

TFG

Franquicia PC Componentes Omar



## Índice

TFG.....	1
Franquicia PC Componentes Omar .....	1
Español: .....	4
English: .....	4
Enlace al repositorio de Github: .....	4
1.    Justificación .....	5
1.1.    Características .....	5
1.2.    Restricciones generales .....	5
1.3.    Aspectos a cubrir .....	6
1.4.    Estudio de las prestaciones de la herramienta.....	6
2.    Tecnología empleada .....	6
3.    Requerimientos de hardware y software .....	7
4.    Análisis y diseño .....	8
4.1.    Diagrama de casos de uso.....	8
4.2.    Diagrama de clases.....	9
4.3.    Descripción de la base de datos.....	9
4.4.    Modelo entidad-relación .....	10
4.5.    Diseño lógico .....	11
5.    Implementación .....	12
5.1.    Estructura general del proyecto .....	12
5.2.    Rutas y Navegación .....	13
5.3.    Hojas de Estilo .....	13
5.4.    Plantillas y Vistas.....	13
5.5.    Formularios .....	13
5.6.    Envío y Procesamiento de Datos .....	14
5.7.    Seguridad .....	14
5.8.    Conexión a Base de Datos .....	14
5.9.    Consultas y Modelos .....	15
5.10.    Fichero de Configuración .....	15
5.11.    Librerías y Herramientas Usadas.....	15
5.12.    Validaciones y Experiencia de Usuario.....	15
5.13.    Buenas Prácticas Implementadas .....	16
6.    Evaluación y prueba .....	16
7.    Manual de estilos .....	17
7.1.    Sketches.....	17
7.2.    Criterios de accesibilidad y usabilidad .....	25

7.3.	Tipografía .....	26
7.4.	Mapa de colores .....	27
7.5.	Dispositivos / Vistas .....	27
8.	Software utilizado .....	28
9.	Mejoras posibles y aportaciones .....	28
9.1.	Funcionalidades o mejoras que se podrían haber añadido .....	28
9.2.	Organización de código .....	29

## Español:

En este documento se muestra el desarrollo de una tienda online especializada en la venta de productos o dispositivos informáticos similar a la de “PC Componentes”, simulando ser una franquicia de dicha empresa.

Para el desarrollo de dicha tienda, se han utilizado una serie de tecnologías como HTML, CSS, PHP, MySQL, Composer y librerías, con la estructura Modelo-Vista-Controlador (MVC).

La tienda online es totalmente funcional, y se caracteriza por contener funcionalidades como comprar productos con un método de pago por tarjeta bancaria, gestionar usuarios, pedidos, categorías y usuarios, envío de correos electrónicos al confirmar o cancelar pedidos... Todo esto con vistas intuitivas y agradables tanto para los usuarios como para los administradores.

A lo largo de este documento, se desglosará cada apartado explicando su funcionamiento y lógica.

## English:

This document shows the development of an online store specialized in the sale of products or computer devices similar to "PC Componentes", pretending to be a franchise of such company.

For the development of this store, a number of technologies such as HTML, CSS, PHP, MySQL, Composer and libraries have been used, with the structure Model-View-Controller (MVC).

The online store is fully functional, and is characterized by containing functionalities such as buying products with a payment method by bank card, manage users, orders, categories and users, sending emails when confirming or cancelling orders... All this with intuitive and pleasant views for both users and administrators.

Throughout this document, each section will be broken down by explaining its operation and logic.

## Enlace al repositorio de Github:

<https://github.com/omar8910/proyectoOmarPHPFinal>

## 1. Justificación

### 1.1. Características

La tienda online tiene el objetivo de vender productos informáticos o dispositivos electrónicos, ya sea para uso personal o empresarial.

Al acceder a la página, sin necesidad de estar iniciado sesión, se observa una serie de productos destacados, obtenidos aleatoriamente, con la posibilidad de ver con más detalle cada producto seleccionado en su vista correspondiente.

Podemos seleccionar productos filtrados por categoría en un submenú en la página principal.

Se puede añadir productos al carrito sin necesidad de iniciar sesión, pero esto tiene una limitación, no se puede proceder al pago si no se ha registrado anteriormente e iniciado sesión.

Cuando un usuario decide registrarse, desbloquea la función de que, una vez seleccionado los productos de su agrado, pueda a proceder a rellenar un formulario con los datos de su domicilio, para así realizar un pedido, el cual llevará a una pasarela de pago por tarjeta bancaria implementada por *Stripe*.

El usuario podrá acceder a una sección “Mis pedidos”, donde podrá ver los pedidos que ha realizado y se mostrará información relevante como el ID del pedido, coste, fecha, hora y estado, también podrá ver los detalles del pedido o eliminar/cancelar el pedido en caso de que el pedido no haya sido confirmado por un administrador.

Los productos de los carritos de los usuarios no registrados se guardarán en una sesión temporal, para que permita la navegación por la tienda online sin perder esos datos.

En caso de los usuarios registrados, se guardarán en una base de datos, dónde cada modificación del carrito también ocurrirá en la base de datos, ya sea añadir, quitar o eliminar el carrito, para mantener la consistencia de los datos y evitar errores. Cada vez que el usuario inicie sesión podrá ver su carrito intacto.

Al iniciar sesión, el usuario puede optar por elegir la opción de ser recordado por una *cookie* durante 7 días.

Todos los usuarios pueden modificar sus datos personales en la sección “Editar perfil”, donde podrán modificar solamente su nombre, apellidos y correo electrónico.

Un usuario puede ser administrador, dicho rol desbloquea funciones especiales para la gestión total de la tienda, como crear, editar y eliminar productos, categorías, pedidos y usuarios. Pueden confirmar pedidos o cancelarlos causando un “*trigger*” automático de la página web enviando un correo electrónico al usuario informándole de cada estado.

### 1.2. Restricciones generales

Los usuarios de la tienda no tienen opción de cambiar su contraseña sin contactar con un administrador.

Los administradores no pueden cambiar su propio rol iniciado sesión.

### 1.3. Aspectos a cubrir

Se han incluido funciones necesarias como la validación de campos en formularios mediante expresiones regulares, autenticación de usuarios, carrito de compra, gestión de productos, usuarios, pedidos y categorías a un nivel funcional real.

No se ha incluido la opción de editar la contraseña de ningún usuario, se deberá hacer manualmente desde la base de datos.

No se ha adaptado la tienda online para dispositivos móviles.

### 1.4. Estudio de las prestaciones de la herramienta

Hay que mencionar la existencia de plataformas online como *WIX*, *Shopify* o *Wordpress* que podrían hacer una página web con plantillas predeterminadas para satisfacer las necesidades de un cliente sin conocimientos de programación, aunque la diferencia principal con la herramienta que presento es que totalmente personalizable y editable al poder acceder al código fuente, sin necesidad de contratar servicios ni pagar por *plugins* para añadir funcionalidades, si no que con conocimiento informático se puede añadir todo gratuitamente.

## 2. Tecnología empleada

Se han implementado una serie de tecnologías que ha permitido el desarrollo y funcionamiento de la tienda online, dichas tecnologías son las siguientes:

**HTML5:** Es la última versión del lenguaje de marcado de hipertexto y se utiliza para estructurar contenido en la web. Sirve principalmente para definir la estructura de una página web, incorporar contenido multimedia sin necesidad de *plugins* externos, mejorar la semántica con nuevas etiquetas para facilitar la accesibilidad y el SEO.

**CSS:** Es un lenguaje que se usa para dar estilos y diseño a páginas web escritas en HTML. Sirve principalmente para definir apariencias visuales de los elementos HTML, controlar la distribución y diseño de dichos elementos, adaptar el diseño a otros dispositivos mediante un diseño responsivo.

**PHP:** Es un lenguaje de programación de *backend* que se utiliza precisamente para desarrollar páginas web dinámicas. Sirve principalmente para generar contenido dinámico extraída de una base de datos o información personalizada, procesar y controlar formularios, conectarse a base de datos y crear sistemas completos con funcionalidades.

**MySQL:** Es un sistema de gestión de bases de datos relacionales que permite almacenar, organizar y recuperar datos. Sirve principalmente para guardar datos estructurados en tablas, realizar consultas en lenguaje SQL permitiendo un CRUD, relacionar datos entre tablas.

**Composer:** Es una herramienta de gestión de dependencias para PHP. Sirve principalmente para automatizar la instalación de bibliotecas o dependencias, cargar automáticamente clases de PHP con un *autoloader*, actualizar herramientas que necesita nuestra aplicación.

***PHPMailer***: Es una librería de PHP que permite el envío de correos electrónicos. Sirve principalmente para enviar correos desde formularios de registro o contacto.

