



# Proyecto Final de FP

## Desarrollo de una Aplicación Web para un Mercado Municipal

El Mercado

**Autora:**

Cintia García Ruiz

**Curso:**

2º Desarrollo de Aplicaciones Web

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>3</b>
Objetivos	3
Requisitos	3
Elección de la aplicación	5
<b>DISEÑO SOLUCIÓN</b>	<b>6</b>
Metodología y Entorno de Trabajo	6
Lenguajes Softwares y Tecnologías	7
2.1 Backend	7
2.2 Front-end	7
BBDD	8
3.1. Modelo E/R	8
3.2. Tablas y columnas	9
Despliegue	11
4.1 Despliegue Local	11
4.2 Despliegue AWS	11
<b>DESARROLLO E IMPLEMENTACIÓN</b>	<b>12</b>
Parte global	12
Cliente	13
Vendedor	13
Administrador	13
<b>DESPLIEGUE</b>	<b>14</b>
4.1 Despliegue Local	14
4.2 Despliegue AWS	14
<b>CONCLUSIONES</b>	<b>14</b>
6.2. Validación de requisitos	14
6.3. Evolutivos	14
Gráficas o KPIs para administradores y vendedores.	14
Integración con pasarelas de pago.	14
Multimercado	14
Multidioma	14
<b>BIBLIOGRAFÍA</b>	<b>15</b>

# 1. INTRODUCCIÓN

## 1. Objetivos

El objetivo de este documento es describir la memoria del proyecto final del curso.... En este proyecto, se propone una aplicación para un mercado online que pondrá en contacto tanto a los comerciantes como a los usuarios del mercado.

Los consumidores podrán encontrar sus puestos de confianza (frutería, pescadería, carnicería,...) de forma online, sin necesidad de desplazarse físicamente al mercado. Por otra parte, los responsables de los puestos, dispondrán de una nueva alternativa para vender sus producto o incluso llegar a un nuevo tipo de público al que no pueden vender sus productos actualmente.

Por tanto, el objetivo del proyecto es diseñar y desarrollar una aplicación que cumpla la funcionalidad anteriormente indicada tomando como referencia los conocimientos adquiridos durante el curso en las distintas asignaturas y cumpliendo los requisitos establecidos en cada una de ellas.

## 2. Requisitos

Los requisitos que deben cumplir la aplicación son los siguientes.

- Desarrollo
  - Patrón de diseño MVC.
  - Tecnologías o software específico
    - Javascript, ES6 y JQuery.
    - CSS3.
    - HTML5.
    - Angular, Vue o React.
    - INKSCAPE o similar.
    - Si se utiliza PHP podrá optarse por Laravel , usar frameworks como Vue.js , y para la comunicación asíncronas en background Axios.
  - Funcionalidades:
    - Client Form Validation
    - Ajax Pagination (incremental or Page numbers)
    - Autocomplete search

- CRUD AJAX and JQuery UI
  - Delete row with fade out.
  - Add register with modal form and list fade in.
  - Galería de imágenes.
  - Banner.
  - El diseño de la interfaz deberá ser “Responsive” utilizando Media Queries, FlexBox y Grid layout.
  - Utilizar el preprocesador SASS para estructurar los archivos css en un único main.css con @import a los demás css (colores, cabecera, pie, cuerpo...).
  - Insertar un elemento multimedia de cada tipo: video, sonido, canvas y SVG.
  - Se podrán utilizar las clases Chartjs y Snap.svg.
  - Se debe ofrecer un servicio a los usuarios registrados, para que puedan realizar peticiones con la API para obtener información JSON.
- Base de Datos
  - Uso de BBDD relacionales como MySQL, MariaDB, PostgreSQL...
  - Debe tener 4 tablas como mínimo y entregar el modelo E/R al profesor.
- Despliegue de la aplicación y generación de código.
  - Despliegue en local de la aplicación usando Servidor Apache o Servidor de Aplicaciones.
  - Despliegue en un hosting externo.
  - Documentación del código fuente en inglés.
  - Generar de forma automática la Documentación Técnica del mismo en HTML y PDF (se deben de emplear herramientas del tipo JavaDoc, PhpDocumentor, JsDoc, o similar).
  - Uso de un Sistema de Control de Versiones. Recomendado GitHub
  - Acreditar un seguimiento de versiones. Ramas.
  - Acreditar un despliegue automatizado, si es posible.
  - Acreditar uso de SSH y/o FTP en el desarrollo y/o despliegue, si es posible.

