

PROYECTO FINAL DE GRADO

BookStore



Maria Jose Salar García

Juan Antonio Rodríguez Toral

Índice

1. Introducción y Objetivos	3
2. Análisis del Proyecto	4
Herramientas de análisis utilizadas.....	4
Páginas agregadas sin análisis previo	10
3. Diseño del Proyecto	12
Responsive.....	12
Paleta de colores	13
Estructura de la web	14
4. Planificación de Fases del Proyecto	15
5. Desarrollo. Contenidos	18
Contenidos	18
Frontend	19
HTML y CSS	19
JavaScript	22
Backend y bases de datos (PHP y MySQL).....	23
6. Seguimiento de Fases del Proyecto	28
7. Resultados. Conclusiones. Propuestas Futuras	29
8. Bibliografía	31
9. Anexos	32

1. Introducción y Objetivos

Este proyecto, **BookStore**, surge como parte del módulo del ciclo de Desarrollo de Aplicaciones Web. Su objetivo principal es aplicar de forma práctica todos los conocimientos adquiridos durante estos años de formación, desarrollando una aplicación web funcional y completa.

La idea de crear una librería online parte de una combinación entre el interés personal por los libros de la compañera Maria Jose Salar García y el gusto por crear soluciones digitales útiles del compañero Juan Antonio Rodríguez Toral. En los últimos años, hemos visto cómo todo tipo de productos y servicios han migrado al entorno online, y el mundo editorial no ha sido una excepción. Cada vez es más habitual que la gente compre libros desde su casa, por lo que me pareció una buena oportunidad plantear una solución web moderna que responda a esa necesidad.

A nivel técnico, el proyecto nos ha permitido trabajar con herramientas y tecnologías que hemos aprendido durante el ciclo, como CSS, HTML5, PHP, JavaScript y bases de datos en MySQL. Hemos querido centrarnos no solo en que la aplicación funcione, sino también en que tenga una estructura limpia, un diseño cuidado y que sea fácil de usar para cualquier tipo de usuario.

Este trabajo también ha sido una forma de retarnos a nosotros mismos, de comprobar hasta dónde somos capaces de llegar con nuestros conocimientos actuales y de aprender cosas nuevas por nuestra cuenta cuando ha hecho falta. Además, hemos intentado organizarnos como lo haríamos en un entorno profesional: planificando las fases, documentando el proceso y haciendo pruebas para asegurar la calidad del resultado.

En resumen, BookStore es más que un simple ejercicio académico: es un proyecto con una finalidad clara, desarrollado con dedicación y con la intención de que sea una base sólida para futuros desarrollos.

Los objetivos principales de este proyecto son:

- Desarrollar una aplicación web funcional y segura.
- Implementar un diseño atractivo, accesible y responsive.
- Utilizar tecnologías actuales como HTML5, CSS3, JavaScript, PHP y Laravel.
- Integrar una base de datos MySQL para gestionar los datos.
- Desplegar la aplicación en un entorno local o servidor de pruebas.

2. Análisis del Proyecto

Para el desarrollo del proyecto BookStore se realizó un análisis funcional centrado en la gestión de productos, usuarios y pedidos. El sistema incluye tanto la parte pública (navegación, búsqueda y compra de libros) como el panel de administración para la gestión de inventario y usuarios.

Se ha realizado una primera segmentación de los usuarios en dos perfiles diferenciados:

- Usuario básico: es el cliente final, quien navega por el catálogo, consulta libros y realiza compras.
- Administrador: es el usuario con acceso privilegiado, encargado de gestionar los productos, pedidos, usuarios y de visualizar métricas relacionadas con la actividad de la web.

Nos hemos centrado en identificar las necesidades reales de los usuarios que compran libros por Internet, así como los distintos perfiles que pueden interactuar con la plataforma. Para definir los requerimientos de la aplicación se han tenido en cuenta las funcionalidades esenciales que se esperan de una tienda online, especialmente del sector editorial, como son:

- Navegación y búsqueda avanzada: los usuarios pueden explorar el catálogo, filtrar por género, precio o idioma, y acceder a detalles completos de cada libro.
- Recomendaciones automáticas: al acceder a la ficha de un libro, se muestran productos similares para fomentar nuevas compras.
- Gestión del carrito: los libros se pueden añadir, modificar o eliminar del carrito, y se puede finalizar la compra mediante un formulario intuitivo.
- Pasarela de pago integrada: integración con PayPal como medio de pago seguro.
- Registro e inicio de sesión: los usuarios pueden crear cuentas para acceder a funciones personalizadas y consultar el historial de pedidos.
- Panel de administración: los administradores pueden gestionar el catálogo, ver estadísticas, editar usuarios y controlar la actividad del sitio.