***PHPDotenv***: Es una librería de PHP que permite cargar variables de entorno desde un archivo “.env”. Sirve principalmente para guardar configuraciones sensibles como claves API y contraseñas.

***Stripe-PHP***: Es una librería oficial de Stripe para PHP, que permite integrar pagos online en páginas web. Sirve principalmente para cobrar con tarjeta de crédito de forma segura.

***Git***: Es un sistema de control de versiones. Sirve principalmente para gestionar el historial de cambios en un proyecto en desarrollo, guardar versiones del código, revertir errores de forma sencilla volviendo a otras versiones.

***Docker***: Es una plataforma que permite empaquetar y ejecutar aplicaciones en contenedores. Sirve principalmente para asegurar que funcione en cualquier entorno la aplicación, ya sea local, en un servidor o nube.

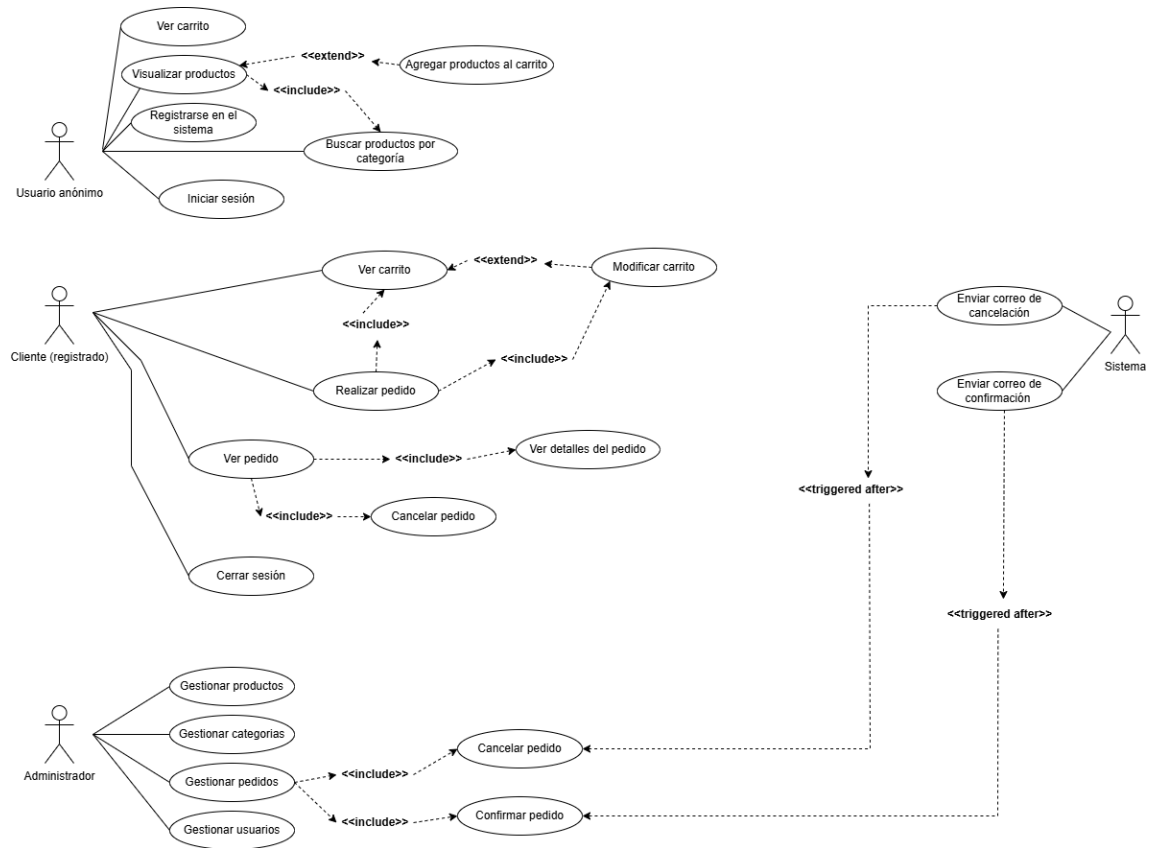
### 3. Requerimientos de hardware y software

Si la aplicación se ejecuta en local, es necesario configurar un servidor, para ello es necesario instalar ***XAMPP***, que es un paquete de software gratuito que incluye todo lo necesario para crear un servidor web local en el ordenador. Es fácil de instalar y configurar, permite probar aplicaciones web sin necesidad de tener una conexión a internet.

Se debe contar con un navegador web independientemente de ejecutarlo en local o remoto, con la única deferencia de que, si se ejecuta en remoto, se precisa de una conexión a internet.

