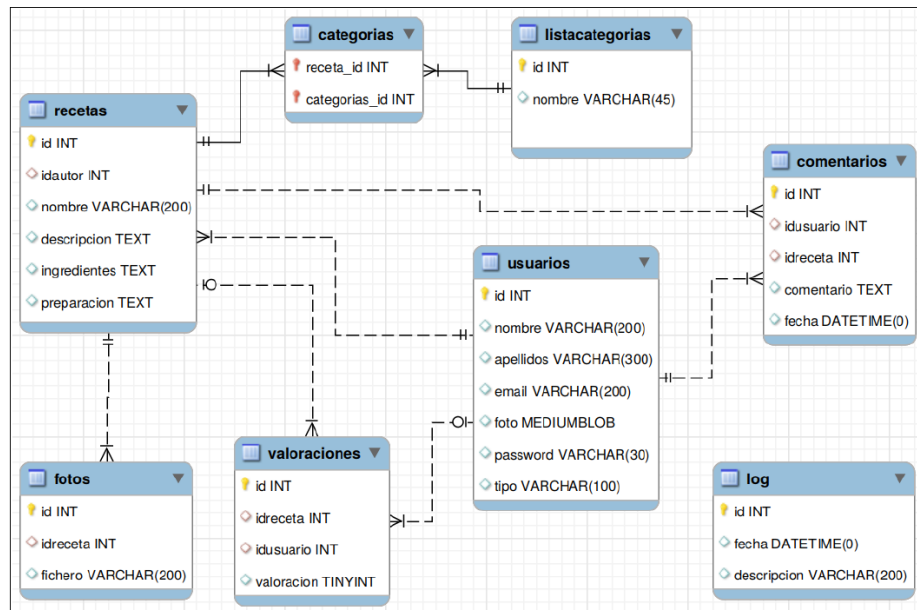
### 3 Diseño de la BBDD

#### 3.1 Modelo Entidad-Relación



#### 3.2 Modelo Relacional

Del esquema anterior, y sirviéndonos del guión explicativo del proyecto, se obtiene el siguiente **Modelo Relacional** :



## 4 Explicaciones técnicas

Todo el proyecto ha sido desarrollado para cumplir los requisitos que supone **mantener el patrón arquitectónico Modelo-Vista-Controlador**.

Cada sección tiene su propio directorio. Las pautas principales son las siguientes:

**Vista:** se incluyen todos los ficheros PHP que generan HTML de forma dinámica. Además, incluye el fichero de estilo css, y todos los ficheros Javascript (los cuales se encargan de modificar el diseño de la web dinámicamente así como realizar peticiones AJAX).

Este directorio también incluye todas las imágenes de la web, tanto las almacenadas de forma estática, como las que se generan al

**Controlador:** se encarga de conectar la vista con el modelo (del cual hablaremos más adelante). Hemos tratado de incluir en el controlador todas las operaciones con respecto a las **peticiones de tipo POST y GET**. De esta forma, se abstraen las peticiones HTTP del modelo. El cuál, recibe las variables que necesita como parámetros de cada función, sin tener en cuenta cómo se hacen las peticiones HTTP al servidor.

Por último, el controlador incluye el fichero más importante del proyecto: *index.php*. Hemos intentado mantener este fichero lo más libre posible de operaciones sobre datos, utilizándolo sólo y exclusivamente para realizar el enrutamiento de la web. Esto permite dar una idea rápida de cómo se organiza la información en el servicio web.

**Modelo:** en este directorio se encuentran todas las funciones que involucran peticiones a la base de datos. Además, el acceso a la base de datos se ha implementado utilizando una clase con el patrón **Singleton**. De esta forma, sólo existe una conexión simultánea para todo el servicio, en vez de crear conexiones cada vez que se quiera realizar una petición.

Por último, cabe mencionar que para algunas peticiones se han utilizado sentencias preparadas para mejorar la seguridad del servicio.

Para el sistema de puntuación, como se pide en su apartado, se muestra una media de todas las puntuaciones recibidas para cada receta. Sin embargo, se utiliza la misma interfaz para puntuar. Por tanto, **aunque se muestre cierta puntuación en una receta, se puede seleccionar el número de estrellas que queramos para enviar la puntuación del usuario conectado. La próxima vez que se muestre la receta se mostrará la media actualizada.**

La mayoría de formularios incluyen los siguientes pasos para enviarlos:

1. Edición
2. Confirmación
3. Información

Estos pasos han sido implementados mediante **Javascript**, hemos pensado que sería una operación indicada para realizar en el cliente, en vez de cargar el servidor. Para ello se desactivan los campos la primera vez que se selecciona el botón para enviar el formulario y se envía el formulario, finalmente, la segunda vez que se selecciona.

Para los formularios sencillos, cómo añadir o editar comentarios, se ha omitido este paso.

Los **widgets** incluidos en la barra lateral son totalmente funcionales, mostrando en tiempo real las 3 recetas más valoradas (con mayor media), más comentadas (con mayor número de comentarios) y la cantidad de recetas en el sistema.

Al añadir una nueva receta al sistema, se ha hecho uso de **Javascript** para mostrar vistas previas de las imágenes incluidas en la receta (tanto la imagen principal como las imágenes que representan los pasos de la receta).

Por último, cabe mencionar que **las categorías son totalmente dinámicas**. Pudiendo añadir y eliminar categorías al sistema, y utilizarlas tanto al añadir recetas cómo al realizar búsquedas o editar recetas.

## 5 Ítems Opcionales

Los ítems opcionales que hemos desarrollado han sido los siguientes :

- **Opcional 1:** Si queremos disponer de múltiples categorías por receta, la solución indicada antes no es la mejor. Podemos pensar en tener una tabla en la que se guardan todas las posibles categorías. Esta tabla estará relacionada con la de recetas. Cuando el usuario añada o edite una receta le aparecerán checkboxes para marcar las categorías a las que pertenece. Esta lógica implica que debe disponer de formularios para editar esa tabla de categorías (añadir, editar, borrar). Esa tabla solo será modificable por usuarios administradores.
- **Opcional 2:** Puede añadir a la aplicación la posibilidad de que los usuarios puntúen las recetas con un valor entre 1 y 5. En este caso, cada usuario registrado puede valorar solo una vez una misma receta. Por tanto, deberán almacenarse todas las valoraciones individuales aunque finalmente lo que se muestre a los usuarios de la página web será la media de dichas valoraciones. Los visitantes sin identificar no podrán valorar las recetas para evitar que puedan puntuar muchas veces una misma receta falseando los datos.
- **Opcional 4:** El inconveniente de la solución propuesta para subir fotografías es que cada vez que se añade una nueva, esta se sube al servidor y este genera un nuevo formulario de edición que permitirá añadir la siguiente. Esto genera un tráfico alto de datos (cada vez que se presenta el formulario se incluyen las fotografías ya almacenadas por si hay que borrarlas) y no es la forma más cómoda para el usuario final. Este es un punto en el que el uso de AJAX y JavaScript mejorarían sustancialmente la usabilidad del sistema. Opcionalmente puede optar por implementar una solución basada en tecnologías de actualización dinámica de la web.