



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

INTRODUCCIÓN	3
Objetivos	3
Requisitos	3
Elección de la aplicación	5
DISEÑO SOLUCIÓN	6
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
DESARROLLO E IMPLEMENTACIÓN	12
Parte global	12
Cliente	13
Vendedor	13
Administrador	13
DESPLIEGUE	14
4.1 Despliegue Local	14
4.2 Despliegue AWS	14
CONCLUSIONES	14
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
BIBLIOGRAFÍA	15

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.

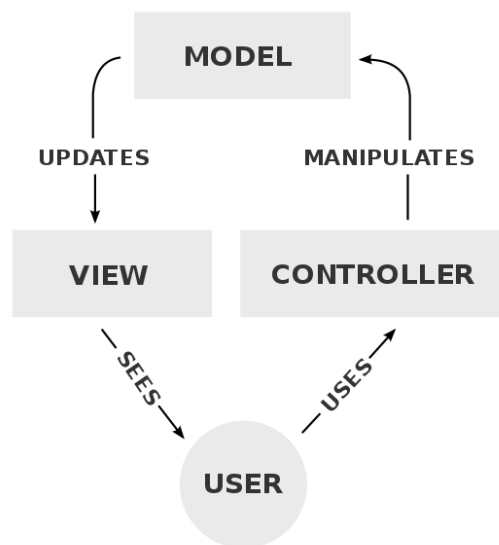


Figura 1. Modelo Vista Controlador [[Wikipedia](#)]

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.

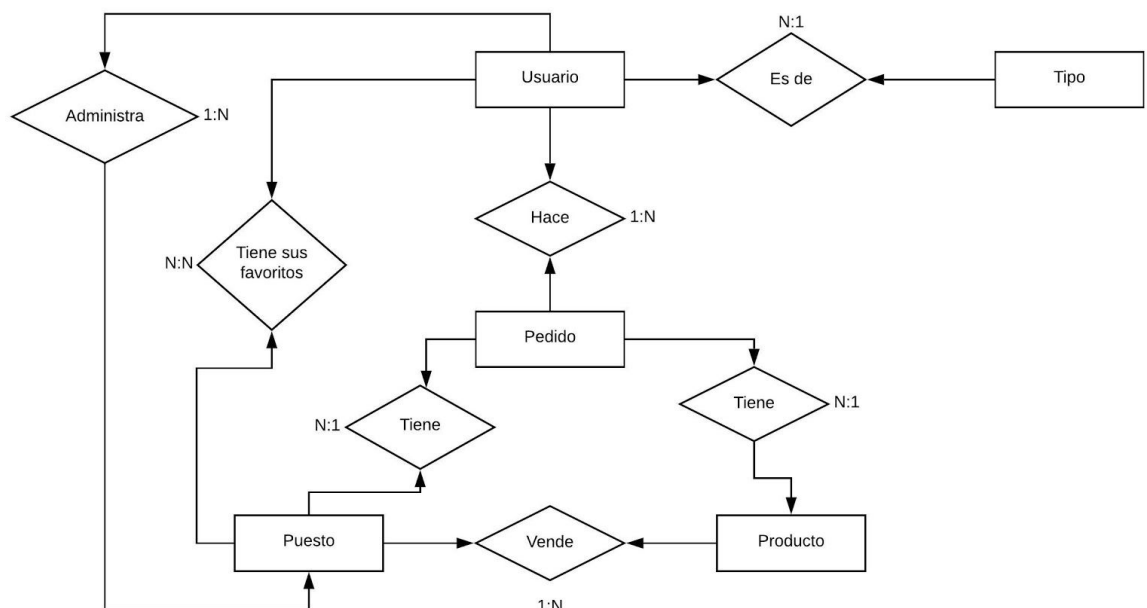


Figura 2. Modelo entidad relación del proyecto.

- **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.