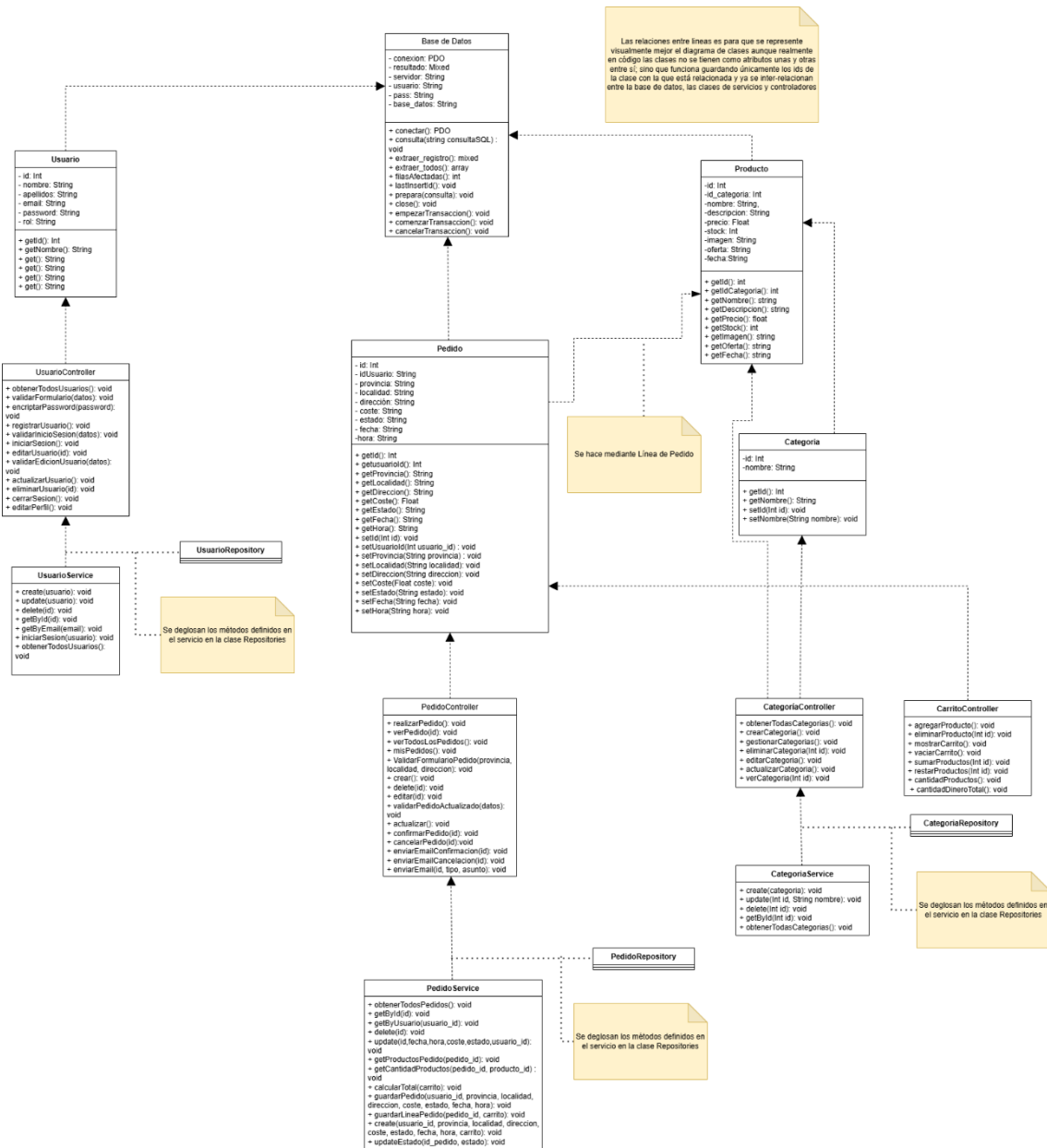
## 4. Análisis y diseño

### 4.1. Diagrama de casos de uso





## 4.2. Diagrama de clases



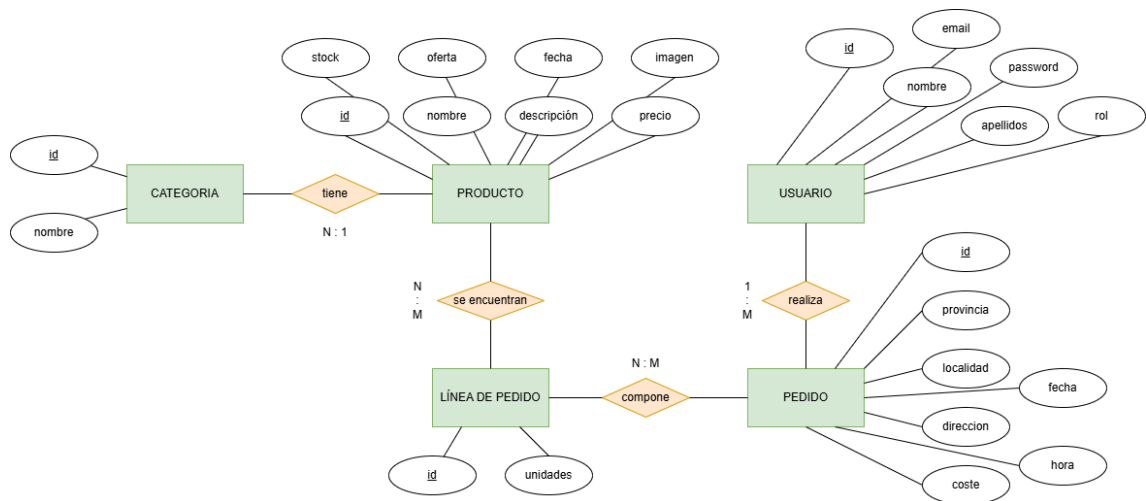
## 4.3. Descripción de la base de datos

La base de datos está diseñada para gestionar una tienda online, pudiendo almacenar y relacionar datos sobre usuarios, productos, pedidos, categorías y carritos de compra. A continuación, se describen las principales tablas:

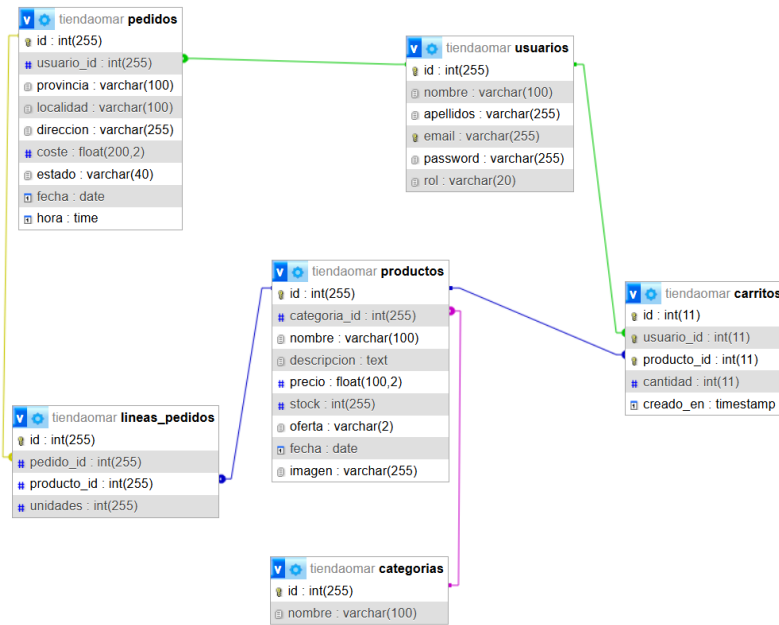
- Usuarios:** Esta tabla almacena información sobre los usuarios que interactúan con la tienda, ya sean, clientes o administradores. Los campos principales almacenados son id, nombre, apellidos, email, contraseña y rol.
- Productos:** Esta tabla contiene los artículos disponibles en la tienda, cada producto está vinculado a una categoría. Los campos principales almacenados son id, id de la categoría, nombre, descripción, stock, imagen, precio, oferta, fecha de creación o modificación.

- **Categorías:** Esta tabla agrupa los productos por tipos. Los campos principales almacenados son id y nombre.
- **Pedidos:** Esta tabla registra los pedidos realizados por los usuarios. Los campos principales almacenados son id, id del usuario, provincia, localidad, dirección, coste, estado, fecha y hora.
- **Líneas de pedido:** Esta tabla representa los productos individuales dentro de un pedido. Los campos principales almacenados son id, id del pedido, id del producto y unidades.
- **Carritos:** Esta tabla almacena los productos añadidos al carrito por parte de los usuarios antes de confirmar pedido. Los campos principales almacenados son id, id del usuario, id del producto, cantidad y una marca temporal del registro.

#### 4.4. Modelo entidad-relación



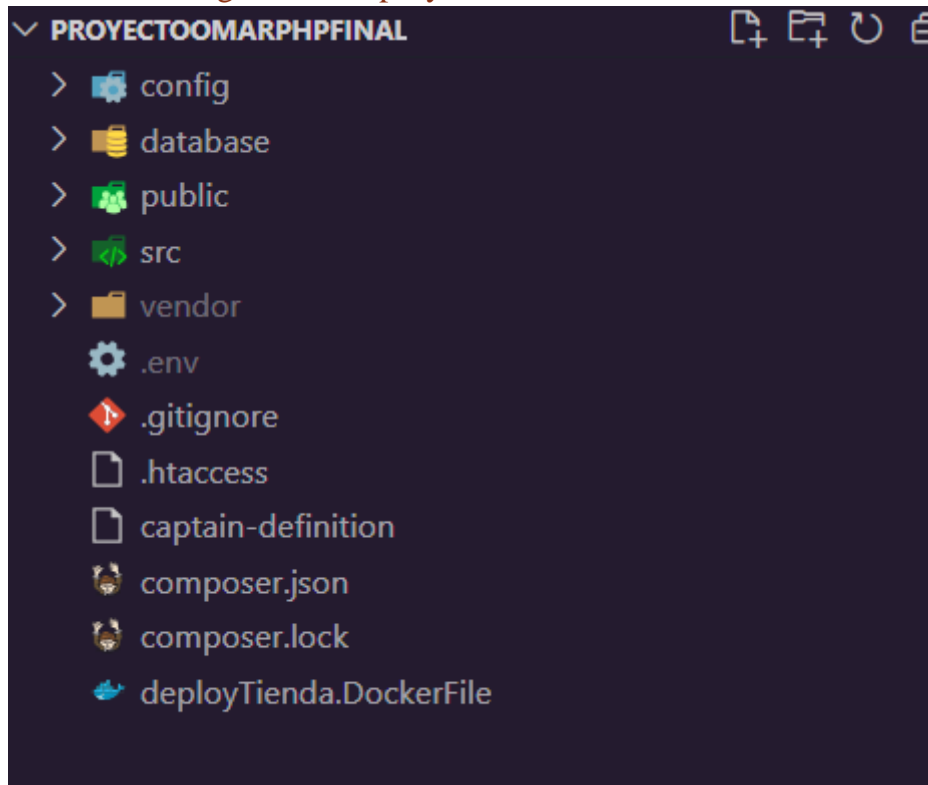
## 4.5. Diseño lógico



- **Categoría** (id, nombre)
- **Producto** (id, nombre, descripción, precio, stock, oferta, fecha, imagen, categoría\_id)
  - *Categoría\_id*: Clave foránea de Categoría(id).
- **Pedido** (id, provincia, localidad, dirección, coste, estado, fecha, hora, usuario\_id)
  - *Usuario\_id*: Clave foránea de Usuario(id).
- **Usuario** (id, nombre, apellidos, email, password, rol)
- **Línea\_pedido** (id, unidades, producto\_id, pedido\_id)
  - *Producto\_id*: Clave foránea de Producto(id).
  - *Pedido\_id*: Clave foránea de Pedido(id).
- **Carritos** (id, usuario\_id, producto\_id, cantidad, creado\_en)
  - *Usuario\_id*: Clave foránea de Usuario(id).
  - *Producto\_id*: Clave foránea de Producto(id).

