

PROYECTO DE CICLO DE G.S. DE DESARROLLO DE APLICACIONES WEB

NEW ERA SHOP



IGNACIO AGUAYO LOPEZ **CURSO 2023/24**

ÍNDICE

1 RESUMEN.....	3
2 INTRODUCCION.....	4
3 OBJETIVOS.....	5 y 6
3.1 OBJETIVOS EN PROGRAMACION.....	7
4 HIPOTESIS DE TRABAJO.....	8
4.1 OBJETIVOS ESPECIFICOS.....	8
4.2 METODOLOGÍA.....	9
5 TECNOLOGIAS.....	10
5.1 HARDWARE.....	10
5.2 SOFTWARE.....	10 - 17
6 BASES DEL PROYECTO.....	18 - 22
7 MÉTODO	23
7.1 LOGIN	23 - 25
7.2 REGISTER	25 - 27
7.3 CARRITO	28 - 31
7.4 PRODUCTO.....	31 - 37
7.5 CONTACTO.....	37 - 38
7.6 MENSAJES	39 - 40
7.7 USUARIOS	41 - 43
7.8 INICIO	44
7.9 NOSOTROS	45
7.10 FOOTER + COOKIES	46
7.11 P. PRIVACIDAD + T. DATOS	47
8 TIEMPO DE EJECUCIÓN	48
9 CONCLUSION	48 - 49
10 ANALISIS DE USO (DIAGRAMAS)	49 -50
11 ANEXOS	50
12 WEBGRAFÍA	51

1. RESUMEN DEL PROYECTO

La idea de este proyecto es hacer una tienda online para ofrecerla a los comercios tradicionales convenciendoles que es hora de que se pasen a la nueva era de comercios, por eso el nombre hace referencia a ello NEW ERA SHOP, lo principal del proyecto es hacer una aplicación fácil y sencilla en la cual hay dos roles el de particular que es para los clientes y otro que es admin que es para el dueño de la tienda, una vez conseguido esto pondremos una sección de productos donde se verán cuales hay para su venta en caso del particular solo se verá los detalles y añadirlo al carrito, por otra parte en el rol de admin se podrá modificar, borrar o incluso agregar un nuevo producto.

Todo esto tiene que tener ciertos componentes que son los siguientes, se usará una base de datos donde tendremos almacenada los productos, usuarios, mensajes y compras, estos dos últimos se implementarán si la planificación es correcta, habrá mensajes para el manejo de los datos de recibir, enviar usuarios, productos... también usaremos bootstrap porque es una de las maneras más fáciles para diseñar una página web y queda más atractivo con models y demás y por último usaremos Vue.js en Visual Studio Code porque es una estructura simple y fácil de entender en la que cada componente tiene su template, script y style (este último se añadirán en caso de que con bootstrap no podamos hacerlo).

Por otro lado también tiene diferentes funciones como administrador después de las mencionadas anteriormente como son la gestión de mensajes cuando el usuario manda un mensaje mediante el contacto y la manipulación de los usuarios, es decir, un listado con los usuarios que hay para que pueda borrar el usuario directamente, modificar alguno de sus datos o simplemente ver sus detalles que son el nombre, apellido, teléfono y el email.

En conclusión, la aplicación está diseñada para que la experiencia tanto de usuario como la del administrador sea la mejor posible y que todo o casi todo se pueda gestionar desde la misma.

2. INTRODUCCION

En el desarrollo de este trabajo de fin de grado el objetivo o el mayor de los objetivos entre otros es el desarrollo de una tienda online, en el mundo de la programación o mejor dicho en el desarrollo de la tecnología somos consciente de que cada vez avanzan a pasos agigantados, por lo que uno de los sectores que se iban a quedar atrás son los comercios tradicionales, esta web va enfocada a esa gente que no estan pegados a la tecnologia que son la mayoría de propietarios de comercios tradicionales, hecha con una experiencia de usuario facil y muy intuitiva tanto para el administrador como para el usuario particular que quiera comprar o simplemente visitar la página.

Esta web ha sido desarrollada por conocimientos que se han adquirido durante el curso tanto el uso de base de datos con php, modelarla con bootstrap 5... el objetivo es implementar gran parte de lo aprendido, para ello he intentado meter varias opciones de modificacion... etc.

Con este trabajo quiero llegar al punto que en cualquier momento puedo presentarsela a cualquier comercio y lo bueno de esto es que se puede usar como plantilla para crear diferentes tiendas, por otro lado si quisiera podria enfocararlo a una pagina web general y que cada empresa se registrase para que ofreciese sus productos ya que en el registro de admin se debe poner la empresa el cif...

En cuanto a seguridad que es una de las cosas mas importantes a dia de hoy, debido a los varios ataques que se realizan a las grandes corporaciones no es que tenga las mismas medidas pero si esta protegido para inyecciones mysql con sentencias preparadas, escape de caracteres especiales, hash en las contraseñas...

En resumen, este proyecto esta desarrollado con lenguajes aprendidos en el curso con unos sistema de seguridad que se han ido adquiriendo a lo largo del ciclo formativo y con una experiencia de usuario bastante buena y muy intuitiva, con el objetivo de ayudar a los comercios tradicionales que no estan tan digitalizados a que el salto a la tecnología le parezca tan sencillo que no tenga que memorizar nada solo con intuición saber cada cosa.

3. OBJETIVOS

1º- Poder hacer Login con comprobación en la base de datos si ese usuario existe y si las credenciales son correctas o no

2º- Poder hacer register para registrar usuarios tanto particulares o un admin de la empresa en la cual se comprobara el email si es formato correcto, el telefono si esta ya registrado o no, el email si esta registrado o no, en caso de ser rol admin se despliega otras 3 secciones que son el nombre, domicilio y el CIF de la empresa, que tambien seran comprobados si existen o no.

3º- Si no hay autenticación en los productos solo tendran disponible ver detalles y cuando se pulse se abrira un model donde pondrá ver video (pequeño video explicativo cogido de youtube), en caso de hacer log si es particular en productos tendra un boton extra que es añadir al carrito que este se añadira automaticamente por lo contrario si el rol es admin tendra 3 opciones que son modificar (podra modificar todo lo del producto, en el codigo de producto habra una comprobacion para que no haya productos duplicados con el mismo codigo), borrar, lo elimina de la base de datos directamente, ver detalles para ver como lo veria el usuario.

4º- Agregar producto, esta opcion solo la tendrá disponible el administrador donde le pidiran todos los datos, nombre, precio, cantidad, descripcion, imagen, video... y cuando le de se añadira a la base de datos y se mostrará al usuario.

5º- Tambien hay un buscador el cual filtra para buscar un producto.

6º- La sección de contacto, formulario simple con nombre, apellidos, email, telefono, asunto y descripción del mensaje, en caso de enviar le saldra un mensaje como que esta enviado y se pondra en contacto con el.

7º- En caso de estar en rol de admin habrá una pestaña llamada mensajes que sera todos los mensajes que mande la gente mediante el formulario de contacto y

podrá verlo se abrirá un model con todo y podrá eliminarlo en caso de que quiera quitarlo de ahí.

8º- Otra sección que tendrá el admin será usuarios, la cual como su nombre indica tendrá una lista de todos los usuarios que hay registrados donde podrá modificar algun dato de ellos o incluso eliminandolo el cual habrá un model que le pregunte si esta seguro de ello o no.

9º- Habra una sección que será inicio con algo de enganche y de ejemplo, otra de nosotros explicando un poco a que se dedicaria, esto es ejemplos porque es un proyecto, en el inicio habra un boton que se llame ir a cursos el cual lleva a la pestaña productos.

10º- No menos importante tendrá su nav como es obvio y su footer con información general y 4 links a redes sociales, a la derecha para poner la dirección y a la izquierda la calle donde se encuentra y demás.

11º- Al iniciar la App.vue saldra un mensajito de las cookies, un mensaje simple donde habra 3 opciones aceptar todas las cookies, aceptar solo necesarias o rechazar, en el mensaje habra un link en las palabras politica de cookies que si se le pincha le llevará a esa pagina donde irian las cookies.

12º- El carrito, aquí es donde ira los productos que se hayan añadido, cada usuario tiene su propio carrito ya que se guarda en el storage y cuando se cierra sesión se limpia pero al volver iniciarla se sigue manteniendo, habrá tambien el formulario de pagar que se abrirá con un model en el cual se pedirá el n.º de tarjeta, el cvc y la fecha de caducidad, se pedirá que se acepte la politica de datos y la de privacidad (cada uno con sus respectivos enlaces por si quieren visitarla de que se trata) y si no se selecciona no dejará pagar, en caso de aceptar le saldra que se ha pagado con éxito y en la base de datos concretamente en purchases aparecerá la compra con los datos registrados y la fecha.