### 3. Elección de la aplicación

He decidido desarrollar esta aplicación porque vi la necesidad de acercar el mercado a todas las personas que a día de hoy realizan sus compras por internet, y además darle un giro a la idea del mercado como tal para actualizarlo. Surgió la idea porque los sábados es el único día que yo puedo ir hacer la compra allí, porque son establecimientos que no abren por la tarde y entonces las personas que están trabajando tienen el acceso más reducido, y ya por no hablar de la mayoría de los jóvenes, que normalmente todo lo compran en las grandes superficies o por internet.

Además en esta situación con el COVID-19 se ha visto reflejado como las ventas online de productos frescos se han incrementado considerablemente, y donde algunos mercados municipales no han podido ofrecer este servicio.

## 2. DISEÑO SOLUCIÓN

### 1. Metodología y Entorno de Trabajo

El diseño de la aplicación se basa en MVC (modelo, vista, controlador) que es un patrón de diseño software que separa una aplicación en tres capas, en donde cada capa maneja un aspecto de la aplicación. Los modelos acceden a la base de datos y recuperan la información, la modifica o la elimina y mediante el controlador se transforma esa información para devolvérsela a la vista.

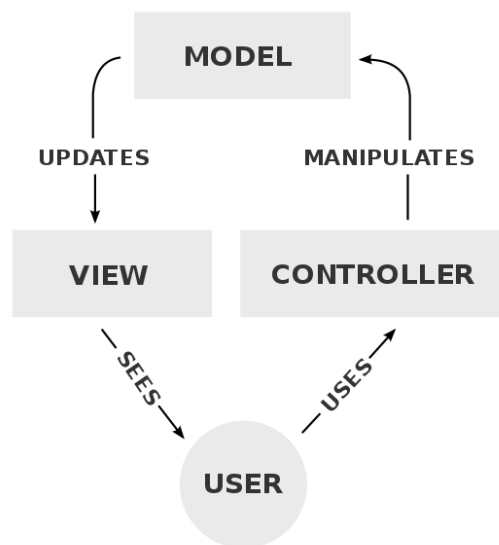


Figura 1. Modelo Vista Controlador [[Wikipedia](#)]

Para la implementación de esta metodología, me ha ayudado el uso del framework de Laravel que implementa este tipo de arquitectura. He decidido utilizar Laravel porque se adapta bien con las necesidades de la aplicación a desarrollar y otra de las razones es porque este framework me proporciona más sencillez de programación en PHP.

Para llevar un control de las versiones y gestión del código he empleado GitHub, lo he usado principalmente por su sencillez y la ventaja de poder compartir código fácilmente. Sobre la metodología empleada para el uso de esta plataforma, he empezado una rama master donde estaban las partes del proyecto que iban funcionando y luego en la rama llamada InicioMigraciones tenía el código sobre el que estaba trabajando.

Como entorno de desarrollo elegido, he utilizado Visual Studio Code, porque me proporciona extensiones relacionadas con el código como Vue.js Extension Pack, Vue VSCode Snippets, Visual Studio IntelliCode, VS Code JavaScript (ES6) snippets...

La base de datos elegida es Mysql porque se integra a la perfección con el framework de laravel, mejor manejo de la conexión con los modelos y como gestor para la base de datos en local utilizo phpMyAdmin.