Herramientas de análisis utilizadas

- **Drawio**: Utilizamos esta herramienta durante la fase de análisis para definir y estructurar correctamente la base de datos del proyecto. En concreto, fue empleada para diseñar el modelo Entidad-Relación (E-R), lo cual nos permitió representar de forma visual y ordenada las distintas entidades del sistema, sus atributos y las relaciones entre ellas.

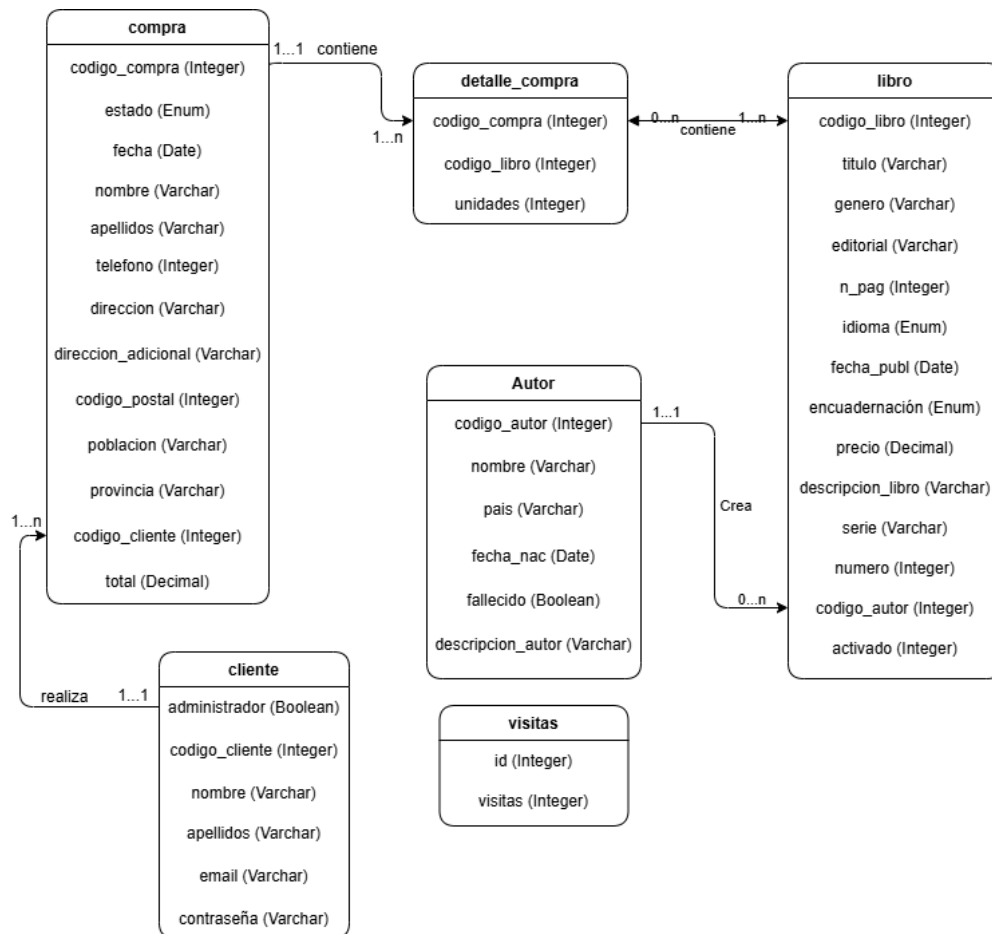


Diagrama E-R

- **Paint:** Como primer paso para visualizar cómo se estructuraría la interfaz de usuario, se optó por una herramienta sencilla pero eficaz: **Paint**. Aunque existen programas más avanzados para diseño UI, la elección de Paint se basó en su rapidez, accesibilidad y facilidad para plasmar de forma inmediata las ideas principales sin necesidad de curva de aprendizaje.

Gracias a Paint, se pudieron crear bocetos claros de las principales vistas de la aplicación web: la **página de inicio**, la **ficha de libro**, la **ficha del autor**, el **carrito de compra** y la **página de confirmación del pedido**. Estos esquemas permitieron organizar de forma visual la estructura de los elementos, los bloques de contenido y la navegación entre secciones antes de comenzar con la maquetación real en HTML y CSS.