3.1. OBJETIVOS NIVEL PROGRAMACION

A parte de lo ya mencionado anteriormente algunos de los objetivos serán los siguientes.

1º – Manejo de base de datos a través de conectividad con php y mysql con sentencias preparadas para evitar inyecciones y protegido para XSS.

2º – Uso de Bootstrap para el diseño utilizando modals, efectos, iconos como puede ser el de ver u ocultar contraseña.

3º – Depurar código ya sea en consola de navegación o con Postman

4º – Uso de Vue.js en Visual Studio Code para implementar el uso de métodos, v-if...

5º – Uso de github para almacenar las fotos utilizadas para los productos

6º – Uso de los métodos de get, post...

7º – Aplicación en los scripts de async, await...

4. HIPOTESIS DE TRABAJO

El titulo principalmente salió hace unos 3 años en el cual aquel tiempo pensaba en montar una API con todo lo que hay a día de hoy como es X, Instragram, Whatsapp es decir todo en uno, y la llamaria NEW ERA ya que seria una revolucion y una buena competencia para ello, la sorpresa fue que cuando empecé el curso mi idea se iba para abajo puesto que todas las aplicaciones estaban implementando todo lo posible en una misma aplicación, vease a día de hoy que twitter hace directos como si fuera Youtube, Instagram tiene mensajería y estados en la sección de mensajes como whatsapp, por lo que al comunicarme que había que hacer un proyecto de fin de grado, este nombre lo tenía que usar por crear una revolución, entonces pensé en la nueva era de los comercios, las tiendas online y me venía perfecto el nombre para llamarlo NEW ERA SHOP.

La realidad es que queria llegar con esta aplicación a esos comercios tradicionales con la mentalidad de hacerle lo mas facil posible esa transición para que pudieran darse a conocer en el sector online y ahora que el e-commerce esta en su auge y cada vez mas supongo que seria de mas ayuda mi API, por otro lado la enfoqué para dos ambitos en caso de que quiera una página personal o tambien poder usarla como una especie de foro o comunidad donde cada empresa se pueda registrar con sus datos y vender sus propios productos.

4.1 OBJETIVOS ESPECIFICOS

Uno de los objetivos especificos es hacerla lo mas intuitiva posible para no crear problemas al usuario a la hora de manejarla y mucho menos al admin por lo que ese objetivo se cumplirá con creces.

El tiempo requerido para que cualquier usuario complete por ejemplo la compra es menos de normal ya que solo se le piden los datos básicos de tarjeta, cvc.. porque como esta registrado los datos de nombre y apellidos ya los tendríamos.

4.2 METODOLOGÍA

Un breve resumen de lo que se ha utilizado, ya que mas abajo lo explicaré mas detenidamente pero lo principal es Vue.js con interfaz interactiva y responsive.

Para la base de datos phpmyadmin con XAMPP para iniciarla y que tiene 4 tablas las cuales son para el manejo de registro, compras, usuarios... y un sistema de pago que guarda la compra en la base de datos en un hipotetico caso si hubiese alguna reclamación como puede ser “esto no lo he comprado yo”.

Por otro lado he registrado un usuario particular y admin para hacer pruebas de que funcione todo y evaluar la experiencia de usuario desde cada punto de vista.

Pero como he comentado antes abajo explico con mas detalle para que he usado cada software y que he desarrollado con el.

5. TECNOLOGIAS UTILIZADAS

5.1. TECNOLOGÍA HARDWARE

Procesador: AMD Ryzen 5 5600X 6-Core Processor 3.70 GHz

RAM: 16 GB

Sistema Operativo: Windows 11 v. 23H2

Gráfica: NVIDIA GeForce RTX 3050

5.2. TECNOLOGÍA SOFTWARE

Aquí explicare las diferentes tecnologías usadas y para que cada una de ellas, las empleadas en mi proyecto son: **PHP, XAMPP, PHPMYADMIN, MYSQL, VSCODE, VUE.JS, JS, HTML, CSS, POSTMAN, BOOTSTRAP.**

PHP.

logo php



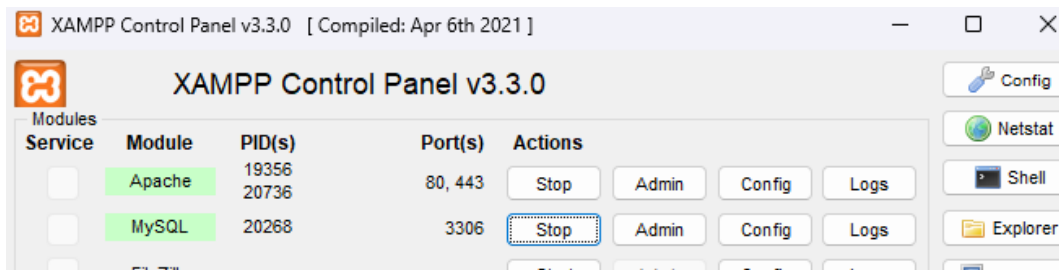
El primero a mencionar es PHP el cual he utilizado para la programación de los archivos de manejo de datos, como puede ser conectarse a la base de datos para el login, o el register donde se almacenan los datos en la base de datos, actualizar el producto, borrarlo... basicamente la mayoría de las funciones relacionadas con la base de datos he utilizado PHP.

La versión utilizada ha sido la **8.2.5**

XAMPP



Con XAMPP ha sido para lanzar el servidor apache el cual es gratuito y así poder trabajar en localhost, lo único que he tenido que hacer es descargarlo (se adjuntará link al final) y cuando lo inicié arrancó tanto el Apache como el MySQL.



y así se vería para saber si está en funcionamiento o no.



PHPMYADMIN



Con phpmyadmin lo he utilizado para gestionar la base de datos en localhost ya que es gratuito en el cual he creado mi base de datos llamada proyectofinal y las diferentes tablas que son:

La base de datos que se llama proyectofinal
tablas users, purchases, products, messages.

MYSQL



Siendo el sistema de administración de base de datos relacionales de código abierto, desarrollado por Oracle. Se utiliza para manipular datos como en este proyecto.

VSCODE



He utilizado Visual Studio Code que es un editor de código bastante manejable, uno de los más utilizados que tiene extensiones, depuración...

VUE.JS



El framework que he utilizado es este Vue ya que con una sola pagina se puede combinar html, css, script, ademas de que se pueden crear componentes y se pueden usar en otros, la version utilizada es la mas reciente que es la 3.

JS

Lenguaje de programación para el desarrollo de los scripts, en Vue y aplicar ciertas funciones a los botones mas añadir v-if y onclick....

creado por **Brendan Eich** mientras trabajaba en Netscape Communications Corporation



HTML

Para la composición del template que a su vez se ha utilizado bootstrap tambien. La version es html5 desarrollado por el **World Wide Web Consortium (W3C)** y **Web Hypertext Application Technology Working Group (WHATWG)**



CSS

Utilizado para aplicar estilos los cuales con Bootstrap no se podía, colores, márgenes... desarrollado por el **World Wide Web Consortium (W3C)**



POSTMAN

Este software lo he utilizado mas que nada para la depuración de código php cuando me fallaba algo que no encontraba y con la consola del navegador no era capaz de sacarlo, puede enviarse solicitudes GET, POST, PUT, DELETE... a la API, también facilita bastante el hecho de hacer diferentes pruebas, la verdad es que es una aplicación que ayuda bastante al desarrollo y mantenimiento de la app.



BOOTSTRAP

No por ser el ultimo es el menos importante de hecho todos los componentes llevan de este framework con el cual es facil de desarrollar mirando la documentación de la misma y tiene diseños responsivos para telefono, pc... permite diseños flexibles, tambien se puede usar iconos, por ejemplo el que hay en la aplicación que es un ojo para ver u ocultar la contraseña.



GITHUB

GitHub

GitHub la he utilizado para crear un repositorio donde poder guardar las imágenes y de ahí sacar los enlaces (raw) y poder meterlos en mi web además de que cuando demos a añadir u modificar un producto podamos meter ahí una url.

6. BASES DEL PROYECTO

Lo primero que hay que tener en cuenta es que este proyecto tuvo varias tormentas de ideas las cuales ninguna me parecía factible hasta que di con esta la cual era llevar a los comercios tradicionales a la nueva era de la tecnología. Al principio como bien sabe estuve dudando entre usar Wordpress o no a pesar de que era una tecnología nueva pero preferí tirar por el camino que ya conocía aunque no tenía claro como iba a plantearla.