## 2. Lenguajes Softwares y Tecnologías

Siguiendo la metodología MVC, la aplicación se ha dividido en dos partes diferenciadas, una para el back y otra para el front donde las tecnologías usadas son distintas.

### 2.1 Backend

Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione, conjunto de acciones que pasan en una web pero que no vemos como, por ejemplo, la comunicación con el servidor.

Para esta parte, se han empleado principalmente PHP. Con este lenguaje, se ha desarrollado para toda la lógica de los controladores de la aplicación y los modelos que se utilizan. Laravel

### 2.2 Front-end

Es una especialidad para el desarrollo web, que trabaja la interfaz web y hace que el usuario pueda interactuar con nuestra web. Está orientado a lenguaje de marcas y al lenguaje de programación web de ejecución en equipos clientes, sin necesidad de uso de servidores externos. Casi todo lo que ves en la pantalla cuando accedes a una web es desarrollo frontend.

Los lenguajes empleados para esta parte son:

- **JavaScript** : para funcionalidades tales como verificar Email en el registro, comprobar si existe un DNI, autocompletar buscador...
- **Axios**: para hacer peticiones a la api, conecta el back con el front.
- **HTML 5**: en todas las vistas para la visualización de mi web y todos sus contenidos.
- **BOOTSTRAP Y CSS3**: para todos los estilos de la web para ello he utilizado sass para unificar los estilos.
- **VUE**: como componentes dentro de laravel, para cosas como mostrar la contraseña...

### 3. BBDD

La base de datos utilizada es Mysql porque encaja con el proyecto elegido, es una base de datos relacional de código abierto con un modelo cliente-servidor.

#### 3.1. Modelo E/R

Para comenzar creo el modelo E/R (figura 2) para el modelado de la base de datos. Obtengo 6 tablas en la base de datos y sus atributos correspondientes.

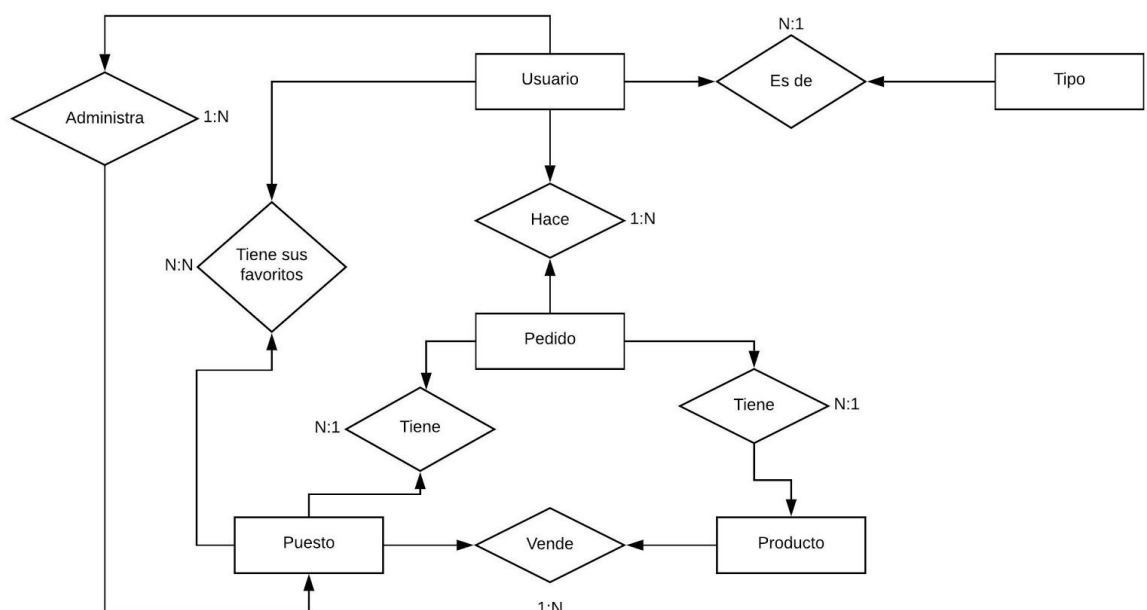


Figura 2. Modelo entidad relación del proyecto.