A continuación, se muestra la imagen utilizada durante esta fase inicial del análisis y diseño:



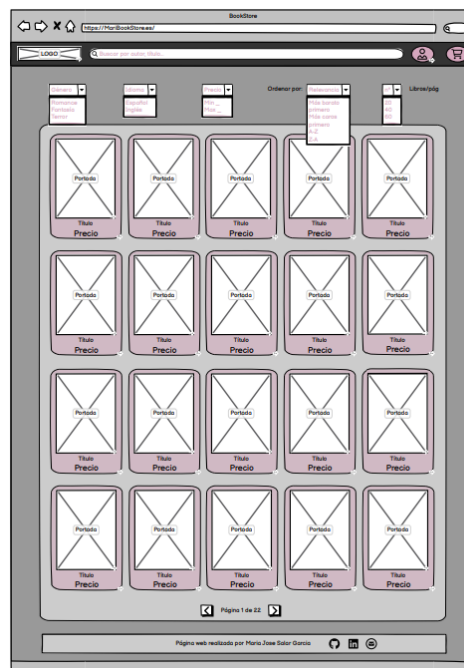
- **Wireframe:** Como la web tenía que evolucionar y dar paso a más páginas con estructuras más complejas, optamos por utilizar una opción más moderna aprendida durante las clases del ciclo. Esta técnica nos permitió planificar visualmente la estructura de las pantallas antes de pasar al desarrollo.

En concreto, utilizamos la herramienta **Balsamiq**, una aplicación especializada en la creación de prototipos de baja fidelidad. Gracias a ella, pudimos definir de forma clara y ordenada la disposición de los elementos, los flujos de navegación,

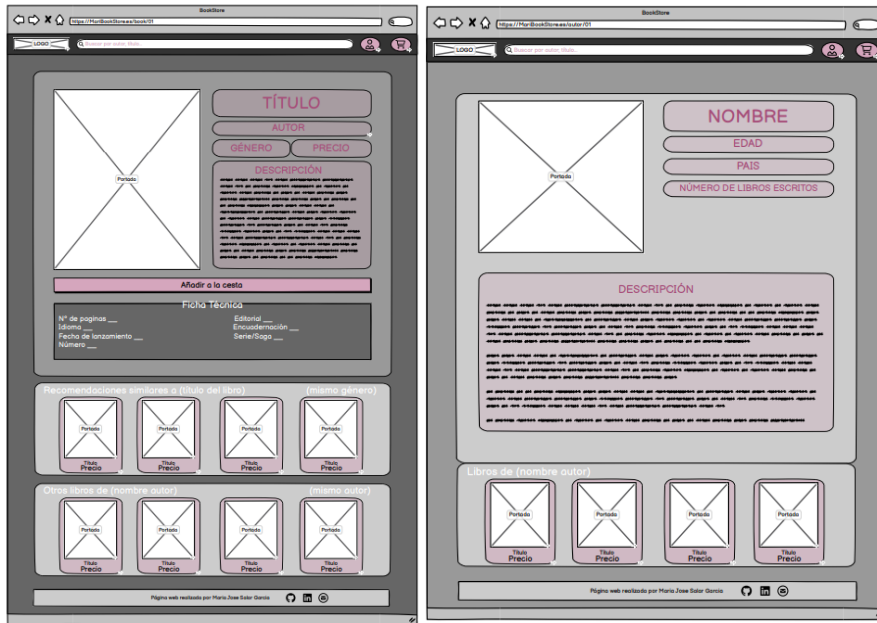
y las funcionalidades clave de cada sección de la plataforma, como el catálogo, la ficha de producto o el proceso de compra.

Balsamiq nos ayudó a tomar decisiones de diseño centradas en el usuario, priorizando la claridad visual y la usabilidad desde el principio. Estos wireframes sirvieron como guía directa durante la maquetación HTML y CSS, reduciendo errores y facilitando la colaboración con los compañeros.