El planteamiento fue primero uno básico con el inicio, nosotros, productos, login y registro y ese extra de meter un carrito y a partir de ahí cuando tuviese esas secciones en funcionamiento ir añadiendo más, por lo que comencé con las funciones, hice una página básica con bootstrap en vue con colores básicos negro, blanco y gris y empecé a darle forma a las funciones.

La pregunta que me hice fue la siguiente ¿Y ahora qué?, cuando tenia ya las funciones que eran muy básicas porque no tenian control de seguridad ni nada me quedé un poco bloqueado, entonces empecé a pensar extras como eran el contacto el cual era un formulario simple para mandar mensajes y en el rol de administrador la manipulación de los productos para agregar, borrar y modificar un producto.

Aun así me parecía bastante básico, eso sí aplicandole por ejemplo un generador de contraseñas el cual mezclaba caracteres especiales, con mayusculas, minusculas y numeros.

```
generatePassword() {
  const length = 16;
  const charset = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()_+";
  let password = "";
  for (let i = 0, n = charset.length; i < length; ++i) {
    password += charset.charAt(Math.floor(Math.random() * n));
  }
  this.password = password;
}
```

Este es el generador de contraseñas con todo incluido con una longitud de 16 caracteres, a dia de hoy se considera segura esa longitud aunque va aumentando para evitar riesgos.

Pero aun así algo mas le faltaba y me acordé que en clase de PHP recordamos el integrar un hash para proteger la contraseña en la base de datos el cual es el siguiente.

```
$password = mysqli_real_escape_string($conn, $data['password']);
$role = mysqli_real_escape_string($conn, $data['role']);
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);
```

Tambien se le añadio para evitar inyecciones mysql

Despues de esto pensé en seguir añadiendo seguridad a mi API por lo que hice que en el login comprobase las credenciales en el navegador y en caso de que no coincidiese saliera un mensaje de que no coinciden.

Seguí pensando en que mas añadir y fue cuando hablando con el profesor de mi proyecto D. Ignacio García Castellano me dijo que intentase crear alguna sección mas o meter mas cosas en un producto como puede ser una imagen por lo que pensé en meter la imagen y que con la url cambiase y se almacenase en la base de datos.

Por otro lado tambien añadí un boton de ver video el cual enlaza con un video a youtube, mas adelante en el recorrido por la aplicación habrá diferentes capturas del código explicandolo.




Hice una nueva sección en el rol de admin llamada Mensajes en la cual se recibiría los mensajes que mandase el usuario desde el formulario de contacto y el admin los podrá ver o incluso borrar que lo que hace es borrarlo de la base de datos (no tiene para contestar porque no me dió tiempo).


Mensajes



En la parte del carrito se verá el listado de productos en caso de que el particular, es decir, el cliente quiera añadir un producto tendrá el botón de pagar a la derecha y si le da se despliega un formulario el cual le pedirá el numero de tarjeta, el cvc, telefono y que acepte las politicas de tratamiento de datos y privacidad.

Método de Pago





Número de Tarjeta:

Fecha de Caducidad (MM/AA):

CVC:

Teléfono de contacto:

Email de contacto:

☐ Acepto la [política de privacidad](#) de NEW ERA SHOP

☐ Acepto el [tratamiento de datos](#) por parte de NEW ERA SHOP

Pedido:

Curso de WordPress + Elementor Pro - 1 x 249.99

Total a pagar: 249.99

Realizar Pago

```
// Validar y sanitizar datos del formulario de pago
$phone = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['phone']));
$email = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['email']));
$cardNumber = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['cardNumber']));
$expiryDate = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['expiryDate']));
$cvc = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['cvc']));
$total = array_reduce($cart, function($sum, $item) {
    return $sum + ($item['price'] * $item['quantity']);
}, 0);

// Verificar que el número de tarjeta y el CVC sean válidos
if (!preg_match("/^\d{16}$/", $cardNumber) || !preg_match("/^\d{3}$/", $cvc)) {
    echo json_encode(["success" => false, "message" => "Datos de pago inválidos"]);
    exit;
}

// Actualizar stock de productos
foreach ($cart as $item) {
    $productId = $item['id'];
    $quantity = $item['quantity'];

    // Verificar si hay suficiente stock
    $sql = "SELECT quantity FROM products WHERE id = ?";
    $stmt = $conn->prepare($sql);
    if (!$stmt) {
        echo json_encode(["success" => false, "message" => "Error al preparar la consulta (stock): " . $conn->
error]);
        exit;
    }
    $stmt->bind_param("i", $productId);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows == 0) {
        echo json_encode(["success" => false, "message" => "Producto no encontrado: " . $item['name'] . " (ID: " .
$productId . ")"]);
    }
}
```

Este es parte del código el cual hace el proceso del pago, como he dicho antes en todos los archivos donde haya que introducir datos esta la seguridad de sanación para evitar ataques de inyecciones, actualización del stock, porque elimina un producto de la base de datos cada vez que se vende para que no haya problemas y haga compras de productos sin haber ninguno, también tiene una validación de tarjeta y cvc que solo hace comprobar que haya 16 dígitos y el cvc sean 3.

Además las compras se registrarían en la base de datos en la tabla purchases con los datos proporcionados por el usuario en el formulario del carrito.

	id	phone	email	cardNumber	expiryDate	cvc	total	cart	purchase_date
<input type="checkbox"/>	1	999999999	fulano@mail.com	1234567874563236	09/47	752	249.99	[[{"id":"3","name":"Curso de WordPress + Elementor ...	2024-06-13 21:19:46

Y por ultimo y para añadir algo mas a mi API cree la sección usuarios en la cual al admin le saldría todos los que hay registrados y en esto puede modificar diferentes campos como son el nombre, el telefono, apellidos...

NEW ERA SHOP

[Inicio](#)
[Nosotros](#)
[Contacto](#)
[Productos](#)
[Carrilo](#)
[Mensajes](#)
[Usuarios](#)
[Hola, admin \(Rol: admin\)](#)
[Cerrar sesión](#)

Lista de Usuarios

Nacho ramiro

[Ver Detalles](#)
[Modificar](#)
[Eliminar](#)

Email: fulano@mail.com

Teléfono: 777777777

Rol: particular

admin admin

[Ver Detalles](#)
[Modificar](#)
[Eliminar](#)

Email: admin@mail.com

Teléfono: 666666666

Rol: admin

7. MÉTODO

En este apartado procederé a explicar todo el código parte por parte tanto en la interfaz como en la programación del mismo y explicando el porque.

7.1 LOGIN

El login como su propio nombre indica es para iniciar una sesion de usuario, lo mas normal es que ya este registrado pero en caso de que no lo este en mi API le saldrá lo siguiente.



The image shows a web form for logging in. At the top is a blue header with the text 'Iniciar Sesión'. Below this are two input fields. The first is labeled 'Email:' and contains the text 'sdsds@s.com'. The second is labeled 'Contraseña:' and contains masked characters '..'. To the right of the password field is a small icon of an eye. Below the password field is a red rectangular box with the text 'Usuario no encontrado.' in red. At the bottom of the form is a blue button with the text 'Iniciar Sesión'.

En este caso es usuario no encontrado porque no esta registrado ese email pero en caso de que si lo estuviera nos inicia sesion. Para ello en el archivo de login tendremos como como podemos comprobarlo, con una sentencia preparada para evitar ataques.


```
// Preparación de la consulta SQL para verificar si el usuario existe en la base de datos
$sql = "SELECT * FROM users WHERE email=?";
$stmt = $conn->prepare($sql);
if ($stmt === false) {
    die("Error preparing statement: " . $conn->error);
}

// Comprobación si se encontró el email
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

// Comprobación contraseña si coincide o no
if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    if (password_verify($password, $user['password'])) {
        echo json_encode(array("success" => true, "name" => $user['first_name'], "role" => $user['role']));
    } else {
        echo json_encode(array("success" => false, "message" => "Contraseña incorrecta."));
    }
} else {
    echo json_encode(array("success" => false, "message" => "Usuario no encontrado."));
}

$stmt->close();
$conn->close();
} else {
    echo json_encode(array("success" => false, "message" => "Por favor, proporciona email y contraseña."));
}
}
```

Como vemos con los comentarios tenemos que hace cada cosa, esta la sentencia preparada para consultar en la tabla users en la base de datos para saber si esta el email o no y despues de esto comprobar si la contraseña coincide con el email que hay registrado



En este caso podemos observar que como el email si esta registrado pero la contraseña es incorrecta nos dice contraseña incorrecta.

Al poner las credenciales bien nos deja meternos y nos saldrá tal que así.

Hola, Nacho (Rol: particular) Cerrar sesión

7.2 REGISTRO

El registro tiene diferentes campos en un formulario ya que puede haber dos tipos de registro el de particular y el de admin.