- **Usuario:** Representa a las personas que se conectan a la aplicación, el objetivo y visión de cada usuario dependerá del rol de cada usuario.
- **Tipo:** Es la diferenciación de los usuarios según su rol: Cliente, Vendedor y Administrador.
- **Pedido:** Petición de un producto realizada por un usuario Cliente a un puesto, que es administrado por un Vendedor de un puesto.
- **Puesto:** Representación de un puesto virtual, donde el proveedor de producto los ofrece y el comprador puede adquirirlos.
- **ProductoPuesto:** Son los alimentos que se venden en el mercado.
- **Favorito:** Representa la relación de los puestos favoritos de los usuarios. Resultado de la relación de mucho a muchos entre usuario y puesto.



### 3.2. Tablas y columnas

#### User

<u>Nombre</u>	<u>Tipo</u>	<u>Descripción</u>
<u>idUsu(pk)</u>	<u>INT</u>	<u>Identificador y clave privada del usuario</u>
<u>nombre</u>	<u>VARCHAR</u>	<u>Nombre del usuario</u>
<u>apellido</u>	<u>VARCHAR</u>	<u>Apellido de la persona a la que pertenece el usuario</u>
<u>dni</u>	<u>VARCHAR</u>	<u>DNI de la persona a la que corresponde el usuario</u>
<u>Email</u>	<u>VARCHAR</u>	<u>Correo electrónico</u>
<u>password</u>	<u>VARCHAR</u>	<u>Contraseña del usuario</u>
<u>telefono</u>	<u>INT</u>	<u>Telefono usuario</u>
<u>fecNac</u>	<u>DATE</u>	<u>Fecha de Nacimiento</u>
<u>idTipo (fk)</u>	<u>INT</u>	<u>id del tipo de usuario</u>
<u>vendedor</u>	<u>BOOLEAN</u>	<u>Representa si quiere ser vendedor</u>
<u>direccion</u>	<u>VARCHAR</u>	<u>Dirección del usuario</u>
<u>tarjeta</u>	<u>BIGINTEGER</u>	<u>Tarjeta de crédito usuario</u>
<u>caducidad</u>	<u>VARCHAR</u>	<u>Fecha caducidad tarjeta</u>
<u>cvc</u>	<u>INT</u>	<u>Cvc de la tarjeta</u>
<u>foto</u>	<u>VARCHAR</u>	<u>Foto usuario</u>
<u>nif</u>	<u>VARCHAR</u>	<u>NIF empresa</u>

### Favorito

<u>Nombre</u>	<u>Tipo</u>	<u>Descripción</u>
<u>idUsu(fk) (pk)</u>	<u>INT</u>	<u>Id del usuario</u>
<u>idPuesto(fk) (pk)</u>	<u>INT</u>	<u>id del puesto</u>

### Puesto

<u>Nombre</u>	<u>Tipo</u>	<u>Descripción</u>
<u>idPuesto (pk)</u>	<u>INT</u>	<u>Identificador y clave privada del puesto</u>
<u>idUsu (fk)</u>	<u>INT</u>	<u>Id del usuario</u>
<u>foto</u>	<u>VARCHAR</u>	<u>Foto del puesto</u>
<u>telefono</u>	<u>VARCHAR</u>	<u>Teléfono del puesto</u>
<u>info</u>	<u>VARCHAR</u>	<u>Información acerca del puesto</u>