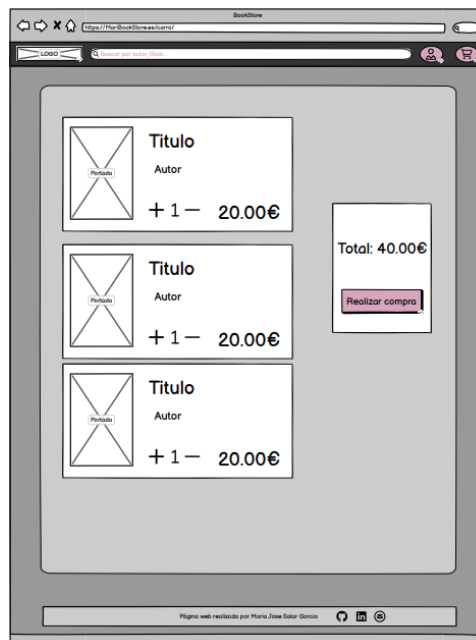
A continuación, mostramos las webs que se detallaron gracias a esta herramienta:



Inicio



Libro a detalle / Autor a detalle



Carrito

Datos de compra

Nombre	Apellidos
<input type="text"/>	<input type="text"/>
Correo electrónico	Teléfono
<input type="text"/>	<input type="text"/>
Dirección	Datos adicionales
<input type="text"/>	<input type="text"/>
Código postal	Población
<input type="text"/>	<input type="text"/>
Provincia	
<input type="text"/>	

[Realizar compra](#)

Compra realizada con éxito

ID de compra: 124123

[Volver a inicio](#)

Página web realizada por Mario José Salar García

Formulario de compra / Compra realizada

Crear cuenta

Nombre

Apellidos

Email

Contraseña

Confirma contraseña

[Crear cuenta](#)

[¿Ya tiene una cuenta? Iniciar sesión](#)

Iniciar sesión

Email

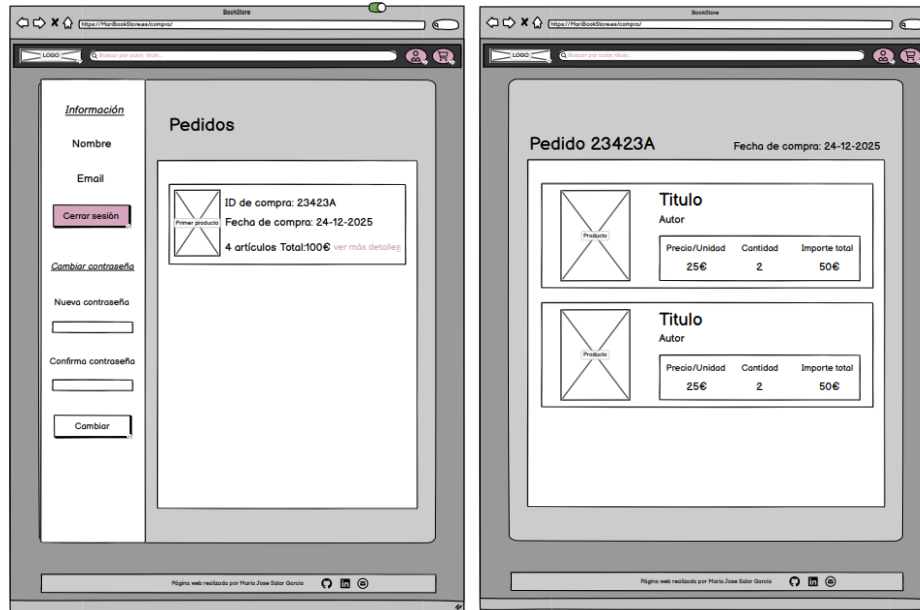
Contraseña

[Iniciar sesión](#)

[¿No tiene una cuenta? Regístrate](#)