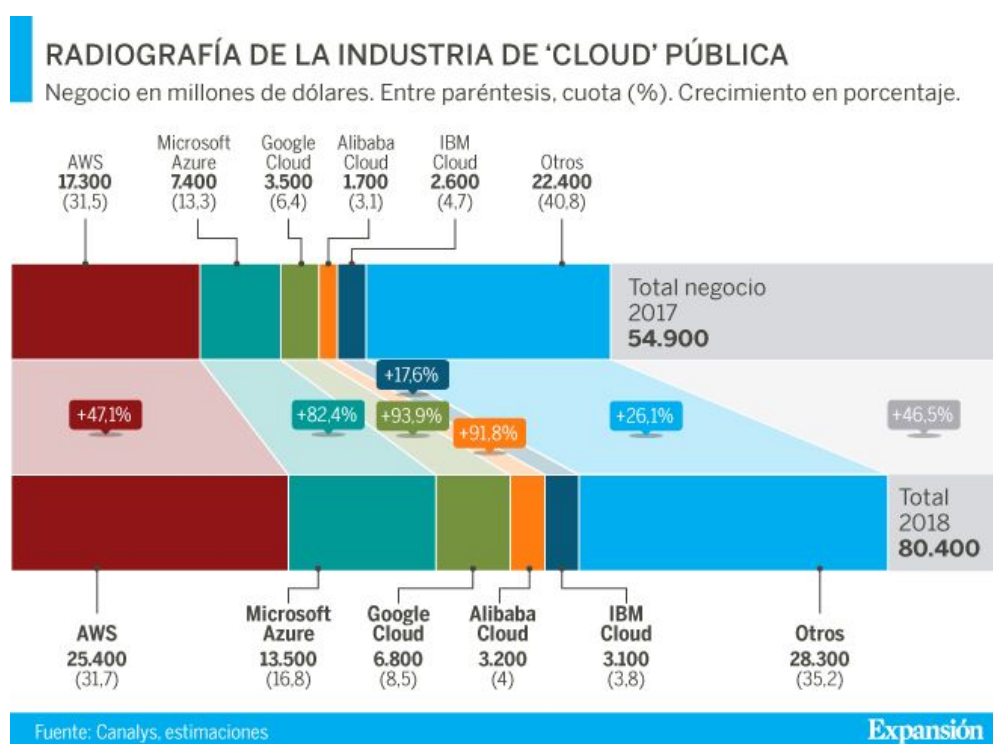


Figura 3. Comparativa año 2018 de Servicios Cloud [Expansión]

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.

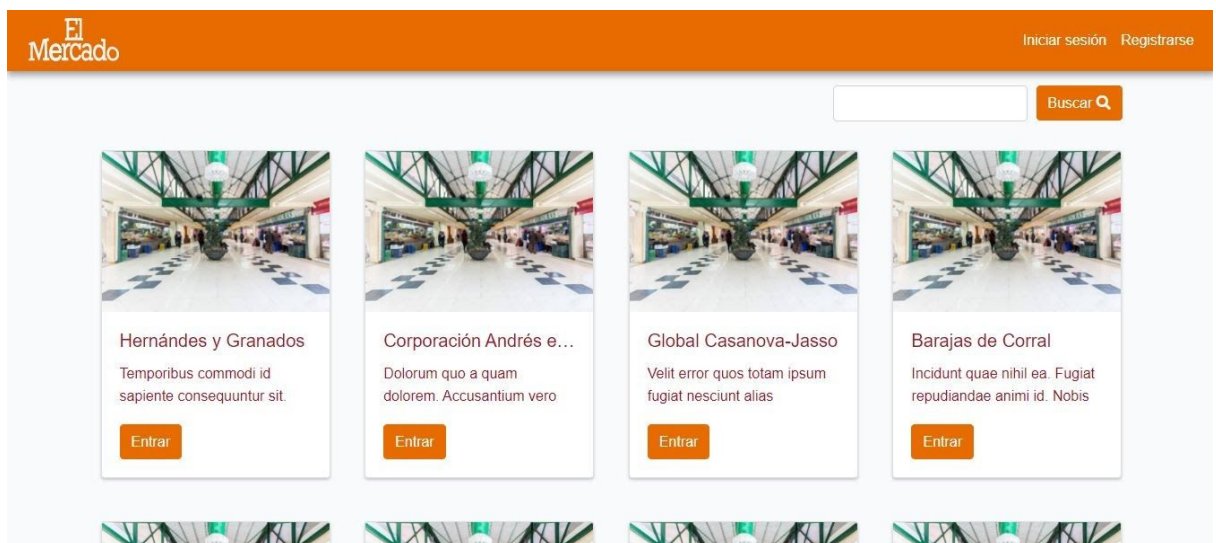


Figura 4. Pantalla inicio.

La pantalla de inicio es la vista que es común a todos los usuarios estén conectados o no (figura 4). Aquí se encuentra el buscador, que está compuesto por un componente Vue y una petición get a la api con axios, nos devuelve las coincidencias que haya en la base de datos y ya seleccionas la que quieras, esto te busca y te muestra los puestos que te devuelva. También tienes las opciones de loguearte y registrarte.

Figura 5. Pantalla Registro.

En la pantalla de registro (Figura 5) se validan los campos con la librería Validator que implementa laravel, pero además con un componente Vue, añadiendo el campo email para comprobar que el email no existe en la base de datos, y el campo dni el cual tiene dos tipos de comprobaciones, una para ver si existe registrado en la base de datos el dni y otra si es válido el dni.