## 5. Implementación

### 5.1. Estructura general del proyecto



A: Estructura general del proyecto

El proyecto está organizado siguiendo una arquitectura **MVC (Modelo-Vista-Controlador)** y dividido en las siguientes carpetas principales:

- **public/**: Punto de entrada público. Contiene index.php y archivos estáticos como CSS, imágenes y un archivo .htaccess.
- **config/**: Configuración global de la aplicación (config.php).
- **src/Controllers/**: Controladores que gestionan la lógica de negocio.
- **src/Models/**: Modelos que representan las entidades de la base de datos.
- **src/Lib/**: Librerías internas como BaseDatos, Router, Pages.
- **views/**: Plantillas de interfaz de usuario (HTML embebido en PHP).
- **.env**: Fichero de configuración para variables sensibles (como claves y credenciales).
- **vendor/**: Librerías cargadas por Composer (incluyendo Dotenv para cargar .env).

## 5.2. Rutas y Navegación

El archivo `index.php` (dentro de `public/`) es el punto de entrada y carga el archivo `init.php`, que a su vez:

- Crea una sesión (`session_start()`).
- Carga las rutas desde `Routes::index()` mediante una clase `Router` personalizada. Gracias a esta clase, se puede permitir obtener una URL amigable y limpia sin caracteres especiales como: “`verProductos/3`”.
- Comprueba si el usuario ha expirado por cookies para cerrar sesión automáticamente.
- Carga automáticamente las clases con Composer (`autoload.php`).

## 5.3. Hojas de Estilo

La apariencia visual de la aplicación se construye con:

- Un archivo CSS de formateo general en: `/public/css/styles.css`.
- Cada vista contiene su propio CSS, para facilitar el retoque de su respectiva vista localizando de forma más rápida y sencilla cada elemento.

## 5.4. Plantillas y Vistas

Las vistas están separadas por módulos y se renderizan con la clase `Pages.php`, que permite cargar contenido dinámico en páginas estructuradas. Se incluye:

- La importación de las secciones del encabezado y pie de página (*`header.php`*, *`footer.php`*) de forma automática en cada vista para evitar la duplicación de código.
- Uso de datos enviados por el controlador a la vista mediante variables con el método *`render`*.

## 5.5. Formularios

Los formularios permiten la interacción del usuario con el sistema en acciones como:

- Realizar el registro de usuarios e inicio de sesión.
- Realizar un pedido, rellenando los datos de domicilio.
- Proceder al pago de un pedido accediendo a la pasarela de pago.
- Crear, eliminar y editar productos, usuarios, pedidos, categorías (para administradores).

Todos los formularios usan el método POST y son validados y procesados en el backend con seguridad.

### 5.6. Envío y Procesamiento de Datos

Cada formulario se enlaza con un **Controlador** responsable de procesar los datos recibidos desde \$\_POST:

- Ejemplo: CarritoController.php recibe la acción de añadir, eliminar un producto o vaciar el carrito, procede a guardarlo en una sesión y sincronizarla con la respectiva tabla en la base de datos.
- Siempre se validan los datos, comprobando si existen y cumplen con el formato solicitado antes de ejecutarse.
- Se manejan redirecciones según éxito o error con (*header("Location: ...")* o *render*).

### 5.7. Seguridad

- Uso de sesiones para mantener la autenticación del usuario.
- Autenticación básica protegida por contraseña (*password\_hash()* y *password\_verify()*).
- Separación de variables sensibles mediante .env (por ejemplo: claves de base de datos o Stripe).
- Protección ante inyección SQL mediante **consultas preparadas con PDO**.

### 5.8. Conexión a Base de Datos

La clase BaseDatos.php maneja la conexión a MySQL usando PDO con estas características:

- Conexión segura mediante .env:

```
DB_HOST = localhost
DB_USER = root
DB_PASS =
DB_DATABASE = tiendaomar
```

- Activación de errores con *PDO::ERRMODE\_EXCEPTION*.
- Codificación utf8 para soporte de caracteres internacionales.
- Soporte para transacciones (*beginTransaction*, *commit*, *rollBack*).

## 5.9. Consultas y Modelos

Cada entidad de la base de datos está representada por una clase modelo (en `src/Models/`). Por ejemplo:

- `Producto.php`: Métodos como `getAll()`, `getById()`, `insertarProducto()`.
- `Usuario.php`: Métodos para registrar, iniciar sesión, validar roles.
- `Pedido.php`: Gestión de pedidos y líneas de pedido.
- `Categoria.php`: Consultas simples de listado.

Las consultas están optimizadas y protegidas contra SQL Injection gracias a los métodos `prepare()` y `bindParam()`.

## 5.10. Fichero de Configuración

Se utiliza un archivo “.env” para las variables de entorno, que se carga con Dotenv (mediante Composer):

Además, el archivo `config/config.php` define constantes como “BASE\_URL” para gestionar rutas relativas en toda la aplicación.

## 5.11. Librerías y Herramientas Usadas

- **PHP** como lenguaje principal.
- **MySQL** como sistema gestor de base de datos.
- **Composer** para gestión de dependencias (vluca/phpdotenv, etc.).
- **PDO** para conexión y gestión segura de datos.
- **Stripe API** para pagos (presente en `StripeController.php`).

## 5.12. Validaciones y Experiencia de Usuario

- Validación de datos tanto del lado cliente (HTML5) como del servidor.
- Redirecciones y mensajes de error/éxito tras formularios.
- Cookies para recordar usuarios.
- Control de sesión y expiración automática si el usuario marcó la opción “recordarme”.

### 5.13. Buenas Prácticas Implementadas

- Separación clara entre lógica, vista y datos.
- Reutilización de componentes mediante plantillas.
- Manejo de errores y fallos de conexión.
- Control de sesiones y redirecciones protegidas.
- Uso de archivos de configuración para centralizar parámetros críticos.
- Comentarios en el código para facilitar el mantenimiento.

## 6. Evaluación y prueba

En el desarrollo de la aplicación web se han hecho numerosas pruebas tanto en campos como en funcionalidades para comprobar su correcto funcionamiento y evitar fallas de lógica, se ha intentado cubrir cualquier posible error, no solo como programador si no adoptando el pensamiento de un cliente al navegar por la tienda.

Todos los formularios poseen métodos de validación para los datos, no solo en el cliente (con HTML) sino también en el procesamiento de datos en el servidor, se ha incluido validaciones en la base de datos para evitar posibles inyecciones de código maliciosos.

Cada fallo en la validación manda un mensaje de error que se mostrará en la interfaz gráfica para el usuario, con el fin de permitir su corrección.

Las pruebas que se han hecho, entre muchas otras, son las siguientes:

- Un administrador no puede eliminarse a sí mismo ni cambiar su rol por uno inferior, evitando fallas en la lógica.
- Un usuario no puede comprar productos sin estar iniciado sesión.
- Un usuario no puede añadir la cantidad superior de stock de productos de un tipo al carrito.
- Un usuario no puede eliminar un pedido realizado si el estado del pedido es “confirmado” o “cancelado”.
- Los productos de una categoría eliminada permanecen en la base de datos con la posibilidad de poder volver a añadirlos a otra categoría, es decir, nunca se pierden.
- Un administrador no puede eliminar pedidos que hayan sido confirmados, pero sí cancelarlos.
- Un usuario no puede acceder de ninguna forma, ni por URL, al panel de administración.



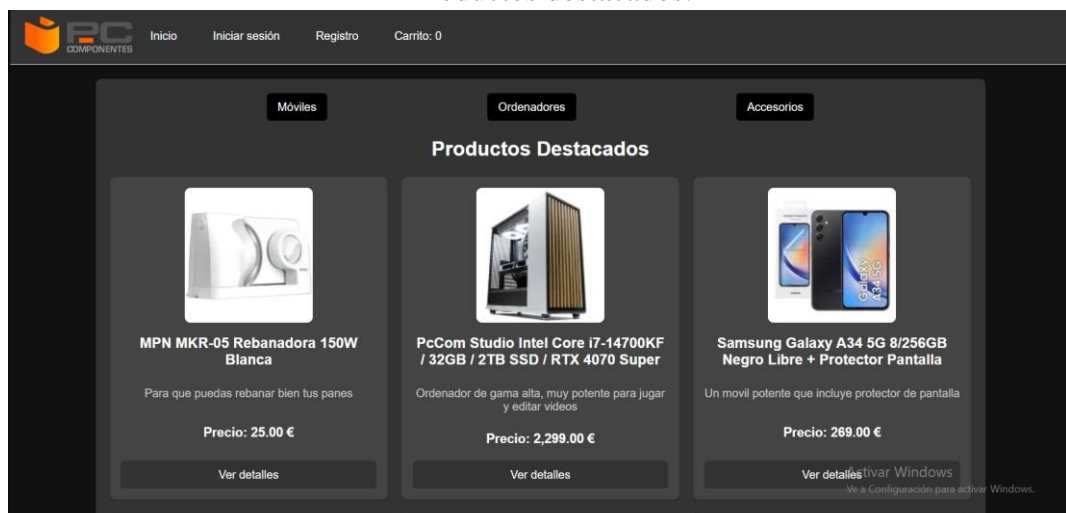
También se han hecho métodos de validaciones de formularios como se ha comentado anteriormente, a continuación, se presentan algunos campos validados:

- Provincia, localidad y dirección a la hora de realizar un pedido.
- Coste, estado, fecha y hora a la hora de actualizar un pedido
- Nombre, apellidos, email, contraseña, rol al registrar un usuario.
- Email y contraseña al iniciar sesión.
- Nombre, apellidos, email y rol al editar un usuario.
- Nombre, descripción, precio, stock, categoría, imagen al crear o editar un producto.
- Nombre al crear o editar una categoría.