Página web realizada por Mario José Salar García

Crear cuenta / Iniciar sesión



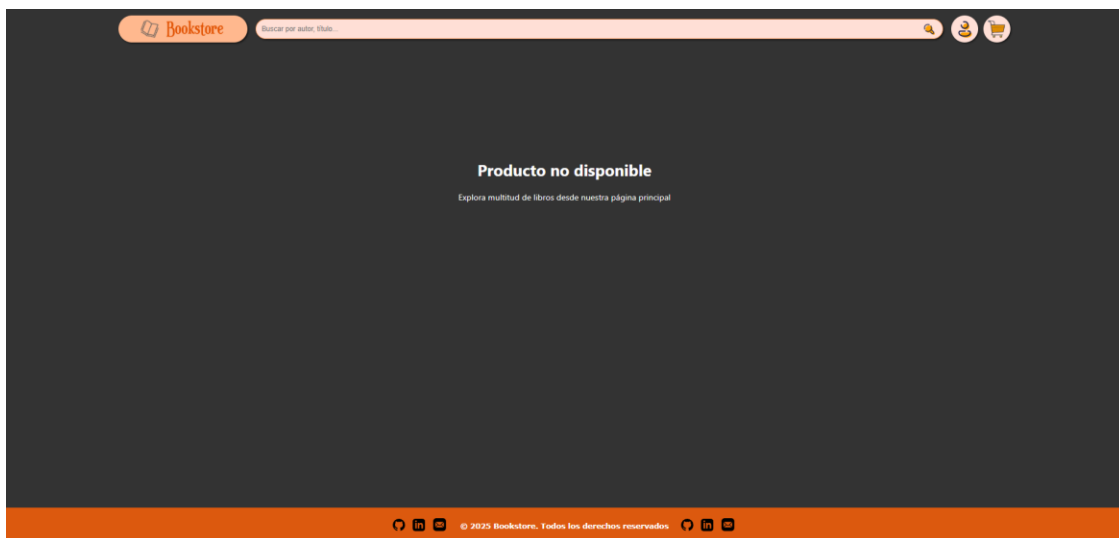
Perfil / Detalle de pedido

Páginas agregadas sin análisis previo

Posteriormente, se agregaron nuevas páginas, pero estas páginas surgieron de manera imprevista sin un análisis previo. Al ir realizando la web, nos fuimos dando cuenta de ciertas páginas que eran necesarias y que podíamos realizar sin análisis:

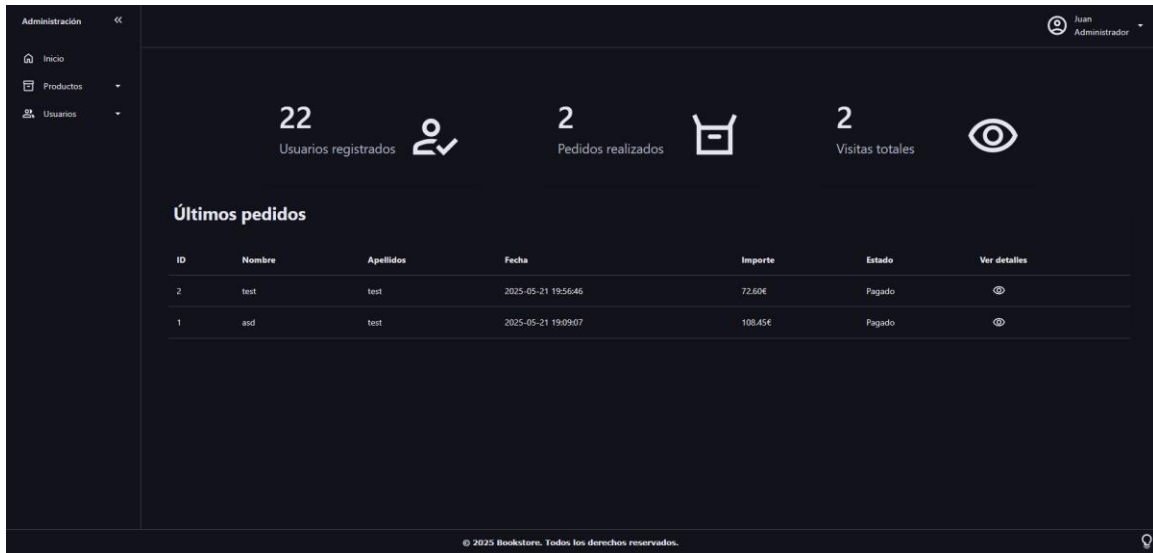
- Páginas de control de errores:

Por ejemplo, cuando el usuario intenta encontrar un libro que no existe, o un autor que no existe:



- **Páginas del panel de administrador:**

Como estas páginas no están diseñadas para el público, no vimos necesario hacer un análisis previo. Queríamos que fuera minimalista y sencilla:



3. Diseño del Proyecto

La interfaz de usuario fue diseñada siguiendo principios de **usabilidad** y **diseño responsive**, con el objetivo de ofrecer una experiencia clara, intuitiva y accesible para cualquier tipo de usuario, independientemente del dispositivo desde el que acceda a la plataforma.

Desde las primeras fases del diseño se tuvo en cuenta que la navegación debía ser sencilla y fluida. Por ello, se priorizó una **estructura limpia**, con menús visibles, jerarquías visuales claras y botones de acción bien diferenciados. Cada elemento fue colocado de forma estratégica para facilitar la toma de decisiones del usuario, evitando la sobrecarga visual y reduciendo al mínimo el número de clics necesarios para completar una acción, como encontrar un libro o finalizar una compra.