Si el usuario quiere ser vendedor tiene que seleccionar la opción de solicita ser vendedor y rellenar el campo del nif relacionado de la empresa, para que el admin pueda darle permisos. En el campo de fecha de nacimiento también se comprueba que la persona que se loguee sea mayor de edad.

```
<template>
  <div class="form-group col-md-6">
    <label for="dni" >DNI</label>
    <input id="dni" type="text" class="form-control "
name="dni" placeholder="00000000G" value="" required
@keyup="dniCheck" />
    <strong v-if="this.respuesta == 'incorrecto'">El dni
no es correcto.</strong>
    <strong v-if="this.respuesta == 'existe'">El dni ya
existe.</strong>
  </div>
</template>
<script>
  const axios = require('axios').default ;
  export default {
    data() {
      return{
```

```

        respuesta:'',
        dni:'',
    },
    methods:{
        dniCheck : function(){
            this.dni = $('#dni').val();
            axios.get('api/dni/'+
this.dni).then((response) => {this.respuesta =
response.data;});
        }
    },
    mounted() {console.log('Component mounted great.')}
}
</script>

```



The image shows a login form titled "Iniciar Sesión" (Login) with a logo consisting of a white 'M' on a red and orange background. Below the logo, there are two input fields: "Email" with the value "mata.victoria@hotmail.co" and "Contraseña" (Password) with the value "12345cya". The password field has an eye icon to toggle visibility. Below the password field is a checkbox labeled "Recordar" (Remember). At the bottom is an orange button labeled "Entrar" (Enter).

Figura 6. Pantalla login.

El login también está compuesto con Vue para añadir la funcionalidad del ojo para poder visualizar la contraseña.

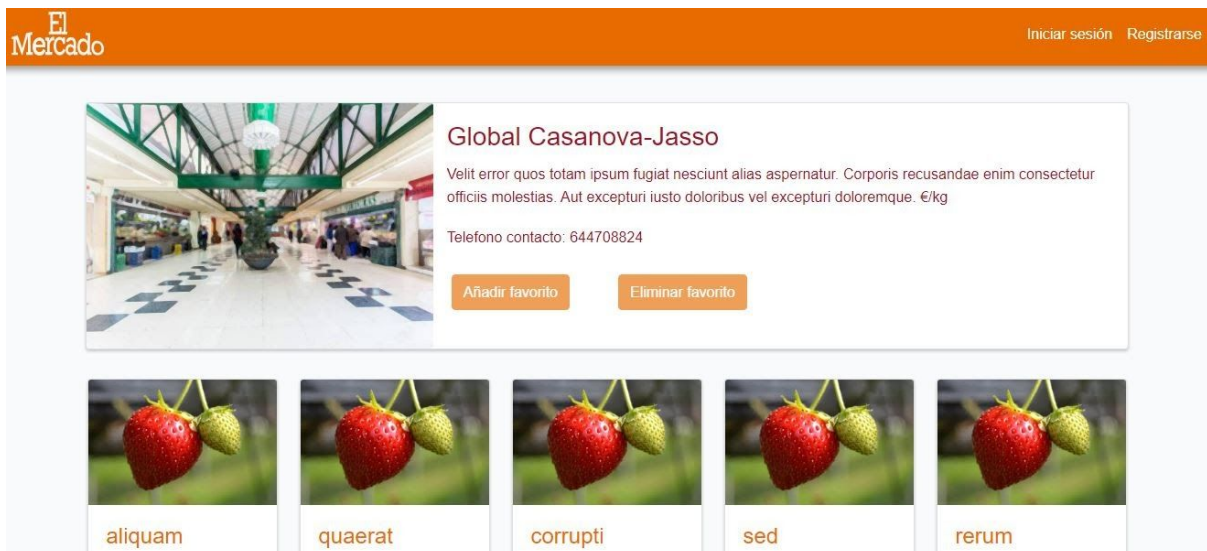


Figura 7. Pantalla puesto, sin loguearse.

En la pantalla del puesto si no eres un cliente logueado los botones con los que puedes añadir o eliminar como favorito un puesto aparecen desactivados. Igual ocurre si intentas comprar un producto, el botón de añadir a la cesta se desactiva (figura 8).