Al ir a registro nos encontraremos con esto.

Registro

Nombre: ⓘ
El nombre es obligatorio

Apellidos: ⓘ
Los apellidos son obligatorios

Teléfono de contacto: ⓘ
El teléfono es obligatorio

Email de contacto: ⓘ
El email es obligatorio

Contraseña: ⓘ Generar Contraseña ⓘ

Rol:
Particular ✓

Registrar

¿Ya tienes una cuenta?
Ir al Login

Lo cual nos dice que los campos son obligatorios, esto se hace con el v-if de vue que es decir si no hay que diga ese mensaje pero si pone alguna pues nada.

```
<div class="invalid-feedback text-danger" v-if="!password">La contraseña es obligatoria</div>
```

El primer error que podemos tener es el siguiente, por ejemplo el email que ya está en uso como es en este caso fulano@mail.com por lo que nos saltara el mensaje de error al hacer la comprobación

```
$duplicateMessage = '';
if (checkDuplicate($conn, 'email', $email)) {
    $duplicateMessage .= 'Este correo electrónico ya está en uso. ';
}
if (checkDuplicate($conn, 'phone', $phone)) {
    $duplicateMessage .= 'Este número de teléfono ya está en uso. ';
}

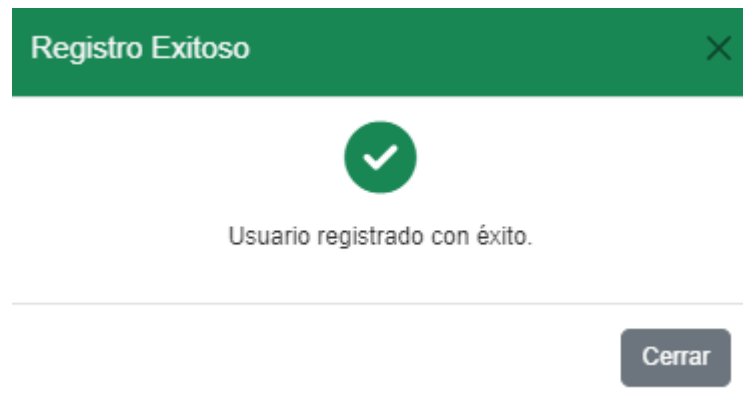
// Inicializar las variables de empresa
$companyName = '';
$companyCIF = '';
$companyAddress = '';

if ($role === 'admin') {
    if (isset($data['companyName']) && isset($data['companyCIF']) && isset($data['companyAddress'])) {
        $companyName = mysqli_real_escape_string($conn, $data['companyName']);
        $companyCIF = mysqli_real_escape_string($conn, $data['companyCIF']);
        $companyAddress = mysqli_real_escape_string($conn, $data['companyAddress']);

        if (checkDuplicate($conn, 'company_name', $companyName)) {
            $duplicateMessage .= 'Este nombre de empresa ya está en uso. ';
        }
        if (checkDuplicate($conn, 'company_cif', $companyCIF)) {
            $duplicateMessage .= 'Este CIF de empresa ya está en uso. ';
        }
    }
}
```

En este código que es el register.php aparte de otros tipos de comprobaciones como son el teléfono o el CIF de la empresa también está el de correo electrónico en uso.

En caso de tener todas las credenciales bien se nos registra y nos saldrá lo siguiente:



Que en nuestra base de datos nos tiene que salir como registrado, lo que se ha hecho ha sido en este caso cambiar el correo electrónico que era el unico error por estar duplicado.

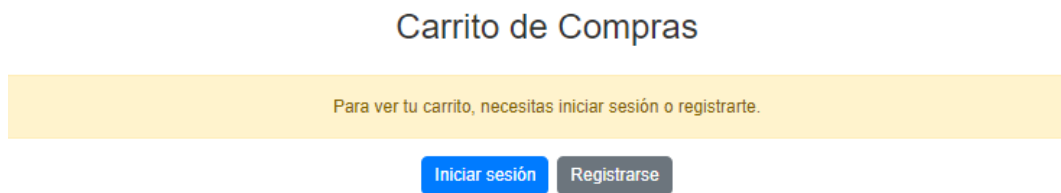
```
3 Nacho martinez 999999999 fulano23@mail.com $2y$10$OQbchINBQBHL5HR5s3RAjOZ8Gau6QvGVQTxamE4zYCC
```

Como podemos observar la contraseña esta protegida con un hash para evitar robos faciles de la misma que se han puesto en el codigo de register.php aparte de la protección contra caracteres especiales en el email y demás.

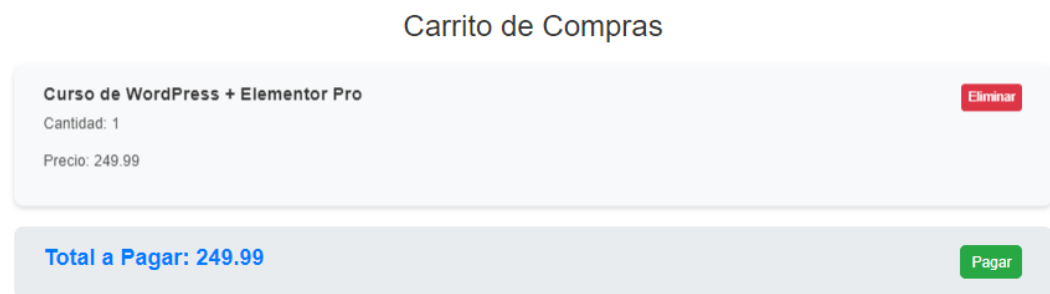
Seguimos con la parte de fuera de la API, ahora veremos las diferentes secciones que hay como son Carrito.

7.3 CARRITO

Lo primero de todo si no esta registrado nos saldrá como en la siguiente imagen.




Por lo que si le damos a un botón u otro nos llevará a iniciar sesion o registrarse, una vez dentro si podremos ver nuestro carrito como es la siguiente imagen.




En este caso podremos eliminar del carrito o pagar, lo hemos hecho de la siguiente manera.

En caso de que cerremos sesión se borrará el carrito, es decir el localStorage de ese usuario se borrará para que si se loguea otro no se quede registrado.

Método de Pago





Número de Tarjeta:

1234567876454321

Fecha de Caducidad (MM/AA):

08/24

CVC:

123

Teléfono de contacto:

965847451

Email de contacto:

fulano@mail.com

☒ Acepto la [política de privacidad](#) de NEW ERA SHOP

☒ Acepto el [tratamiento de datos](#) por parte de NEW ERA SHOP

Pedido:

Curso de WordPress + Elementor Pro - 1 x 249.99

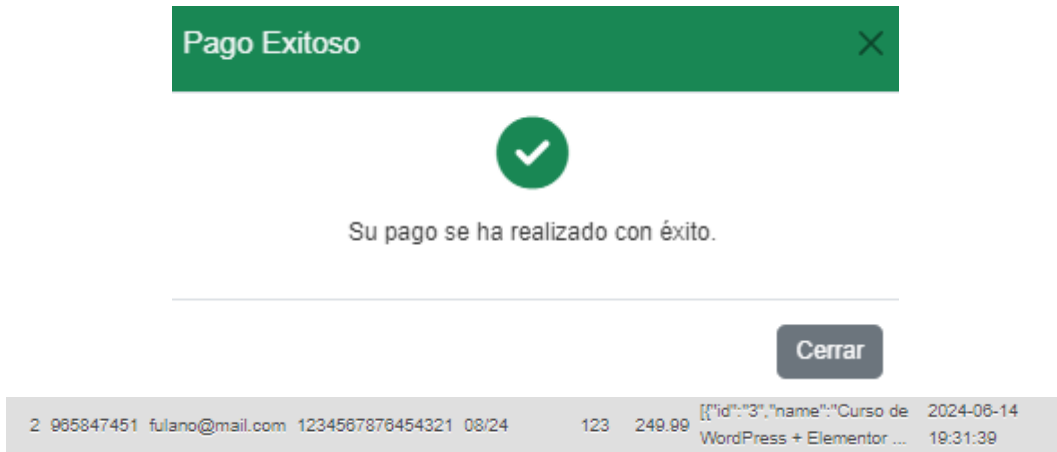
Total a pagar: 249.99

Realizar Pago

Al darle a pagar para comprar los productos nos saldra este formulario que debe completar con los datos que sea del usuario, comprobación de que tenga 16 dígitos, con el cvc y la fecha de caducidad, el teléfono que cumpla con los 9 dígitos y el email asociado en caso de no aceptar las dos casillas no nos dejará continuar ya que el botón no estara disponible.

Si hacemos la venta se registrara en la tabla de purchases como vemos aquí abajo.

Nos saldrá este modal.