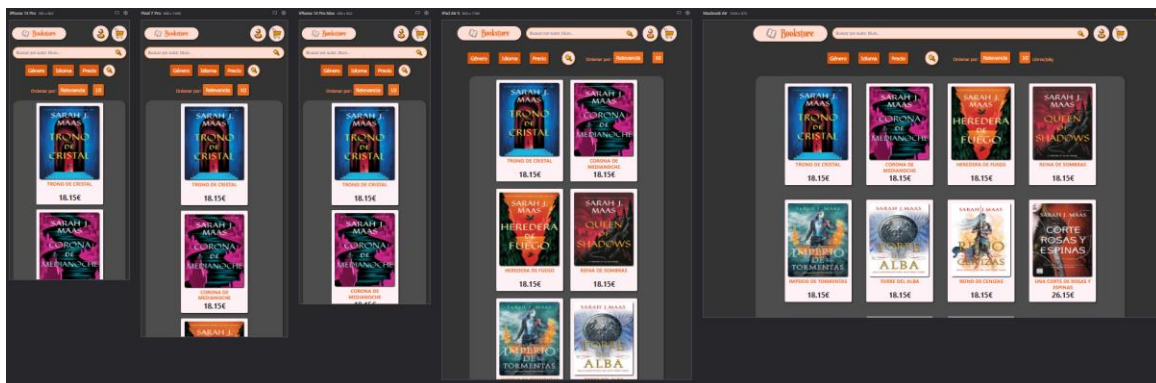
Además, se utilizaron combinaciones de colores suaves, tipografías legibles y un diseño visual equilibrado para crear una experiencia agradable que invite a permanecer en la web. También se implementaron elementos de interacción visual como hover en botones, recomendaciones dinámicas o alertas contextuales que mejoran la experiencia de uso sin saturar la navegación.