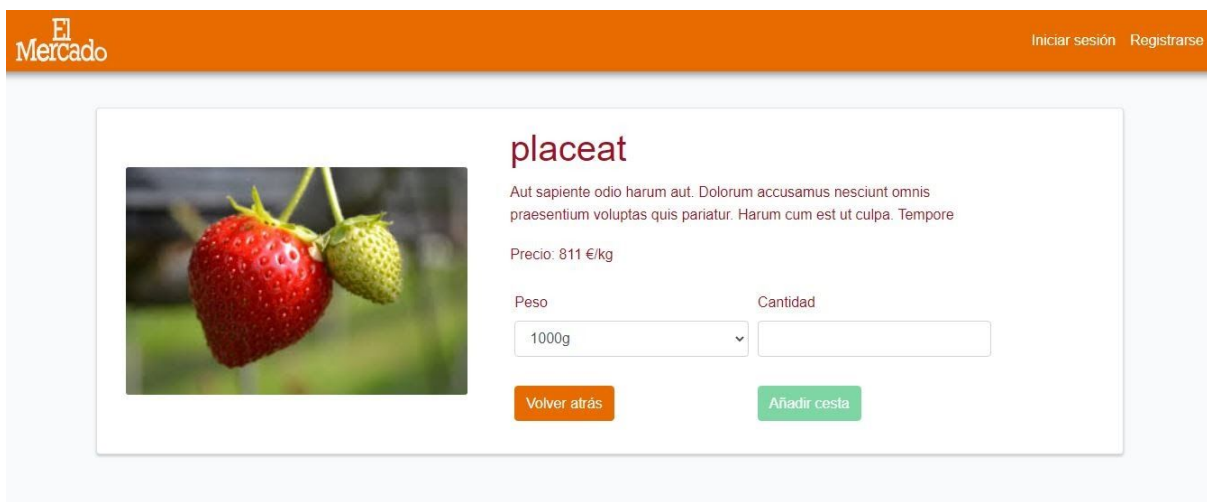


Figura 8. Producto sin loguearse

En la pantalla de producto he añadido un select para que el usuario pueda poner las cantidades y el peso del producto que quiere, ya internamente en el controlador hace la operación para calcular el total y realizar el pedido.

```
$total = $cantidad * $peso;
$precio = $producto->precio /1000;
$totalPedido = $total * $precio ;
```


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.

En esta parte ya tendría el botón de la cesta habilitado (figura 8) y realizar el pedido sin problema.


	Nombre	Puesto	Cantidad	Peso	Fecha	Enviado	Pagado
	totam	Valdivia, Pagan y Porras e Hijos	2	250 g	12/06/2020	No	0.5 €
	totam	Valdivia, Pagan y Porras e Hijos	4	250 g	12/06/2020	No	1 €
	doloremque	Ureña-Mireles	1	500 g	12/06/2020	Si	4.5 €

Figura 9. Pantalla Mis pedidos del cliente

En la pestaña de mis pedidos he añadido una tabla para representar los datos de los pedidos hechos por el cliente. Aquí puede comprobar si el pedido le ha sido enviado ya por el vendedor (figura 9).

	Nombre	Cantidad	Peso	Precio	Eliminar
<input checked="" type="checkbox"/> 	et	1	100 g	6.4 €	

€

Figura 10. Pantalla mis pedidos del cliente

En el icono de la cesta del navbar, indico el número de productos que tengo en la cesta de la compra sin pagar. Al entrar en la cesta, represento los datos de los productos en una tabla, pero además le he añadido un checkbox con el que añado los productos a un array para pagar los seleccionados, a la vez cuando

seleccionas algún elemento, el precio se va sumando al total del dinero a pagar (figura 10).

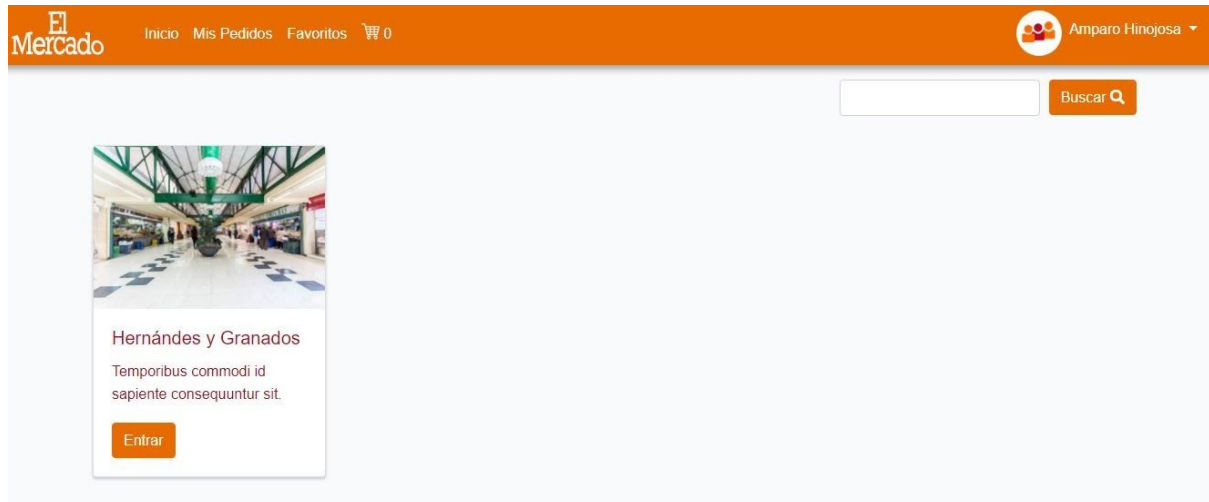


Figura 11. Pantalla favoritos

En la pantalla de favoritos solo aparecerán los puestos favoritos del usuario y no todos los puestos (figura 11).

Figura 12. Perfil de usuario

Los clientes también tienen una pestaña para los datos del usuario donde podrán modificar sus datos personales, o cambiar la imagen de usuario dentro de la aplicación web. En esta pantalla el usuario podrá darse de baja en cualquier momento (figura 12).

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.