## 7. Manual de estilos

### 7.1. Sketches

- Productos destacados:



- Ver detalles del producto:



- Iniciar sesión:

**Iniciar sesión**

Correo electrónico

Contraseña

Recordar usuario ☐

Iniciar sesión

Regístrate aquí

**Enlaces Rápidos**  
Inicio  
Productos  
Ofertas

**Contacto**  
Email: soporte@PcComponentesOmar.com  
Teléfono: +34 123 456 789

**Síguenos**  
Facebook  
Twitter

Activar Windows  
Ve a Configuración para activar Windows.

- Registro:

**Registro de usuario**

Nombre:

Apellidos:

Correo electrónico:

Contraseña:

Registrar

Volver a inicio de sesión

Activar Windows  
Ve a Configuración para activar Windows.

- Carrito sin productos:

**Carrito**

Imagen	Producto	Precio	Stock	Cantidad	Total
No hay productos en el carrito					

Total: 0€  
Número de artículos: 0

Vaciar Carrito Realizar Pedido

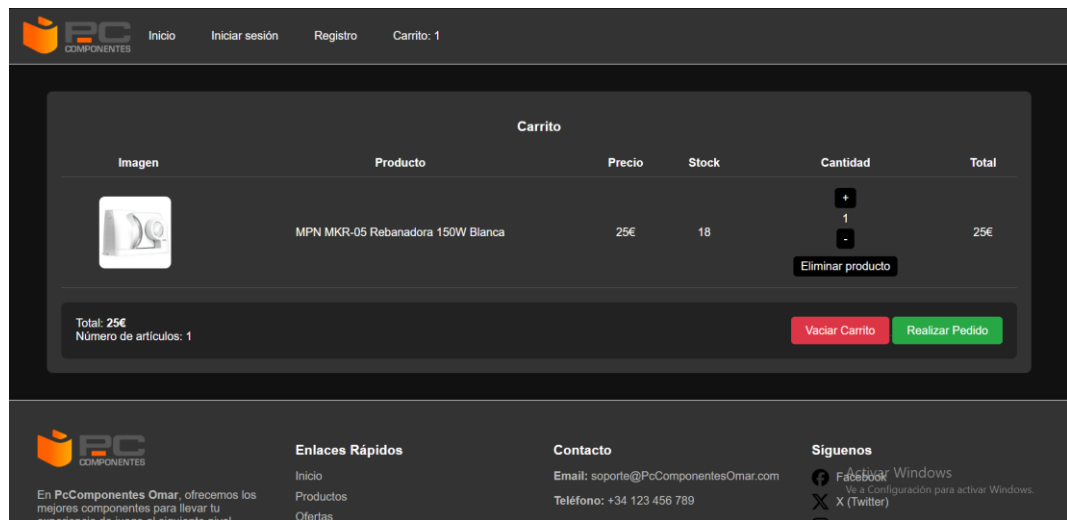
**Enlaces Rápidos**  
Inicio  
Productos  
Ofertas

**Contacto**  
Email: soporte@PcComponentesOmar.com  
Teléfono: +34 123 456 789

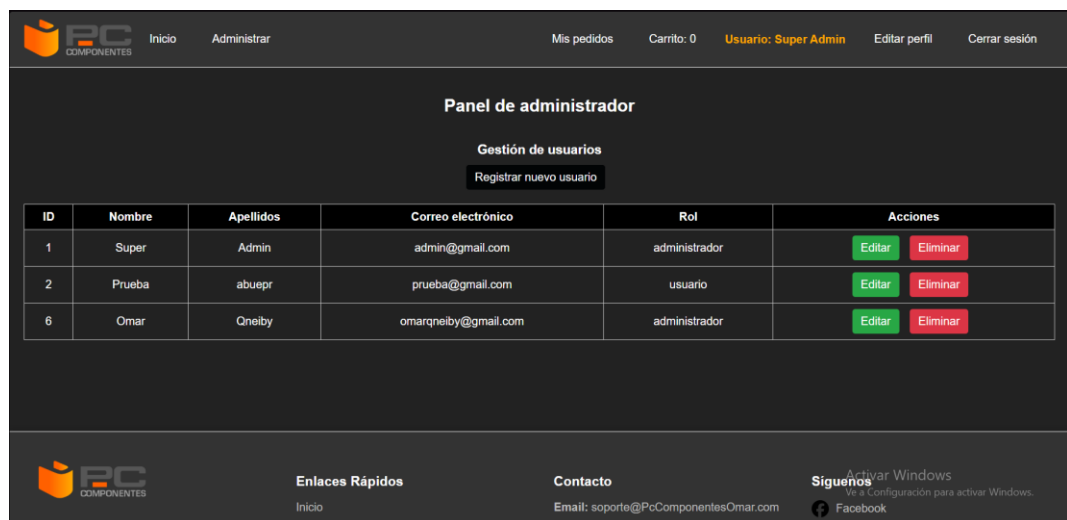
**Síguenos**  
Facebook  
Twitter

Activar Windows  
Ve a Configuración para activar Windows.

- Carrito con productos:



- Panel de administrador (gestión de usuarios):



- Formulario registro usuarios (panel administrador):

The screenshot shows the 'Registro de usuario' form in the administrator panel. The form is centered on a dark background and contains the following fields: 'Nombre:', 'Apellidos:', 'Correo electrónico:', 'Contraseña:', and 'Rol:'. The 'Rol:' field is a dropdown menu with 'Usuario' selected. Below the fields is a 'Registrar' button. At the bottom of the form is a 'Volver' button. The top navigation bar includes links for 'Inicio', 'Administrar', 'Mis pedidos', 'Carrito: 0', 'Usuario: Super Admin', 'Editar perfil', and 'Cerrar sesión'. The bottom of the page features a footer with 'PC COMPONENTES', 'Enlaces Rápidos', 'Contacto', 'Siguenos', and a copyright notice for 2025.

- Gestionar categorías (panel administrador):

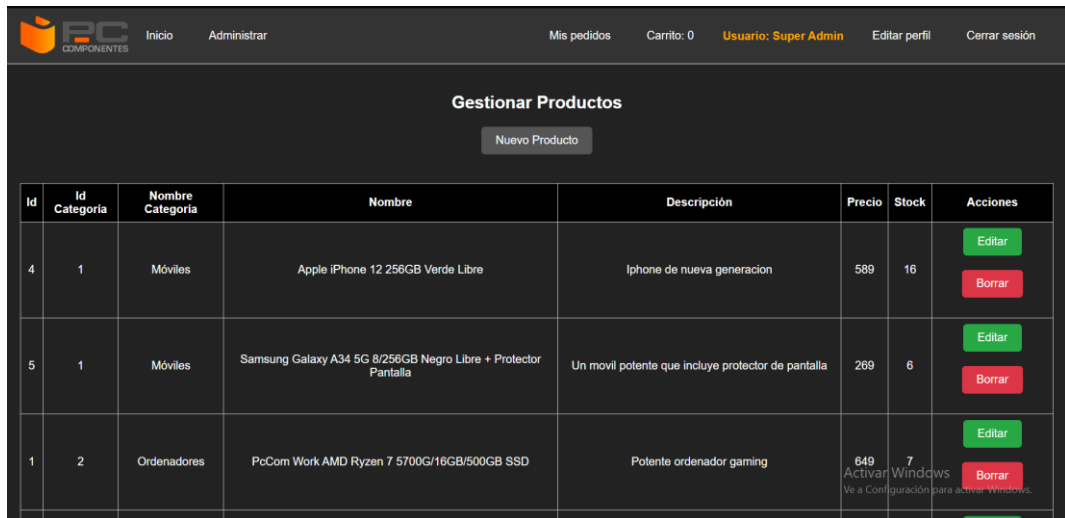
The screenshot shows the 'Gestionar categorías' page in the administrator panel. It features a table with three columns: 'ID', 'Nombre', and 'Acciones'. The table contains three rows of data. Above the table is a 'Crear nueva categoría' button. The bottom of the page features a footer with 'PC COMPONENTES', 'Enlaces Rápidos', 'Contacto', 'Siguenos', and a copyright notice for 2025.