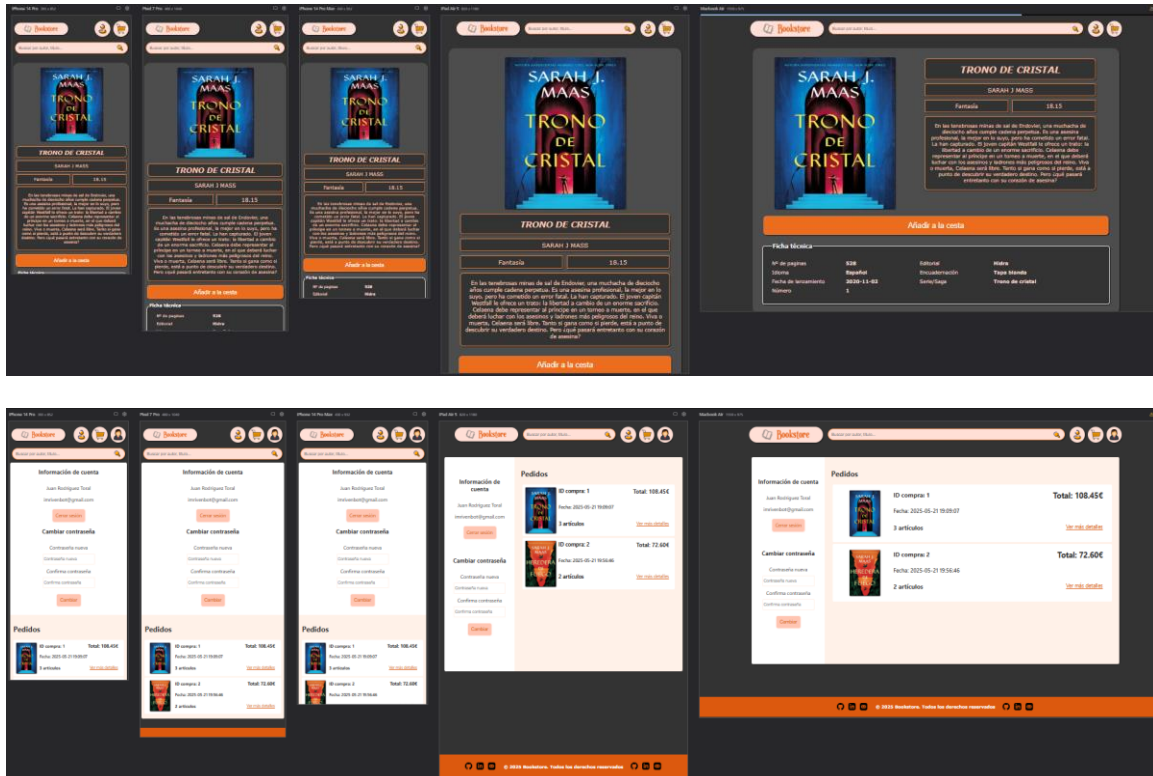
Responsive

En cuanto a la adaptabilidad, se utilizó un enfoque **responsive**, permitiendo que la interfaz se adapte automáticamente a diferentes tamaños de pantalla: ordenadores de sobremesa, tablets y smartphones. Esto garantiza que los usuarios puedan acceder a la tienda online en cualquier momento y desde cualquier lugar sin perder funcionalidades ni calidad visual.

Este enfoque orientado al usuario no solo mejora la accesibilidad y retención, sino que también aporta una imagen profesional a la plataforma, algo fundamental en un entorno tan competitivo como el comercio electrónico.

Ejemplos de la visualización de la web en diferentes dispositivos:





Paleta de colores

Hemos utilizado **variables en CSS** para definir y gestionar la **paleta de colores principal** empleada en toda la plataforma. Esta práctica no solo mejora la organización y legibilidad del código, sino que también facilita el mantenimiento y la coherencia visual del sitio a lo largo de todas sus páginas.

Al principio optamos por una gama cromática compuesta principalmente por **rosados** pero finalmente optamos por **tonos anaranjados, blancos y grises suaves**. Los tonos anaranjados se eligieron por su calidez y su asociación con la creatividad, la motivación y la acción, lo que encaja perfectamente con una tienda online dedicada a los libros. Estos colores se utilizaron principalmente en elementos interactivos como botones, encabezados o enlaces destacados, para captar la atención del usuario de manera sutil pero eficaz.

El **blanco** se empleó como color base, contribuyendo a una estética limpia, ordenada y minimalista que mejora la legibilidad del contenido y aporta sensación de amplitud visual. Por su parte, los **grises** se aplicaron en elementos secundarios como el fondo, bordes, menús, sombreados y zonas menos prioritarias, ayudando a establecer una jerarquía visual clara sin restar protagonismo a los contenidos principales.

```
:root {
  --sbg-color: #141414;
  --bg-color: #333;
  --bgl-color: rgb(77, 77, 77);
  --lighter-color: #fff1e8;
  --ll-color: #ffdfd6;
  --l-color: #ffc5b2;
  --lm-color: #ff9e5e;
  --m-color: #f28447;
  --md-color: #ec6d1e;
  --d-color: #dc590e;
}
```

Paleta de colores CSS

Estructura de la web

La mayoría de páginas de la web se componen de una **barra de navegación superior** que se mantiene fija en todo momento, facilitando el acceso rápido a las funciones más importantes. En esta barra se incluye el **logotipo de la web**, que actúa como acceso directo a la página de inicio, una **barra de búsqueda** para localizar libros de forma rápida, y accesos visibles tanto al **carrito de compra** como al **perfil del usuario** (o inicio de sesión si no está autenticado).

Justo debajo, se encuentra la **parte central o cuerpo principal**, donde se muestra el contenido específico de cada sección. Por ejemplo, en la página principal se muestra todo el catálogo de libros y el acceso a los mismo. En otras páginas, como la ficha de un producto, se presenta información detallada del libro seleccionado, mientras que en el panel de usuario o administrador se muestran formularios, listados o paneles de control.

Finalmente, en la parte inferior de todas las páginas, se encuentra el **footer o pie de página**, diseñado con un estilo simple y limpio. Este incluye enlaces de interés como información de contacto y enlaces a redes sociales. También sirve como cierre visual de la página y aporta coherencia estética a toda la plataforma.



4. Planificación de Fases del Proyecto

La planificación del proyecto se dividió en varias fases, cada una con objetivos definidos, tareas específicas y plazos estimados.