Para ayudar al vendedor ante cualquier duda acerca de la web he habilitado un apartado específico(figura 13,14), donde he añadido un vídeo que ayudaría al usuario a registrar su propio puesto. Y ante cualquier otra pregunta hay datos de contacto con los que podría ponerse en contacto con el administrador. Si el usuario quisiera también he añadido unos mandos donde podría poner música de fondo(figura 14).




Figura 13. Ayuda vendedor



Figura 14. Ayuda vendedor

En esta parte el usuario tendrá la posibilidad de crear sus propios puestos donde añadir sus productos, información acerca del puesto, añadir una foto y algunos datos más (figura 15). Cuando creas el puesto por primera vez, no se podrán añadir los productos, y para que el usuario tenga constancia de ello justo debajo le indico los pasos a seguir (figura 16).



Nombre Teléfono

Pedroza, Echevarría y Ayala y Asoc. 683523452

Información puesto

Ipsum tempore placeat veniam ratione sunt. Nihil similique ab veniam officiis quia quia qui aliquid. Deserunt culpa laboriosam impedit est ratione beatae. Cupiditate et sunt id

Foto del puesto

Elige la foto Browse

Guardar

Nuevo Producto




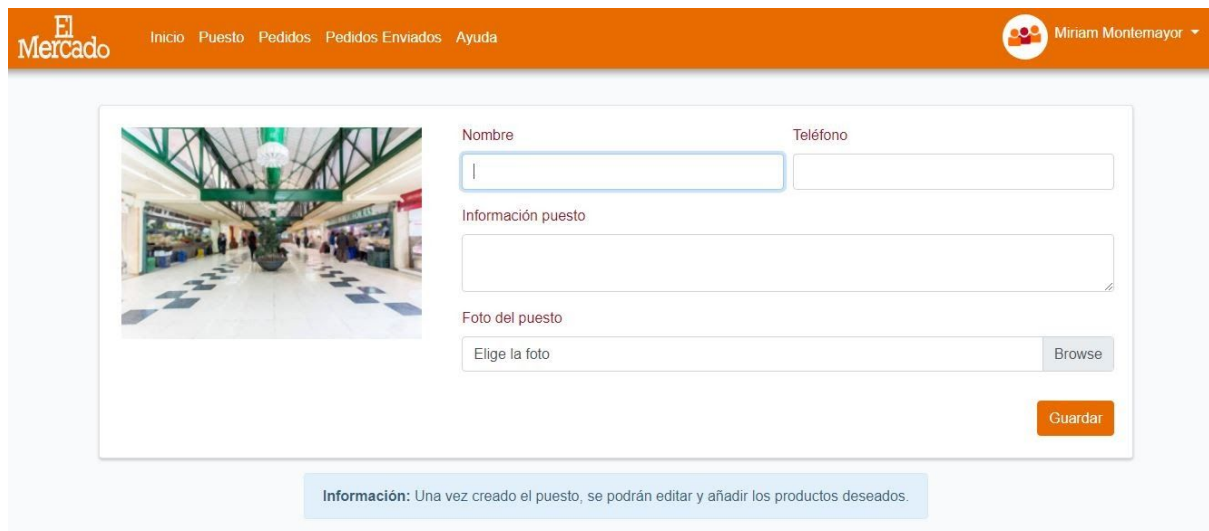
	Nombre	Precio	Descripción	Stock	Editar	Borrar
	in	1606818 €/kg	Delectus quisquam consequatur amet incidunt omnis distinctio. Quod voluptas porro fugiat. Quod qui quis ut hic. Perferendis eum	No		

Figura 15. Puesto para modificar



El Mercado Inicio Puesto Pedidos Pedidos Enviados Ayuda Miriam Montemayor

Nombre Teléfono

Información puesto

Foto del puesto
Elige la foto

Información: Una vez creado el puesto, se podrán editar y añadir los productos deseados.

Figura 16. Nuevo puesto.

El vendedor también puede visualizar los pedidos que han realizado los clientes desde sus cuentas, para ello he representado en una tabla todos los detalles necesarios para el vendedor, para que pueda hacer el envío del producto a la persona adecuada (figura 18) aquí tienen también un checkbox para cambiar el estado de los pedidos a enviados, los cuales se pueden visualizar desde otra pantalla (figura 17).



#	Producto	Puesto	Cantidad	Cliente	Dirección	Teléfono	Fecha	Precio
28	vitae	Barajas de Corral	50 g	Amparo Gaytán	Travessera Alex, 2, Bajo 0º, 51015, Os Guerrero del Pozo	607637175	11/06/2020	199.65 €

Figura 17. Pedidos enviados.