Aquí podemos ver el registro de la compra con la fecha en la que se hizo por si hubiese alguna incidencia con la tienda que tenga esta página y el cliente.

```
// Validar y sanitizar datos del formulario de pago
$phone = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['phone']));
$email = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['email']));
$cardNumber = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['cardNumber']));
$expiryDate = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['expiryDate']));
$cvc = mysqli_real_escape_string($conn, htmlspecialchars($paymentDetails['cvc']));
$total = array_reduce($cart, function($sum, $item) {
    return $sum + ($item['price'] * $item['quantity']);
}, 0);

// Verificar que el número de tarjeta y el CVC sean válidos
if (!preg_match("/^\d{16}$/", $cardNumber) || !preg_match("/^\d{3}$/", $cvc)) {
    echo json_encode(["success" => false, "message" => "Datos de pago inválidos"]);
    exit;
}

// Actualizar stock de productos
foreach ($cart as $item) {
    $productId = $item['id'];
    $quantity = $item['quantity'];

    // Verificar si hay suficiente stock
    $sql = "SELECT quantity FROM products WHERE id = ?";
    $stmt = $conn->prepare($sql);
    if (!$stmt) {
        echo json_encode(["success" => false, "message" => "Error al preparar la consulta (stock): " . $conn->error]);
        exit;
    }
    $stmt->bind_param("i", $productId);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows == 0) {
        echo json_encode(["success" => false, "message" => "Producto no encontrado: " . $item['name'] . " (ID: " . $productId . ")"]);
    }
}
```

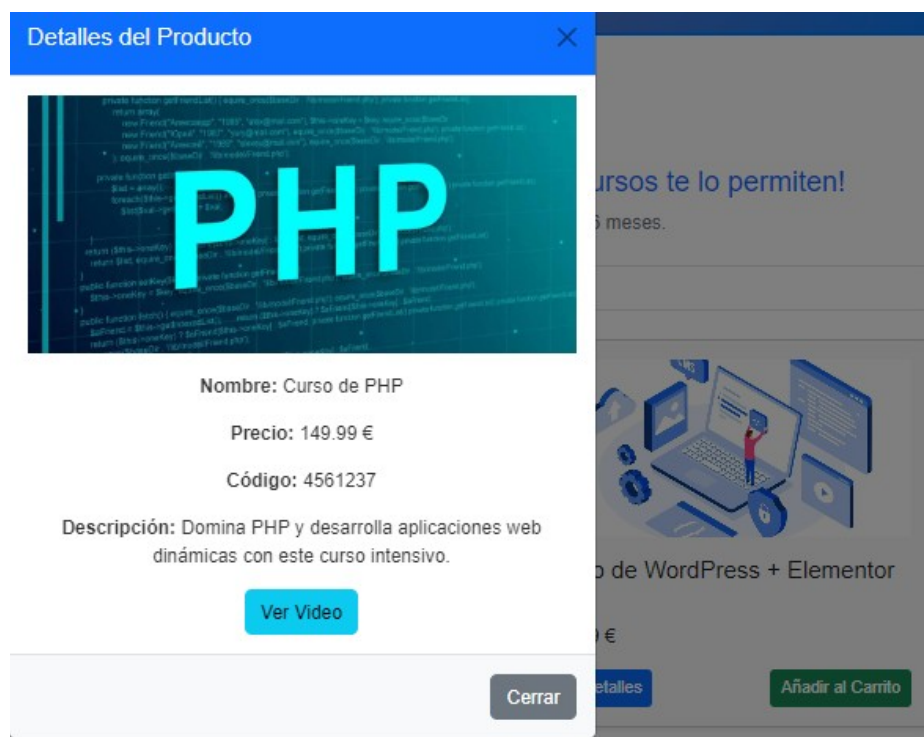
Como podemos ver en la imagen hay parte del código donde se puede observar la sanación contra ataques e inyecciones y abajo esta lo de verificar el número de tarjeta y cvc además de que actualiza el stock para evitar comprar productos que no existen.

7.4 PRODUCTOS

Este ha sido uno de las secciones mas complejas para mi ya que habia que hacer que el rol de particular, es decir, el usuario tenga que ver unas cosas y el administrador otras.

Empezamos por el particular el cual tiene un buscador para filtrar por nombre y en cada producto le aparece 2 botones, uno de ver detalles y otro de añadir al carrito, en caso de no estar login no aparecerá el botón de añadir al carrito.

Si le damos a ver detalles saldrá lo siguiente.



Es un modal en el cual se muestran los detalles del producto que se han cogido de la base de datos para mostrarlos al usuario, como se puede ver hay un botón que se llama ver video que es un enlace a youtube en este caso a no ser que la persona interesada en la página ponga uno almacenado en otro lado y si le damos nos lleva a ese video, por lo demás es genérico.

Se coge de la base de datos de esta tabla en concreto.

id	name	price	quantity	codeProduct	description	video	image
1	Curso de Vue.js	199.99	50	1234567	Aprende Vue.js desde cero y convierte en un expert...	https://www.youtube.com/watch?v=AprJzDvbpgI	https://raw.githubusercontent.com
2	Curso de PHP	149.99	30	4561237	Domina PHP y desarrolla aplicaciones web dinámicas...	https://www.youtube.com/watch?v=ykGRYEX0n60	https://raw.githubusercontent.com
3	Curso de WordPress + Elementor Pro	249.99	14	9632584	Construye sitios web profesionales con WordPress y...	https://www.youtube.com/watch?v=1PqZ7JBFW3I	https://raw.githubusercontent.com
4	Curso de Marketing Digital	299.99	25	2354796	Conviértete en un experto en marketing digital y a...	https://www.youtube.com/watch?v=CPNez1ZDXD0	https://raw.githubusercontent.com
5	Curso de SEO	179.99	40	7391465	Mejora el posicionamiento de tu sitio web en los m...	https://www.youtube.com/watch?v=AavsQP3gPj4	https://raw.githubusercontent.com

En caso del rol admin que es el que vamos a ver ahora tendremos otros botones diferentes que son los siguientes.



Como se puede apreciar hay 3 botones pero no se encuentra el de añadir a carrito porque no hace falta, tenemos el de borrar que directamente se borra de la base de datos, tenemos el de modificar que se puede hacer cualquier cambio pero con un condicionante y es que el código de producto si se cambia no puede coincidir con otro porque entonces serian productos duplicados.

Este código ya existe en otro producto.

Código de Producto:

Descripción:

```
// Obtener el código de producto enviado desde el frontend
$data = json_decode(file_get_contents("php://input"), true);

if (isset($data['code'])) {
    $code = $conn->real_escape_string($data['code']);

    // Verificar si el código de producto ya está en uso
    $stmt = $conn->prepare("SELECT COUNT(*) AS count FROM products WHERE codeProduct = ?");
    $stmt->bind_param("s", $code);
    $stmt->execute();
    $result = $stmt->get_result();
    $count = $result->fetch_assoc()['count'];

    // Enviar la respuesta al frontend
    echo json_encode(['exists' => ($count > 0)]);

    $stmt->close();
} else {
    echo json_encode(['success' => false, 'message' => 'Código de producto no proporcionado']);
}

$conn->close();
?>
```

En esta imagen se puede observar el código encargado de ver si esta ya ese código de producto o no, en caso de que este pues saldrá el mensaje anterior.

También podemos modificar otras cosas como se puede ver en este modal.

Editar Producto ✕

Nombre:

Precio:

Cantidad:

Código de Producto:

Descripción:

URL del Video:

URL de la Imagen:

Actualizar

Hay scripts dedicados para el update del producto o para el de add product o delete product como son los siguientes.

update

```
if (isset($data['id']) && isset($data['name']) && isset($data['price']) && isset($data['quantity']) &&
isset($data['codeProduct']) && isset($data['description']) && isset($data['video']) && isset($data['image'])) {
    $id = $conn->real_escape_string($data['id']);
    $name = $conn->real_escape_string($data['name']);
    $price = $conn->real_escape_string($data['price']);
    $quantity = $conn->real_escape_string($data['quantity']);
    $codeProduct = $conn->real_escape_string($data['codeProduct']);
    $description = $conn->real_escape_string($data['description']);
    $video = $conn->real_escape_string($data['video']);
    $image = $conn->real_escape_string($data['image']);

    // Verificar si el código del producto ya existe y pertenece a otro producto
    $checkSql = "SELECT COUNT(*) as count FROM products WHERE codeProduct = '$codeProduct' AND id != '$id'";
    $checkResult = $conn->query($checkSql);
    $row = $checkResult->fetch_assoc();

    if ($row['count'] > 0) {
        echo json_encode(["success" => false, "message" => "Este código ya existe en otro producto."]);
    } else {
        $sql = "UPDATE products SET name='$name', price='$price', quantity='$quantity', codeProduct='$codeProduct',
description='$description', video='$video', image='$image' WHERE id='$id'";

        if ($conn->query($sql) === TRUE) {
            echo json_encode(["success" => true, "message" => "Producto actualizado con éxito."]);
        } else {
            echo json_encode(["success" => false, "message" => "Error al actualizar el producto: " . $conn->error]);
        }
    }
} else {
    echo json_encode(["success" => false, "message" => "No se han recibido todos los datos necesarios"]);
}
```

En este caso tambien debe comprobar si el codigo de producto está ya en uso o no porque es cuando se esta actualizando y sale el mismo error.