### ProductoPuesto

<u>Nombre</u>	<u>Tipo</u>	<u>Descripción</u>
<u>idproPues (pk)</u>	<u>INT</u>	<u>Identificador y clave privada del producto</u>
<u>idPuesto (fk)</u>	<u>INT</u>	<u>Id del puesto</u>
<u>nombre</u>	<u>VARCHAR</u>	<u>Nombre del producto</u>
<u>descripcion</u>	<u>VARCHAR</u>	<u>Información acerca del producto</u>
<u>precio</u>	<u>DOUBLE</u>	<u>Coste del producto</u>
<u>stock</u>	<u>BOOLEAN</u>	<u>Si queda producto disponible</u>

## Pedido

<u>Nombre</u>	<u>Tipo</u>	<u>Descripción</u>
<u>idPedido (pk)</u>	<u>INT</u>	<u>Identificador y clave privada del pedido</u>
<u>idUsu (fk)</u>	<u>INT</u>	<u>Id del usuario que hace el pedido</u>
<u>cantidad</u>	<u>INT</u>	<u>Cantidad que solicita</u>
<u>peso</u>	<u>INT</u>	<u>Peso que puede tener</u>
<u>fecha</u>	<u>DATE</u>	<u>Fecha de realización del pedido</u>
<u>pagado</u>	<u>BOOLEAN</u>	<u>Si el pedido está pagado</u>
<u>enviado</u>	<u>BOOLEAN</u>	<u>Si el pedido está pagado</u>
<u>total</u>	<u>DOUBLE</u>	<u>Precio total del pedido</u>

## Tipo

<u>Nombre</u>	<u>Tipo</u>	<u>Descripción</u>
<u>idTipo (pk)</u>	<u>INT</u>	<u>Identificador y clave privada del tipo de usuario</u>
<u>nombre</u>	<u>VARCHAR</u>	<u>Nombre del rol del usuario</u>

## 4. Despliegue

### 4.1 Despliegue Local

Para el despliegue de la aplicación en local es un requisito obligatorio hacerlo en Apache. Este servidor es el que hemos estado utilizando a lo largo de este curso para hacer las diferentes prácticas de clase.

Es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, es el servidor HTTP más usado.

## 4.2 Despliegue AWS

Para el despliegue de la aplicación en internet que permita el acceso a los usuarios, he optado por hacer uso de un servicio de cloud pública, principalmente por estos motivos:

- Crecimiento en los últimos años de este tipo de servicios, que se traduce en la importancia que tienen hoy en día que han seleccionado esta opción para sus servicios.
- Facilidad de despliegue del código y rapidez para instanciar otros servicios necesarios como BBDD.
- Además, nos permite obtener una independencia del hardware y sistema operativo, porque no es necesario mantenerlo y por tanto puedo centrarme en el desarrollo de la aplicación que es el objetivo.