#	Producto	Puesto	Cantidad	Cliente	Dirección	Teléfono	Fecha	Precio	Enviar
28	vitae	Barajas de Corral	50 g	Amparo Gaytán	Travessera Alex, 2, Bajo 0º, 51015, Os Guerrero del Pozo	607637175	11/06/2020	199.65 €	<input checked="" type="checkbox"/>

Figura 18. Pantalla pedidos del vendedor

Cuando entramos en la pantalla puesto, nos aparece un listado de los puestos ya creados del usuario, desde aquí se pueden hacer modificaciones del puesto o borrarlo si ya no es necesario (figura 19).



	Nombre	Teléfono	Info	Editar	Borrar
	Pedroza, Echevarría y Ayala y Asoc.	683523452	Ipsum tempore placeat veniam ratione sunt. Nihil similique ab veniam officiis quia quia qui aliquid. Deserunt culpa laboriosam impedit est ratione beatae. Cupiditate et sunt id consequuntur. Ex occaecati natus perspiciatis fugiat. Alias qui ipsam architecto similique qui voluptas. g		
	Silva de Valles e Hijos	637336718	Provident aperiam et cum in. Eos perferendis adipisci sunt ducimus tempora eum maxime. Beatae aliquid facilis veniam inventore ab sit incidunt. Ea est fugiat expedita quis aliquam rerum nemo. Eveniet in vel cum asperiores. Itaque qui aliquam aut necessitatibus optio rerum. g		

Figura 19. Pantalla puesto del usuario vendedor

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.

En la pantalla productos como se puede ver en la figura 20, aparece una tabla con todos los productos añadidos a la base de datos, y si el administrador tuviera conocimiento de algún fallo podría eliminarlo.





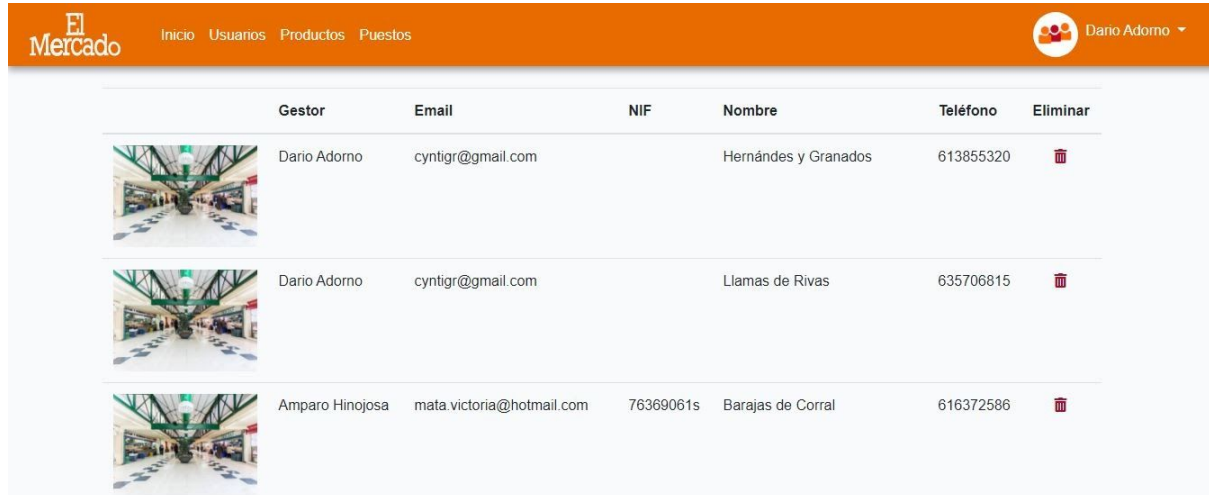
	Producto	Descripción	Precio / kg	Stock	Puesto	Gestor	NIF	Eliminar
	omnis	Nostrum maxime rerum quidem. Cupiditate veritatis quisquam esse soluta ullam. Aliquid inventore quisquam repellat maiores dolor aperiam molestiae. Aut expedita impedit impedit nihil at aperiam.	3 €	No	Hernández y Granados	Dario Adorno		
	sunt	Fugit id optio amet explicabo quod voluptatem quia. Qui sed expedita dolorem dolorem expedita. Vel cupiditate laboriosam quo omnis iure modi.	66121 €	Si	Hernández y Granados	Dario Adorno		

Figura 20. Pantalla productos administrador.

Al igual que en los productos, en la pantalla puestos podemos visualizarlos en una tabla con los datos correspondientes, era importante mostrar el correo y el teléfono por si el administrador tuviera que ponerse en contacto con el vendedor por algún problema en la web. Desde aquí también he habilitado que pueda borrar si fuera necesario.