Delete

en la siguiente hoja.

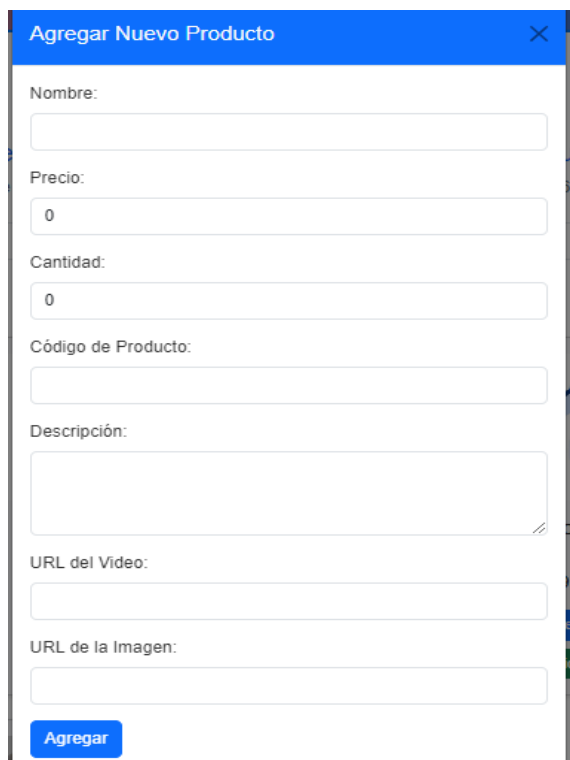
```
// Preparar y ejecutar la consulta para eliminar el producto
$sql = "DELETE FROM products WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $data['id']);

if ($stmt->execute()) {
    // Si la consulta se ejecuta correctamente, enviar una respuesta exitosa
    echo json_encode(array("success" => true, "message" => "Producto eliminado exitosamente"));
} else {
    // Si hay un error al ejecutar la consulta, enviar un mensaje de error
    echo json_encode(array("success" => false, "message" => "Error al eliminar el producto"));
}

// Cerrar la conexión a la base de datos
$stmt->close();
$conn->close();
} else {
    // Si no se recibió correctamente el ID del producto a eliminar, enviar un mensaje de error
    echo json_encode(array("success" => false, "message" => "ID del producto no especificado"));
}
} else {
    // Si la solicitud no es del tipo POST, enviar un mensaje de error
    echo json_encode(array("success" => false, "message" => "Método no permitido"));
}
}
```

Es la sentencia ya prepara de borrar el producto de una tabla típica en mysql.

El admin tiene la opción de agregar producto el cual se desplegará lo siguiente y si añade el producto se añade a la base de datos.



Formulario "Agregar Nuevo Producto" con los siguientes campos:

- Nombre:
- Precio:
- Cantidad:
- Código de Producto:
- Descripción:
- URL del Video:
- URL de la Imagen:
- Botón: Agregar

```
if (isset($data->name) && isset($data->price) && isset($data->quantity) && isset($data->codeProduct) &&
isset($data->description) && isset($data->video) && isset($data->image)) {
    $name = $conn->real_escape_string($data->name);
    $price = $conn->real_escape_string($data->price);
    $quantity = $conn->real_escape_string($data->quantity);
    $codeProduct = $conn->real_escape_string($data->codeProduct);
    $description = $conn->real_escape_string($data->description);
    $video = $conn->real_escape_string($data->video);
    $image = $conn->real_escape_string($data->image);

    // Verificar si el código de producto ya está en uso
    $checkSql = "SELECT COUNT(*) AS count FROM products WHERE codeProduct = '$codeProduct'";
    $checkResult = $conn->query($checkSql);
    $count = $checkResult->fetch_assoc()['count'];

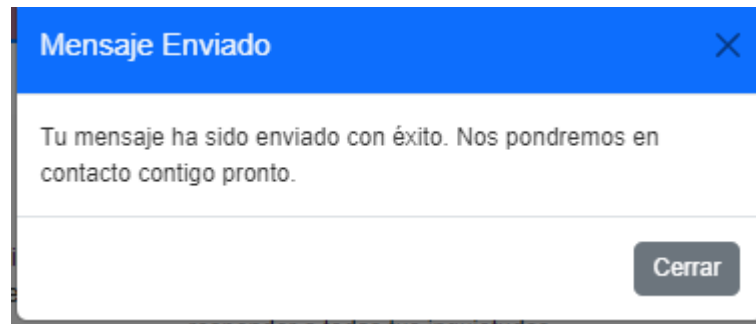
    if ($count > 0) {
        echo json_encode(["success" => false, "message" => "Este código de producto ya está en uso."]);
    } else {
        $sql = "INSERT INTO products (name, price, quantity, codeProduct, description, video, image) VALUES
('$name', '$price', '$quantity', '$codeProduct', '$description', '$video', '$image')";

        if ($conn->query($sql) === TRUE) {
            echo json_encode(["success" => true, "message" => "Producto agregado con éxito."]);
        } else {
            echo json_encode(["success" => false, "message" => "Error al agregar el producto: " . $conn->error]);
        }
    }
} else {
    echo json_encode(["success" => false, "message" => "No se han recibido todos los datos necesarios"]);
}
```

En este script tambien hace lo mismo ya que al añadir producto tiene que comprobar que el codigo no este en uso y en caso de que no lo este pues lo inserta en la tabla de products como se ve en la sentencia preparada que hay para evitar ataques a la base de datos.

7.5 CONTACTO

En esta seccion es un formulario simple con un texto típico de que si tiene consulta o algo por el estilo que nos contacte, se le pide el nombre, apellidos, telefono, email y el rol en caso de que sea empresa se le pide el nombre de la empresa y abajo para los dos roles se pide el asunto y el mensaje, una vez que se envia el admin lo recibirá en su apartado creado para mensajes que a su vez esto se guarda en la base de datos.



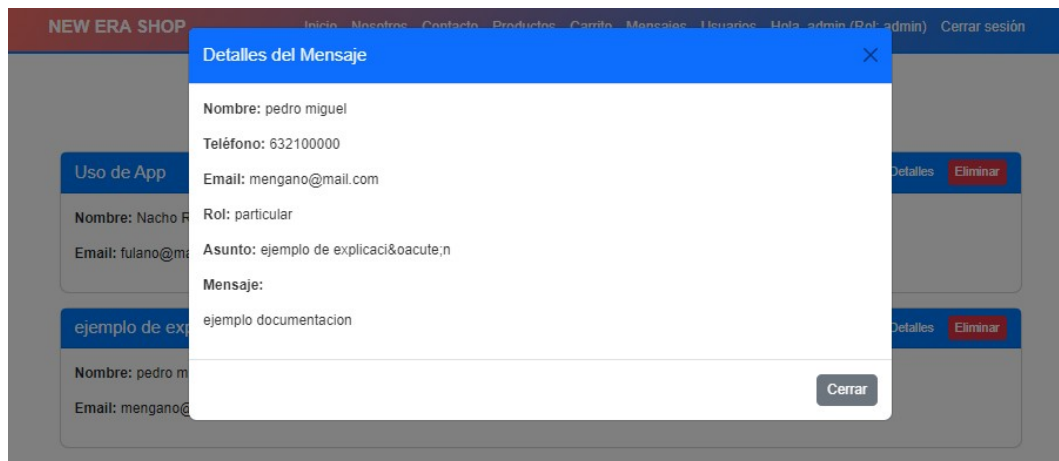
Como se puede observar una vez enviado le aparecera este modal diciendo que se ha enviado y que pronto se pondrá en contacto con el, si nos vamos a la base de datos nos aparecerá este mensaje registrado.

name	surname	phone	email	role	company_name	subject	message	created_at
Nacho	Ramiro	999999999	fulano@mail.com	particular	NULL	Uso de App	mensaje enviado desde un particular para comprobar...	2024-06-13 21:21:02
pedro	miguel	632100000	mengano@mail.com	particular	NULL	ejemplo de explicación	ejemplo documentacion	2024-06-14 19:55:19

Con su fecha tambien para evitar el típico problema de te lo he enviado y no lo has recibido pues aquí se sabe con seguridad si ha llegado y que día y hora fue el recibirlo, ahora lo veremos en la sección de mensaje como aparece.