ID	Nombre	Acciones
1	Móviles	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Ordenadores	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Accesorios	<a href="#">Editar</a> <a href="#">Eliminar</a>

- Formulario para crear categorías (panel administrador):

The screenshot shows the 'Crear categoría' form in the administrator panel. The form is centered on a dark background and contains a single text input field for 'Nombre'. Below the input field are 'Crear' and 'Volver' buttons. The top navigation bar includes links for 'Inicio', 'Administrar', 'Mis pedidos', 'Carrito: 0', 'Usuario: Super Admin', 'Editar perfil', and 'Cerrar sesión'. The bottom of the page features a footer with 'PC COMPONENTES', 'Enlaces Rápidos', 'Contacto', 'Siguenos', and a copyright notice for 2025.

- Gestionar productos (panel administrador):

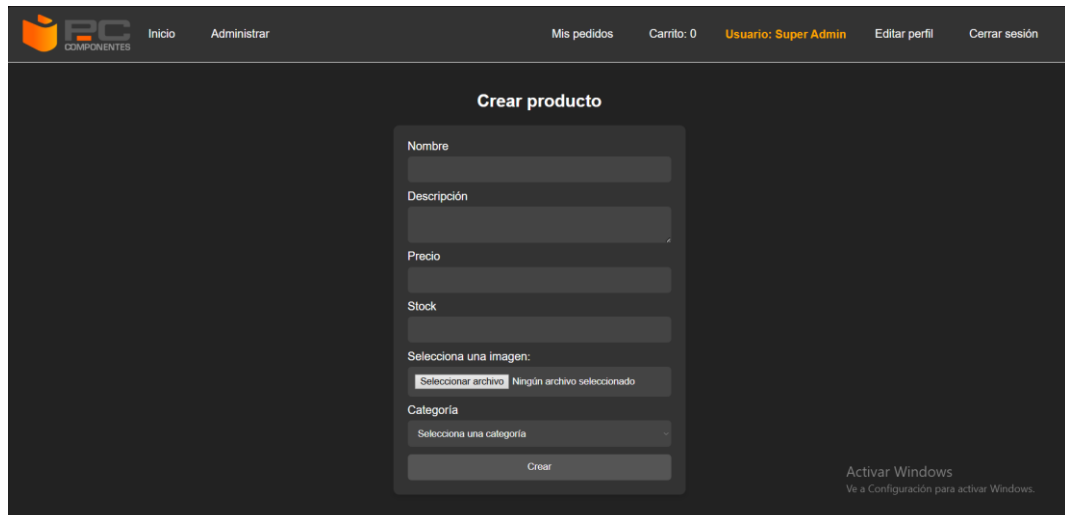


**Gestionar Productos**

[Nuevo Producto](#)

Id	Id Categoría	Nombre Categoría	Nombre	Descripción	Precio	Stock	Acciones
4	1	Móviles	Apple iPhone 12 256GB Verde Libre	Iphone de nueva generacion	589	16	<a href="#">Editar</a> <a href="#">Borrar</a>
5	1	Móviles	Samsung Galaxy A34 5G 8/256GB Negro Libre + Protector Pantalla	Un movil potente que incluye protector de pantalla	269	6	<a href="#">Editar</a> <a href="#">Borrar</a>
1	2	Ordenadores	PcCom Work AMD Ryzen 7 5700G/16GB/500GB SSD	Potente ordenador gaming	649	7	<a href="#">Editar</a> <a href="#">Borrar</a>

- Formulario para crear producto (panel administrador):



**Crear producto**

Nombre

Descripción

Precio

Stock

Selecciona una imagen:

[Seleccionar archivo](#) Ningún archivo seleccionado

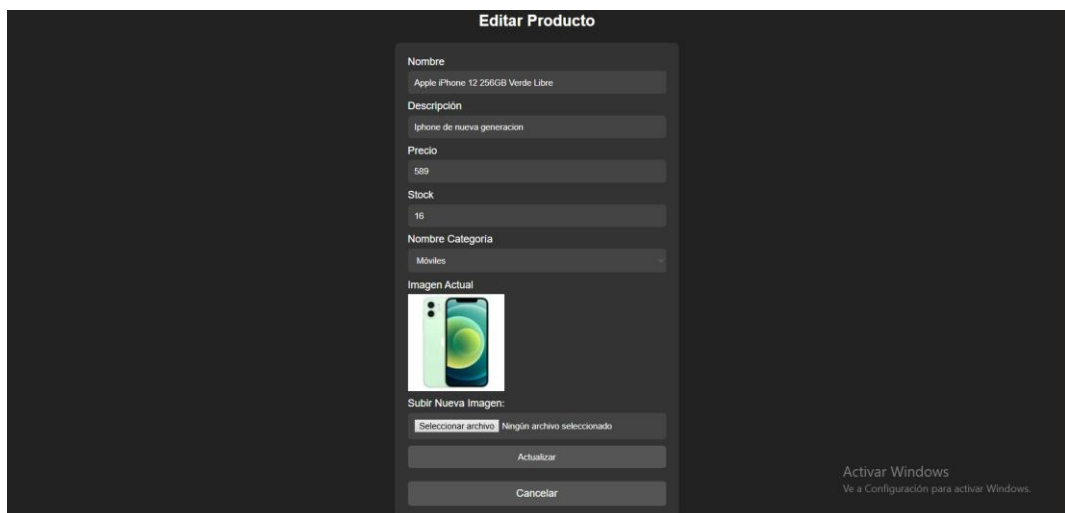
Categoría

Selecciona una categoría

[Crear](#)

Activar Windows  
Ve a Configuración para activar Windows.

- Formulario para editar productos (panel administrador):



**Editar Producto**

Nombre

Apple iPhone 12 256GB Verde Libre

Descripción

Iphone de nueva generacion

Precio

589


Stock

16

Nombre Categoría

Móviles

Imagen Actual



Subir Nueva Imagen:

[Seleccionar archivo](#) Ningún archivo seleccionado

[Actualizar](#)

[Cancelar](#)

Activar Windows  
Ve a Configuración para activar Windows.

- Gestionar pedidos (panel administrador):

The screenshot shows the 'Gestionar Pedidos' panel. At the top, there's a navigation bar with 'Inicio', 'Administrar', 'Mis pedidos', 'Carrito: 0', 'Usuario: Super Admin', 'Editar perfil', and 'Cerrar sesión'. The main content area has a table with the following data:

Id	Coste	Fecha	Hora	Estado	ID Usuario	Acciones
18	1359€	2025-06-14	21:11:16	Pendiente a confirmar	1	<a href="#">Confirmar pedido</a> <a href="#">Cancelar pedido</a> <a href="#">Eliminar</a> <a href="#">Ver Pedido</a>

The footer contains the 'PC COMPONENTES' logo, a description of the company, 'Enlaces Rápidos' (Inicio, Productos, Ofertas), 'Contacto' (Email: soporte@PcComponentesOmar.com, Teléfono: +34 123 456 789), and 'Síguenos' (Facebook, X (Twitter), Instagram) with a Windows activation notice.

- Ver detalles del pedido:

The screenshot shows the 'Detalles del Pedido #18' page. The navigation bar is the same as the previous screenshot. The main content area displays the order details for user ID 1, including the product 'PcCom Ready AMD Ryzen 7 5800X / 32GB / 1TB SSD / RTX 4060 Ti' with a quantity of 1, a unit price of 1359€, and a total of 1359€. A 'Volver' button is at the bottom of the details section. The footer is identical to the previous screenshot.

- Mis pedidos:

The screenshot shows the 'Mis Pedidos' page. The navigation bar is the same as the previous screenshots. The main content area displays a message: 'No has realizado ningún pedido' (You have not made any order). The footer is identical to the previous screenshots.