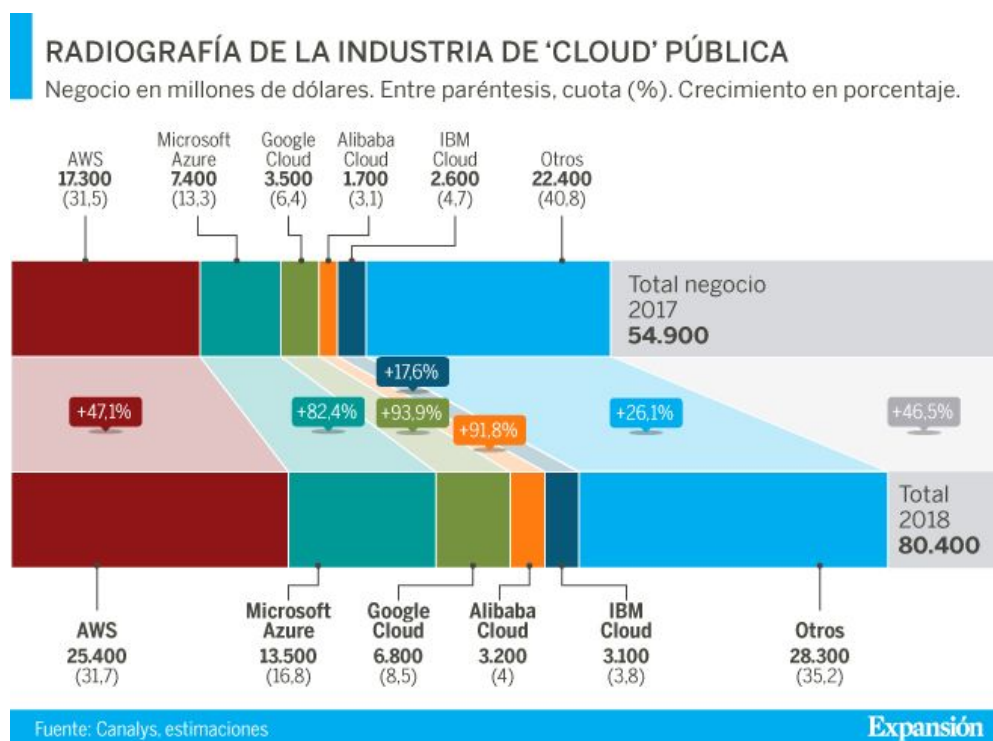


Figura 3. Comparativa año 2018 de Servicios Cloud [Expansión]

Debido a las facilidades ofrecidas a estudiantes para este tipo de proyectos, a que es la empresa más conocida del mercado he considerado apropiado hacer uso de AWS (Amazon Web Services). Además, este proveedor era compatible con las tecnologías empleadas (PHP 7.2, MySQL, Javascript, Laravel,...).

### 3. DESARROLLO E IMPLEMENTACIÓN

La aplicación se divide en cuatro partes a desarrollar, una primera parte global para todo el que se conecte a la web, una segunda parte para los clientes que consumen los productos, una tercera parte para los vendedores que tienen los puestos y una cuarta parte para el administrador de la web.

#### 1. Parte global

Aquí se encuentran las vistas comunes a todos los que se conectan a la web, pueden visualizar los puestos del mercado y sus productos, pero no podrían hacer uso de sus funcionalidades, tales como comprar o añadir un puesto como favorito. También pueden acceder al registro de usuarios, para poder hacer compras o solicitar ser vendedor, y pueden ir al login para ingresar en la web.

#### 2. Cliente

Una vez te registras como cliente tienes acceso a otras vistas diferentes en las que ya puedes desarrollar compras, ver el estado de tus pedidos, ver los productos que tienes en la cesta de la compra. Puede este usuario también añadir los puestos a favoritos para tener un acceso más rápido y una parte en la que puede modificar sus datos de perfil.

#### 3. Vendedor

Para este tipo de usuario las vistas cambian, este tendría acceso a una parte diferente en la que podría ver los pedidos que le han realizado y a que puesto pertenece, otra en la que aparece los pedidos que ya han sido enviados por el vendedor, y la parte importante donde el vendedor crea los puestos y puede visualizar los que tiene, desde aquí se accede también a los productos que hay en cada puesto y puede realizar el CRUD de ellos.

#### 4. Administrador

Esta parte es la que tiene el control sobre los usuarios y sus contenidos dentro de la web, tiene tres vistas desde las que puede visualizar los usuarios, puestos y productos. También puede hacer operaciones como cambiar el tipo de usuario y borrarlos. Los productos y los puestos también podría borrarlos.

## 4. DESPLIEGUE

Ficheros de Configuración acreditativos (del Servidor usado para el despliegue local de la Aplicación Web).

### 4.1 Despliegue Local

Para este apartado he montado XAMPP que viene con el servidor Apache ya operativo, una vez instalado añado una carpeta con el nombre del proyecto ( mercado ) en la carpeta htdocs de XAMPP. Después en el archivo de configuración de Apache (http) añado el virtualHost que hace referencia a mi proyecto para que cuando entre por el puerto 80 redirija a la web con https.

Después para que sea accesible desde https mi aplicación web he añadido la configuración () y así poder acceder de forma segura.