7.6 MENSAJES

Esta sección solo le aparecerá al admin ya que es como un correo interno para el donde puede ver los mensajes y los detalles del mismo como vemos en la imagen.



De fondo se puede observar que tiene el boton de ver detalles y al lado el de borrar que como su propio nombre indica lo elimina de la base de datos y por otro lado el de ver detalles nos muestra el mensaje asi como vemos ahi, importante recalcar que por la seguridad que tiene de escape string hace que los simbolos como las tildes las ponga con esa expresi&on, esto se puede cambiar si el cliente quiere con quitar ese ajuste.

```
if (isset($data['name']) && isset($data['surname']) && isset($data['phone']) && isset($data['email']) &&
isset($data['subject']) && isset($data['message'])) {
    $name = htmlentities($conn->real_escape_string($data['name']));
    $surname = htmlentities($conn->real_escape_string($data['surname']));
    $phone = htmlentities($conn->real_escape_string($data['phone']));
    $email = htmlentities($conn->real_escape_string($data['email']));
    $role = htmlentities($conn->real_escape_string($data['role']));
    $companyName = $role === 'empresa' ? htmlentities($conn->real_escape_string($data['companyName'])) : null;
    $subject = htmlentities($conn->real_escape_string($data['subject']));
    $message = htmlentities($conn->real_escape_string($data['message']));

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo json_encode(['success' => false, 'message' => 'Formato de email inválido']);
        exit;
    }

    if (!preg_match('/^[0-9]{9}$/', $phone)) {
        echo json_encode(['success' => false, 'message' => 'El teléfono debe contener 9 dígitos']);
        exit;
    }

    $sql = "INSERT INTO messages (name, surname, phone, email, role, company_name, subject, message) VALUES
(?, ?, ?, ?, ?, ?, ?, ?)";
    $stmt = $conn->prepare($sql);

    if ($role === 'admin') {
        $stmt->bind_param("sssssss", $name, $surname, $phone, $email, $role, $companyName, $subject, $message);
    } else {
        $stmt->bind_param("sssssss", $name, $surname, $phone, $email, $role, $companyName, $subject, $message);
    }
}
```

Como vemos en este php que es el de contact tenemos lo de htmlentities además del real escape del que he comentado anteriormente, después de esto se introduce en la base de datos de messages y con el código del php de get-messages

```
$sql = "SELECT id, name, surname, phone, email, role, company_name, subject, message FROM messages";
$result = $conn->query($sql);

$messages = array();

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $messages[] = $row;
    }
}

echo json_encode($messages);

$conn->close();
?>
```

El admin los recibe ahí y con Bootstrap se pone bien y tiene creada su correo interno, faltaría que pudiera contestar al usuario particular y crear al 100% el correo interno pero no llegó el tiempo.

7.5 USUARIOS

Esta sección fue creada de ultima hora para añadir algo más para que el admin tuviese todo a su alcance y que menos que la típica gestión de sus usuarios que estan registrados en su plataforma.

Lista de Usuarios

Nacho ramiro	Ver Detalles	Modificar	Eliminar
Email: fulano@mail.com			
Teléfono: 777777777			
Rol: particular			
admin admin	Ver Detalles	Modificar	Eliminar
Email: admin@mail.com			
Teléfono: 666666666			
Rol: admin			
Nacho martinez	Ver Detalles	Modificar	Eliminar
Email: fulano23@mail.com			
Teléfono: 999999999			
Rol: particular			

Como se puede observar a parte del estilo que tiene en tarjeta y demás tenemos 3 botones, ver detalles para ver la informacion de ese usuario como se muestra aquí abajo.

Detalles del Usuario

Nombre: Nacho ramiro

Email: fulano@mail.com

Teléfono: 77777777

Rol: particular

Cerrar

Por otro lado tenemos modificar el cual nos deja tocar todos los datos del usuario que si hacemos un cambio por ejemplo con el de arriba que el telefono es todo 7 vamos a cambiarlo a todo 7 y al final un 0.

Modificar Usuario

Nombre

Nacho

Apellidos

ramiro

Teléfono

77777770

Email

fulano@mail.com

Actualizar

Nacho ramiro [Ver Detalles](#) [Modificar](#) [Eliminar](#)

Email: fulano@mail.com

Teléfono: 777777770

Rol: particular

En la tarjeta aparece cambiado y si vamos a mi base de datos tambien nos aparece cambiado como se pùede ver aquí abajo.

id	first_name	last_name	phone	email	password	role
1	Nacho	ramiro	777777770	fulano@mail.com	\$2y\$10\$z3sBT8VBrbw3Dkr/Sq5rLOjbPywubSHFRLVh9oDGH2Q...	part

Tambien aparecería modificado como se ve ahí con el 0 al final.

Si le da al botón de borrar lo eliminara de la base de datos a ese usuario, como se ve aquí abajo ya no existe ese usuario con ese numero ni email.

Nos sale este modal para decirnos si estamos seguros o no de que queremos borrarlo de la base de datos y ya el admin puede elegir is hacerlo o no.

Confirmar Eliminación ✕

¿Estás seguro de que deseas eliminar al usuario Nacho ramiro?

[Cancelar](#) [Eliminar](#)

7.8 INICIO

¿Quieres aprender en tan solo 6 meses?

¡Únete a nuestra comunidad de estudiantes exitosos! Con nuestros cursos certificados, podrás dominar habilidades clave en marketing digital, programación, SEO, y manejo de redes sociales. Aprovecha esta oportunidad y transforma tu carrera profesional.



Nuestros Cursos

Marketing Digital

Aprende a crear campañas efectivas de marketing digital y lleva tu negocio al siguiente nivel. Nuestros cursos están diseñados para convertirte en un experto en el ámbito digital.

Certificado Oficial

Posicionamiento SEO

Domina las técnicas de SEO y mejora la visibilidad de tu sitio web en los motores de búsqueda. Nuestros cursos te brindarán las herramientas necesarias para destacar en el mundo online.

Certificado Oficial

Programación

Convértete en un desarrollador profesional con nuestros cursos de programación. Aprende desde los conceptos básicos hasta las tecnologías más avanzadas.

Certificado Oficial

Manejo de Redes Sociales

Aprende a gestionar y optimizar el uso de redes sociales para empresas y marcas personales. Nuestros cursos te ayudarán a convertirte en un experto en redes sociales.

Certificado Oficial

[Ver Todos los Cursos](#)

Esta es la página de inicio que sale cuando se ejecuta el App.vue, un poco de texto de enganche, como ejemplo y con un botón que si le damos nos lleva a la pestaña de productos.

7.9 NOSOTROS

Sobre Nosotros

Descubre nuestra historia y lo que nos impulsa a ofrecerte lo mejor en formación online.



Quiénes Somos

En New Era Shop, somos un equipo apasionado por la educación y la tecnología. Nuestra misión es proporcionar cursos de alta calidad en marketing digital, programación, SEO y manejo de redes sociales, capacitando a nuestros estudiantes para alcanzar el éxito en sus carreras.

Desde nuestros humildes comienzos, hemos crecido hasta convertirnos en una referencia en la formación online, con miles de estudiantes satisfechos que han transformado sus vidas profesionales gracias a nuestros cursos.

Nuestra Misión

Queremos democratizar el acceso a la educación de calidad, ofreciendo cursos accesibles y actualizados que permitan a nuestros estudiantes desarrollar habilidades clave para el mundo digital de hoy.

Por Qué Elegirnos

- **Expertos en la materia:** Nuestros instructores son profesionales experimentados en sus campos.
- **Flexibilidad:** Aprende a tu propio ritmo, desde cualquier lugar y en cualquier momento.
- **Actualizaciones constantes:** Nuestros cursos se actualizan regularmente para incluir las últimas tendencias y tecnologías.



Formación de Calidad

Cursos diseñados por expertos para garantizar tu éxito profesional.



Acceso 24/7

Accede a nuestros cursos en cualquier momento y desde cualquier lugar.



Satisfacción Garantizada

Tu éxito es nuestra prioridad. Nos esforzamos por brindarte la mejor experiencia de aprendizaje.

Esta sección esta dedicada a nosotros en la cual se hablaría de la empresa, típica página para darse a conocer a los nuevos usuarios.

7.9 FOOTER + COOKIES

El footer de la página es este.



Es modificable pero lo interesante es que tiene la dirección con el telefono a la izquierda, en medio las diferentes redes sociales y a la derecha el mapa que casi todas tienen para ver justamente donde esta por si la calle no la conoces, abajo los créditos y el copyright.