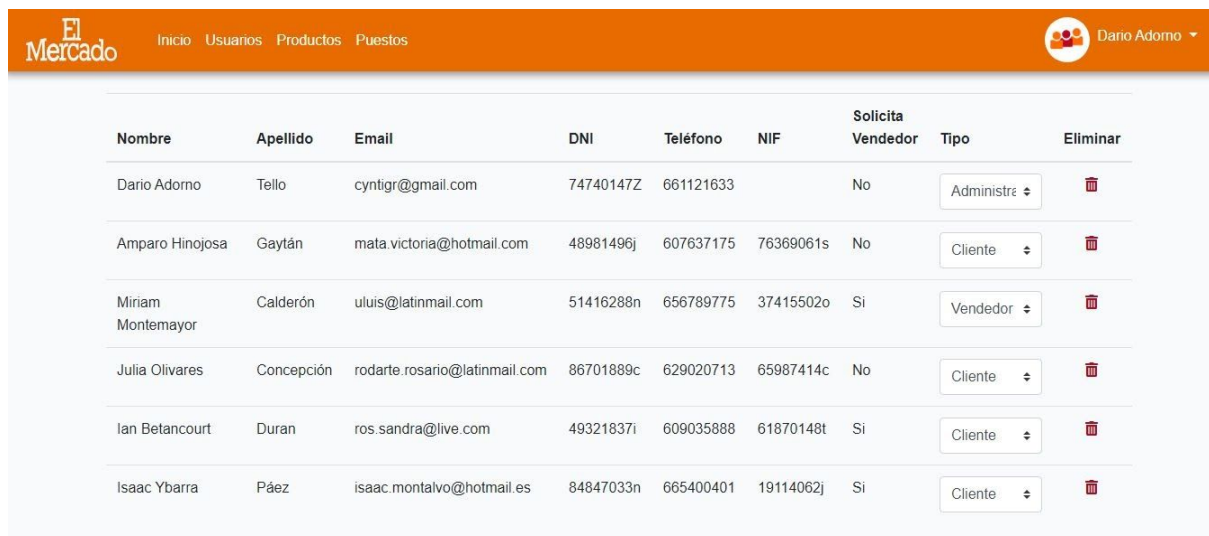
Gestor	Email	NIF	Nombre	Teléfono	Eliminar
 Dario Adorno	cyntigr@gmail.com		Hernández y Granados	613855320	
 Dario Adorno	cyntigr@gmail.com		Llamas de Rivas	635706815	
 Amparo Hinojosa	mata.victoria@hotmail.com	76369061s	Barajas de Corral	616372586	

Figura 21. Pantalla puestos del administrador.

Por último nos encontramos con la pantalla de datos de usuario, desde la que el administrador tiene el control para cambiar el tipo de usuario con ajax, para no tener que recargar la página cada vez que haga un simple cambio de usuario (figura 22).









Nombre	Apellido	Email	DNI	Teléfono	NIF	Solicita Vendedor	Tipo	Eliminar
Dario Adorno	Tello	cyntigr@gmail.com	74740147Z	661121633		No	Administrador	
Amparo Hinojosa	Gaytán	mata.victoria@hotmail.com	48981496j	607637175	76369061s	No	Cliente	
Miriam Montemayor	Calderón	uluis@latinmail.com	51416288n	656789775	37415502o	Si	Vendedor	
Julia Olivares	Concepción	rodarte.rosario@latinmail.com	86701889c	629020713	65987414c	No	Cliente	
Ian Betancourt	Duran	ros.sandra@live.com	49321837i	609035888	61870148t	Si	Cliente	
Isaac Ybarra	Páez	isaac.montalvo@hotmail.es	84847033n	665400401	19114062j	Si	Cliente	

Figura 22. Pantalla usuarios del administrador.

4. DESPLIEGUE

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 (httpd) añado el virtualHost que hace referencia a mi proyecto para que cuando entre por el puerto 80 redirija a la web con https (figura 23).

```
<VirtualHost *:80>
ServerName mercado.cgr-ies.edu
Redirect / https://mercado.cgr-ies.edu
ErrorDocument 404 'No me encuentro,prueba con otro recurso'
</VirtualHost>
```

Figura 23. Archivo de configuración de apache httpd.conf.

Después para que sea accesible desde https mi aplicación web, he añadido la configuración a https.conf y así poder acceder de forma segura (figura 24).

```
298 <VirtualHost *:443>
299     DocumentRoot "C:/xampp/htdocs/mercado/public"
300     ServerName mercado.cgr-ies.edu
301     ServerAlias www.mercado.cgr-ies.edu
302     SSLEngine on
303     SSLCertificateFile "conf/ssl.crt/claveSSL.crt"
304     SSLCertificateKeyFile "conf/ssl.key/claveSSL.key"
305 </VirtualHost>
```

Figura 24. Archivo de configuración de apache httpd-ssl.conf.

Y por último configuro DNS zona inversa(figura 25) y zona directa(figura 26) para poder acceder desde la red interna o dentro del sistema en el que me encuentre en ese momento.