Y por último configuro DNS zona inversa() y zona directa para poder acceder desde la red interna o dentro del sistema en el que me encuentre en ese momento.

### 4.2 Despliegue AWS

Para el despliegue en el cloud de amazon he utilizado Elastic Beanstalk, es un servicio que proporciona amazon en su consola de administración.

Los pasos a seguir son:

- ❑ Añadir al proyecto laravel una carpeta llamada “.ebextensions” con un archivo de configuración que lance las migraciones y seed de la aplicación, este archivo puede ser de código YAML o JSON.

```
container_commands:
```

```
01-migrate:
```

```
command: php artisan migrate
```

```
02-seed:
```

```
command: php artisan db:seed
```

- ❑ En la carpeta bootstrap/cache borrar el archivo config.php para que no nos coja la configuración del proyecto en desarrollo.
- ❑ Después modificamos el archivo database.php que se encuentra en la carpeta config , para que la aplicación coja la configuración de la base de datos del servidor automáticamente, sin tener que pasarle los datos de la conexión a mano.

```
'host' => env('RDS_HOSTNAME', '127.0.0.1'),  
'port' => env('RDS_PORT', '3306'),  
'database' => env('RDS_DB_NAME', 'forge'),  
'username' => env('RDS_USERNAME', 'forge'),  
'password' => env('RDS_PASSWORD', ''),
```

- ❑ Una vez hechos todos estos cambios en el proyecto se seleccionan todos los archivos que tenemos dentro de la carpeta del proyecto y se comprimen en un archivo ZIP.
- ❑ Ya con todo lo anterior hecho, en la consola de administración podemos crear un entorno de servidor web.
- ❑ Se añade el nombre de la aplicación , información del entorno que es la parte donde hay que darle nombre a la web, plataforma que es donde defino que mi proyecto es de tipo PHP y la versión la 7.2 y código de la aplicación que escojo la opción cargar código.
- ❑ Después añado dos opciones más: en opciones de contenedor cambio el raíz del documento que corresponde a /public, y activo una base de datos Mysql en la que se añade usuario y contraseña.
- ❑ Ya con todo esto hecho se lanza y te crea la aplicación en la web. Después dentro de la consola puedes llevar un control de versiones de la aplicación y puedes tener las versiones en la cloud y volver atrás en el momento que quieras.

## 5. CONCLUSIONES



## 6. BIBLIOGRAFÍA

### **Páginas para Bootstrap:**

- <https://www.w3schools.com/bootstrap4/default.asp>
- <https://bootstrap-vue.org/docs/components/table>
- <https://getbootstrap.com/docs/4.5/getting-started/introduction>

### **Código JS:**

- [https://www.w3schools.com/js/js\\_intro.asp](https://www.w3schools.com/js/js_intro.asp)

### **Documentación php:**

- <https://www.phpdoc.org/>

### **Código php:**

- <https://www.php.net/docs.php>

### **Laravel:**

- <https://laravel.com>

### **Ayuda y fallos:**

- <https://es.stackoverflow.com/>
- [https://blog.pleets.org/article/mvc-en-laravel#:~:text=MVC%20\(modelo%2C%20vista%2C%20controlador,un%20aspecto%20de%20la%20aplicaci%C3%B3n.](https://blog.pleets.org/article/mvc-en-laravel#:~:text=MVC%20(modelo%2C%20vista%2C%20controlador,un%20aspecto%20de%20la%20aplicaci%C3%B3n.)
- <https://rafarjonilla.com/que-es/backend/>
- <https://www.hostinger.es/tutoriales/que-es-mysql/>
- [https://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache)

### **Apuntes DWES:**

- [https://docs.google.com/document/d/1VSkjNkwesTLDnWjM\\_im6zWAqh\\_1Z-ZDOByKLDiL3Rq8/edit](https://docs.google.com/document/d/1VSkjNkwesTLDnWjM_im6zWAqh_1Z-ZDOByKLDiL3Rq8/edit)