Las cookies son estas.

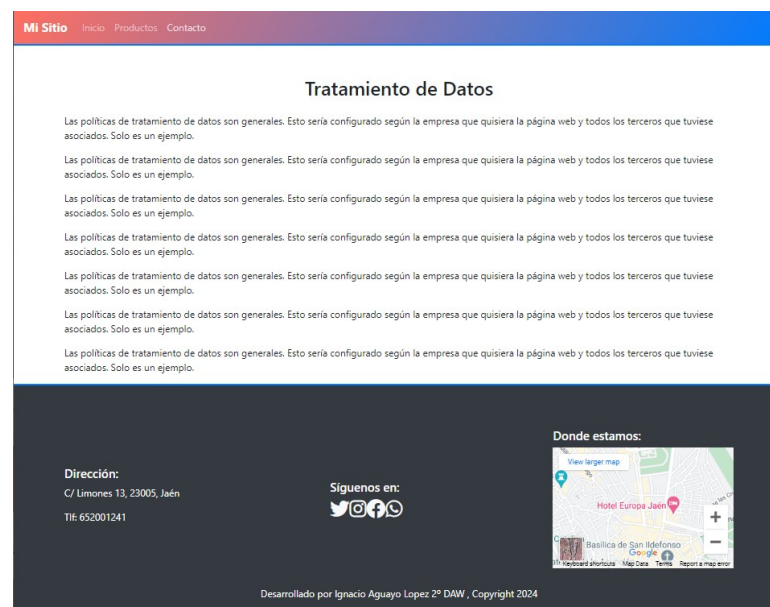


Tiene las 3 opciones de aceptar, aceptar esenciales o rechazarlas todas en caso de que la persona interesada quisiera configurarlas habria que añadir personalizarlas.

7.10 P. PRIVACIDAD + T. DATOS.



Esta es la página de la politica de privacidad que sale en el formulario de pago para que el usuario que va a pagar la vea, deberá estar configurada con la empresa que se encargue y use esta API.



8. TIEMPO DE EJECUCIÓN

F. INVESTIGACIÓN	MAQUETACIÓN	IMPLEMENTACIÓN	FIX ERRORES Y DOCUMENTACIÓN
19 Marzo – 19 Abril	20 Abril – 10 Mayo	11 Mayo – 30 Mayo	1 Junio – 13 Junio
4 -5 Horas/dia	3 – 4 Horas/dia	7 - 8 Horas/dia	5 -6 Horas/dia
150 horas	80 Horas	+/- 152 Horas	72 Horas
Total Horas:	454 Horas		

Tengo que decir que los tiempos de investigación son largos ya que por lo comentando anteriormente en lo del titulo se me truncó todo y tuve que pensar en algo mas hasta que di con los comercios aparte de investigar si alguna tienda tenia implementado lo que queria llegar yo a conseguir.

Por otra parte tuve bastante errores y en la implementación se me alargó bastante debido a que por seguridad para mi y tener las cosas mejor cada archivo php para cada función va por separado y con concetividad de base de datos cada una, me hubiera gustado hacerlo todo en uno pero me fue imposible.

9. CONCLUSIÓN

Pensandolo bien este apartado me permite decir varias cosas acerca de mi API por una parte estoy bastante contento de lo que he sido capaz de hacer con lo aprendido en clase pero por otro lado siento que si hubiera tenido mas tiempo o a lo mejor tendria que ser mas atrevido y haber cogido otra tecnologia podia haber sacado mas potencial de mi, jugó en mi contra que no pude aplicar nada de la empresa puesto que hablamos de ello de crear un plugin, pregunté a trabajadores de la misma y me dijeron que es

complicado y que se tarda bastante sin que te de fallos en crearlo, aunque luego aun habiendolo creado podria dar alguno.

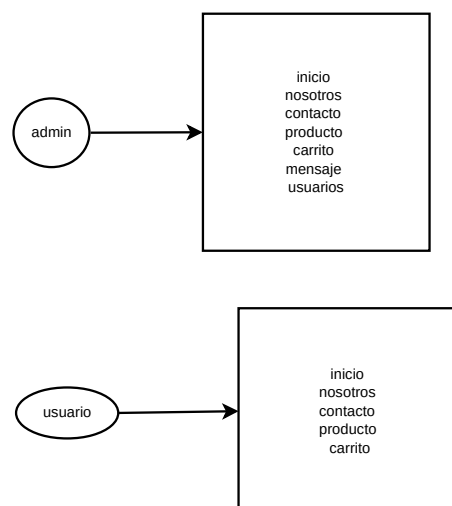
Por lo demás espero que les guste la API ya que es algo que bajo mi punto de vista esta en auge los e-commerce y pensé que seria buena idea hacer una con PHP, Vue.js, Bootstrap...

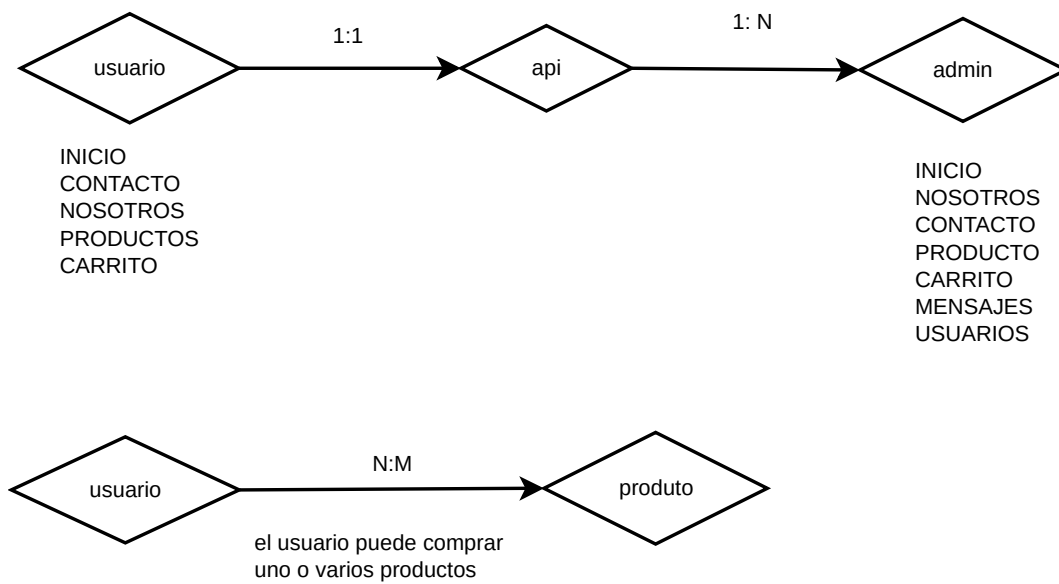
Cabe destacar que tuve demasiados problemas con los php por eso los separé y me acordé de usar Postman para ello, si no yo creo que hubiera sido dificil implementar algunas cosas, aunque sigue faltando secciones completas como puede ser en el rol de particular poner mensajes para que cuando el admin quiera contestarle que le llegue ahí...

O por otro lado poner que si el particular compra un curso hacer una sección que se llame mis cursos y apareciese ahí como comprado y con 3-4 links de youtube de ejemplo.

Pero por todo lo hablado y documentado estoy satisfecho con mi API, me ha gustado como ha quedado y el proceso ha sido en momentos muy malo y en otros bastante bueno al ver como me salian las cosas.

10. DIAGRAMAS





11. ANEXOS

- Instalar XAMMP: <https://www.apachefriends.org/download.html>
- Instalar Vue.js: <https://es.vuejs.org/v2/guide/installation.html>
- Instalar Bootstrap 5: En la consola con **npm install bootstrap**
- Node.js: <https://nodejs.org/en>
- Github: Es online, <https://github.com/>
- Apache y MySQL ha sido con el XAMPP
- VSCODE: <https://code.visualstudio.com/>
- Postman: <https://www.postman.com/>

12. WEBGRAFÍA

-Bootstrap: Documentación

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

- PHP cabeceras (por problemas con CORS y en el mismo foro prefligth)

<https://stackoverflow.com/questions/8719276/cross-origin-request-headerscors-with-php-headers>

- Generador de Texto e Imagenes

<https://www.bing.com/chat?q=Microsoft+Copilot&FORM=hpcodx>

- CSS: No hay sitio oficial pero los estilos se puede consultar aquí.

<https://www.w3schools.com/Css/>

JS

- <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>

- https://www.w3schools.com/js/js_functions.asp

Vue.js

- <https://vuejs.org/guide/introduction.html>

Postman

- <https://www.postman.com/downloads/>