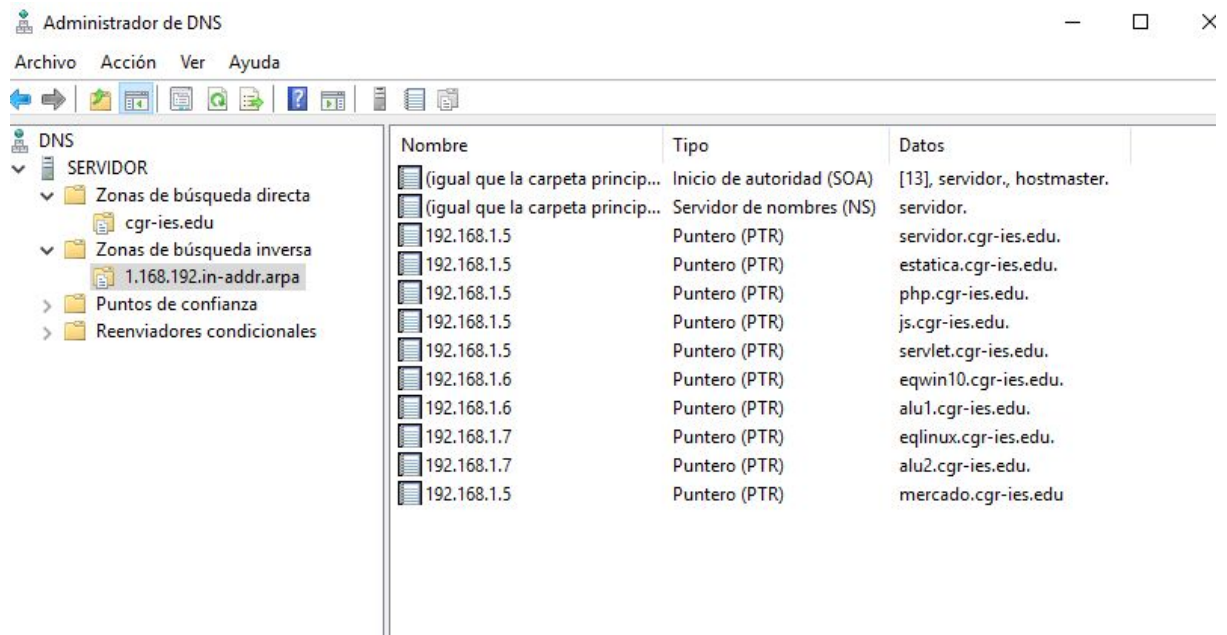


Figura 25. Configuración DNS zona inversa.

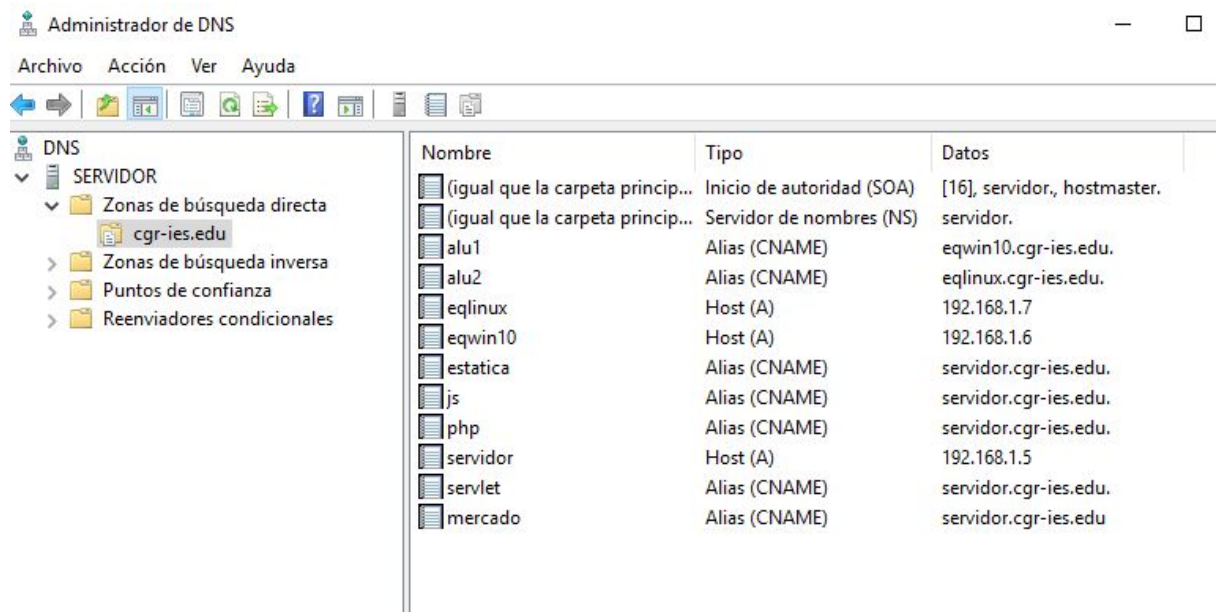


Figura 26. Configuración DNS zona directa.

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 añadido 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

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

Apuntes DWES:

- https://docs.google.com/document/d/1VSkJNkwesTLDnWjM_im6zWAqh_1Z-ZDOByKLDiL3Rq8/edit