- Editar perfil:

**Editar Perfil**

Nombre  
Prueba

Apellidos  
abuepr

Correo electrónico  
prueba@gmail.com

[Guardar Cambios](#) [Cancelar](#)

**Enlaces Rápidos**  
Inicio  
Productos  
Ofertas

**Contacto**  
Email: soporte@PcComponentesOmar.com  
Teléfono: +34 123 456 789

**Síguenos**  
Facebook  
Twitter  
Instagram  
YouTube

- Detalles del pedido al proceder a realizar la compra del carrito:

**Detalles del pedido**  
Introduce tus datos para realizar el pedido

Provincia  
Localidad  
Dirección

[Confirmar Pedido](#)

**Enlaces Rápidos**  
Inicio  
Productos  
Ofertas  
Contacto

**Contacto**  
Email: soporte@PcComponentesOmar.com  
Teléfono: +34 123 456 789  
Dirección: Calle Falsa 123, Ciudad Gaming, País

**Síguenos**  
Facebook  
Twitter  
Instagram  
YouTube

- Pasarela de pago por Stripe:

← Entorno de prueba de PcCompon... Entorno de prueba

Pedido en Tienda Omar  
2299,00 €

[Pagar con link](#)

Correo electrónico  
correo@ejemplo.com

Información de la tarjeta  
1234 1234 1234 1234  
MM / AA CVC

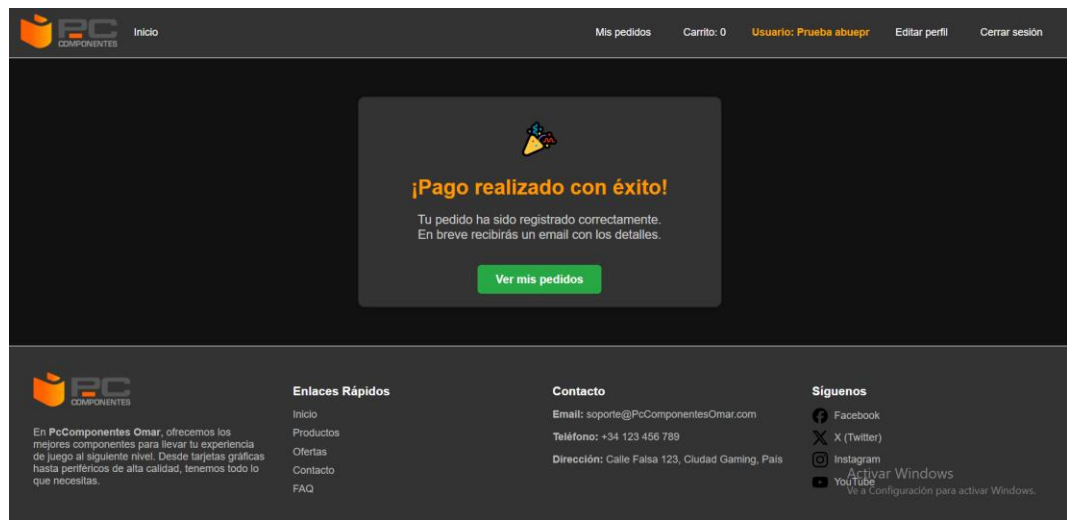
Nombre del titular de la tarjeta  
Nombre completo

País o región  
España

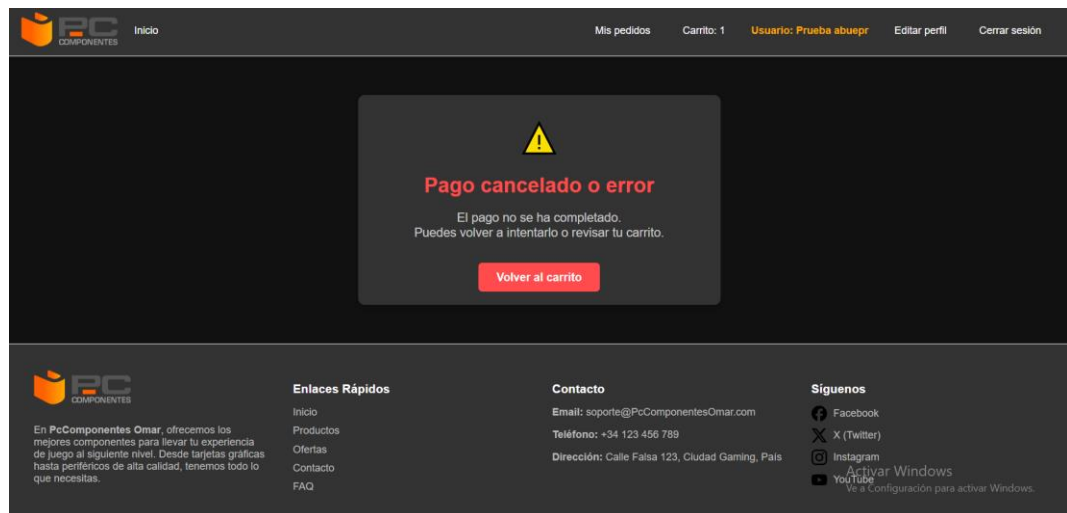
☐ Guardar mis datos de forma segura para un proceso de compra en un clic  
Paga más rápido en Entorno de prueba de PcComponentesOMAR y en todos los comercios que acepten Link.

[Pagar](#)

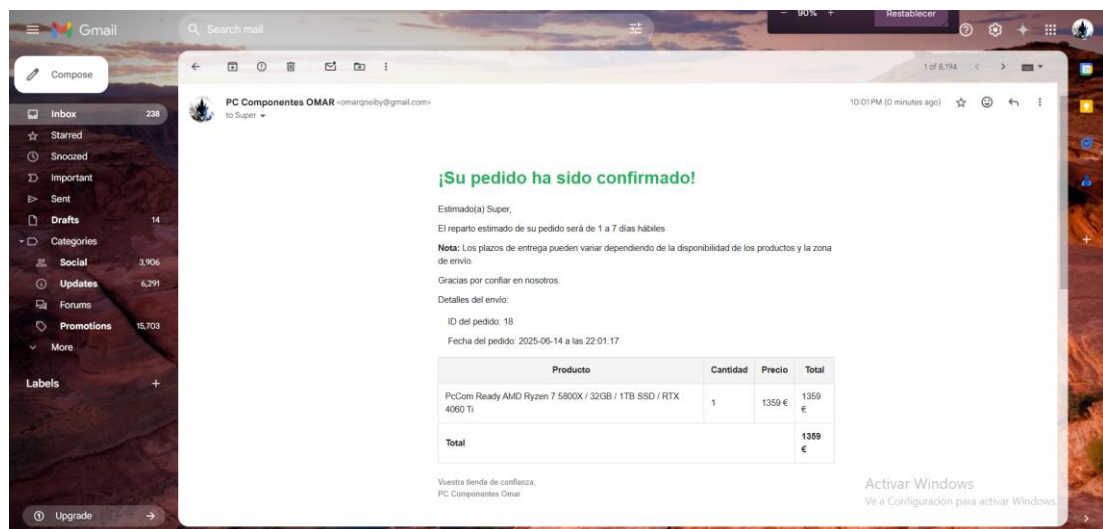
- Confirmación de pago:



- Error en el pago:

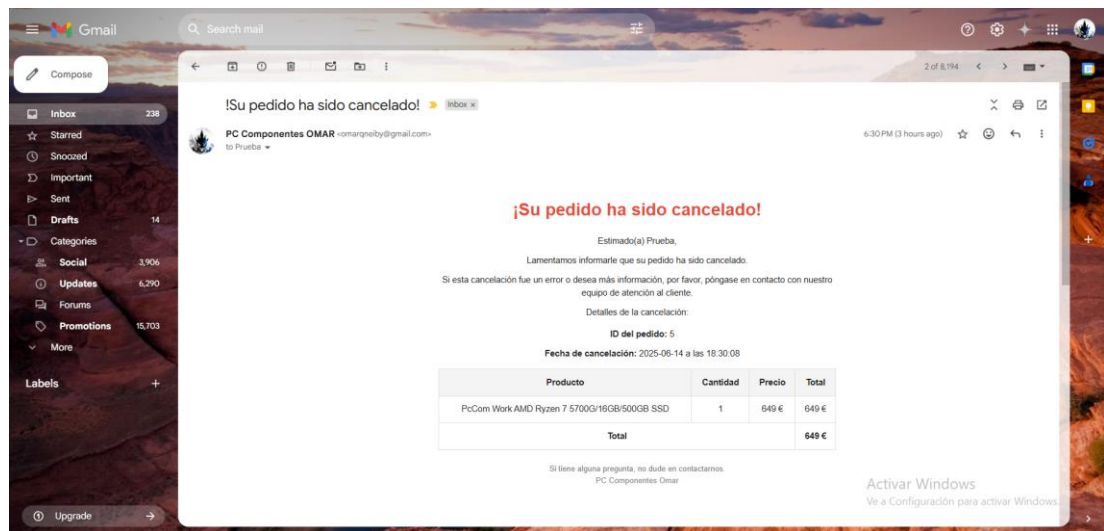


- Correo de confirmación de pedido:





- Correo de cancelación de pedido:



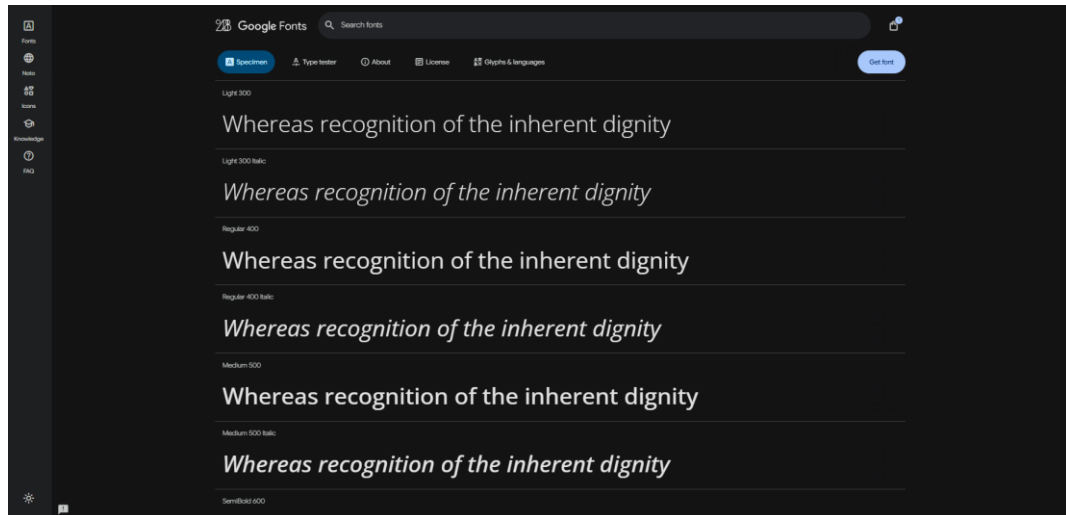
## 7.2. Criterios de accesibilidad y usabilidad

Se han tenido en cuenta ciertos criterios de accesibilidad para mejorar la experiencia del usuario como:

- Navegación clara, simple y concisa. Con una buena estructura lógica del contenido.
- Formularios bien estructurados y entendibles que muestran mensajes de éxito o error y mantienen los datos introducidos en caso de un envío fallido por no cumplir una validación.
- Mensajes descriptivos al realizar completar un proceso o funcionalidad correctamente o incorrectamente, informando al usuario si su acción ha sido exitosa o no.
- Contraste adecuado de colores para evitar ruido visual, y facilitar la lectura con un fondo oscuro.

### 7.3. Tipografía

Se ha hecho uso de la fuente Open Sans, una fuente importada de Google Fonts. Se ha utilizado un tamaño predeterminado de “18px”, y en la página aparece la fuente no solamente en estilo predeterminado, sino que también en negrita.



## 7.4. Mapa de colores

La siguiente tabla muestra los colores que se han utilizado en la página web, con su color de muestra, formato RGB, formato hexadecimal y en qué elementos se utiliza.

Color muestra	Hexadecimal	RGB	Nombre Aproximado	Uso
	#111111	rgb(17, 17, 17)	Eerie Black	Fondo principal (body)
	#222222	rgb(34, 34, 34)	Dark Gray	Fondos secundarios, formularios
	#333333	rgb(51, 51, 51)	Dark Gray	Header, menús, cards, footer
	#444444	rgb(68, 68, 68)	Gray	Fondos, cards, bordes
	#555555	rgb(85, 85, 85)	Gray	Fondos, inputs, botones
	#666666	rgb(102, 102, 102)	Gray	Fondos, inputs, botones
	#bbbbbb	rgb(187, 187, 187)	Light Gray	Textos secundarios, footer
	#cccccc	rgb(204, 204, 204)	Light Gray	Textos secundarios
	#dddddd	rgb(221, 221, 221)	Light Gray	Textos secundarios
	#ffffff	rgb(255, 255, 255)	White	Textos, bordes, fondos, hover
	#020405	rgb(2, 4, 5)	Very Dark Blue	Botones, detalles
	#28a745	rgb(40, 167, 69)	Green	Botón éxito, confirmación
	#218838	rgb(33, 136, 56)	Dark Green	Hover éxito
	#d4edda	rgb(212, 237, 218)	Light Green	Fondo éxito
	#c3e6cb	rgb(195, 230, 203)	Light Green	Borde éxito
	#155724	rgb(21, 87, 36)	Dark Green	Texto éxito
	#dc3545	rgb(220, 53, 69)	Red	Botón error, texto error
	#c83333	rgb(200, 35, 51)	Dark Red	Hover error
	#f8d7da	rgb(248, 215, 218)	Light Red	Fondo error
	#f5c6cb	rgb(245, 198, 203)	Light Red	Borde error
	#721c24	rgb(114, 28, 36)	Dark Red	Texto error
	#ff4c4c	rgb(255, 76, 76)	Red	Botón error
	#ff4d4d	rgb(255, 77, 77)	Red	Texto error
	#ff0000	rgb(255, 0, 0)	Red	Texto error
	#ff9900	rgb(255, 153, 0)	Orange	Texto destacado
	#ffa500	rgb(255, 165, 0)	Orange	Texto destacado, user-info
	#00ff00	rgb(0, 255, 0)	Green	Mensaje éxito
	#331515	rgb(51, 21, 21)	Dark Brown	Fondo error
	rgba(0,0,0,0.1)	rgb(0,0,0,0.1)	Negro translúcido	Sombra, box-shadow
	rgba(0,0,0,0.2)	rgb(0,0,0,0.2)	Negro translúcido	Sombra, box-shadow
	rgba(255,255,255,0.2)	rgb(255,255,255,0.2)	Blanco translúcido	Hover enlaces, menús

## 7.5. Dispositivos / Vistas

El proyecto se ha diseñado solamente para vistas de ordenador y portátil, no se ha contemplado el uso de *media queries* para ajustar las resoluciones a otros dispositivos.

Se ha hecho usos de maquetaciones de diseño bidimensional como “*grid*” para mostrar vistas como los productos destacados de la página y unidimensional como “*flexbox*” para mostrar los menús del *header* y del menú de categorías.

## 8. Software utilizado

Aunque anteriormente se ha mencionado que tecnologías se han utilizado para el desarrollo y funcionamiento de la aplicación, el software utilizado es el siguiente:

- **Visual Studio Code:** Es un editor de código (IDE). Es una herramienta que permite desarrollar código en una gran variedad de lenguajes de programación, se caracteriza por ser una herramienta ligera, rápida, personalizable (en cuanto a extensiones) y gratuita.
- **XAMPP:** Se ha definido anteriormente.
- **phpMyAdmin:** Es una herramienta web escrita en PHP que nos permite administrar las bases de datos MySQL o mariaDB a través de una interfaz gráfica. Es una alternativa a usar comandos SQL manualmente en una consola
- **Opera GX:** Navegador web utilizado para abrir la aplicación web en local y realizar pruebas.
- **Git:** Se ha definido anteriormente.
- **Github:** Es una plataforma online que permite almacenar, compartir y colaborar en proyectos de programación.
- **Stripe:** Es una plataforma de pagos online que permite integrar en páginas web o apps una pasarela de pago para efectuar pagos mediante tarjeta de crédito o débito, bizum, entre otros más, de manera segura, fiable y rápida.

## 9. Mejoras posibles y aportaciones

A pesar de que la aplicación web, como opinión personal, está bastante completa en cuanto a funcionalidades, siendo autocrítico, se podrían haber añadido funcionalidades o contenido para que luzca más profesional.

### 9.1. Funcionalidades o mejoras que se podrían haber añadido

- Se podría haber implementado una función de búsqueda de productos en específico mediante un buscador integrado en la página.
- Se podría haber añadido una opción para cambiar la contraseña de los usuarios en caso de que se le olvidara la contraseña, ya sea al editar el perfil o enviando un correo electrónico con un enlace para restaurar la contraseña.
- Se podría haber añadido un catálogo más extenso de productos y categorías.
- Se podría haber adaptado la aplicación web para mostrarse correctamente en dispositivos de una resolución menor a la de un portátil o sobremesa.

## 9.2. Organización de código

Para finalizar, es preciso comentar que el código siempre se puede mejorar y optimizar. En este caso, he de mencionar que se podría haber organizado el código de los estilos (CSS) de forma más global, en vez de que cada vista tuviese su propio estilo.

La ventaja de esto habría sido que todo el CSS se encontraría en un archivo en específico, pero como opinión personal, he decidido hacerlo por separado para facilitarme el diseño de cada componente de forma más rápida y sin tener que hacer *scroll* en un archivo CSS muy extenso.