El desarrollo del proyecto **BookStore** comenzó en el verano de **2024**, impulsado inicialmente por **María José Salar García**, quien asumió tanto la parte de **frontend** como de **backend** del núcleo funcional de la plataforma. En esta primera fase se construyeron las páginas principales que componen la experiencia del usuario, incluyendo:

- index.php (página de inicio)
- libro.php (vista individual del producto)
- carrito.php (gestión del carrito de compra)
- formulariodecompra.php (formulario para finalizar pedidos)
- comprarealizada.php (confirmación de compra)
- autor.php (vista de autor)
- iniciodesesion.php y registro.php (autenticación de usuarios)
- perfil.php (gestión del perfil del usuario)

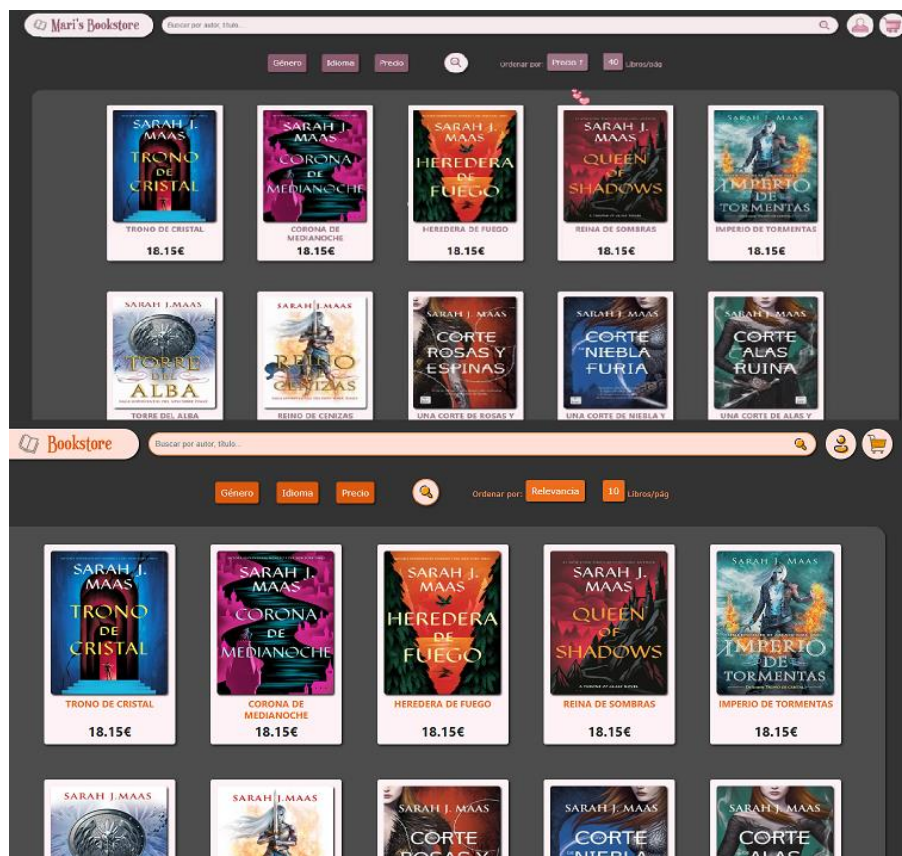
Esta versión inicial del proyecto fue denominada "**Mari BookStore 1.0**", y su desarrollo se prolongó durante aproximadamente **cinco meses**, en los que se estableció la arquitectura básica del sistema, se implementaron funcionalidades clave y se construyó una interfaz sencilla pero operativa.

Sin embargo, al finalizar esta etapa, se identificaron varias áreas de mejora, tanto en términos funcionales como estéticos. Fue entonces cuando se incorporó al equipo el compañero **Juan Antonio Rodríguez Toral**, quien participó activamente en la evolución del proyecto hacia una versión más avanzada y completa.

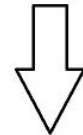
Durante los meses de **marzo a mayo de 2025**, Juan Antonio se encargó de implementar mejoras clave como:

- La **integración del método de pago PayPal** en el formulario de compra, garantizando transacciones seguras.
- La creación de un sistema de **roles de usuario**, diferenciando entre clientes y administradores.
- El desarrollo completo del **panel de administración**, tanto en la parte visual como en su lógica de backend, permitiendo gestionar productos, usuarios y pedidos.

Una vez que la aplicación fue completamente funcional y estable, María José retomó el trabajo en la plataforma para llevar a cabo una **refactorización estética y visual**. Se revisaron los estilos generales, se ajustaron los colores para obtener una estética más moderna y profesional, y se diseñó un nuevo logotipo. Estos cambios dieron lugar a una versión mejorada de la plataforma, que pasó a llamarse oficialmente "**BookStore 2.0**".



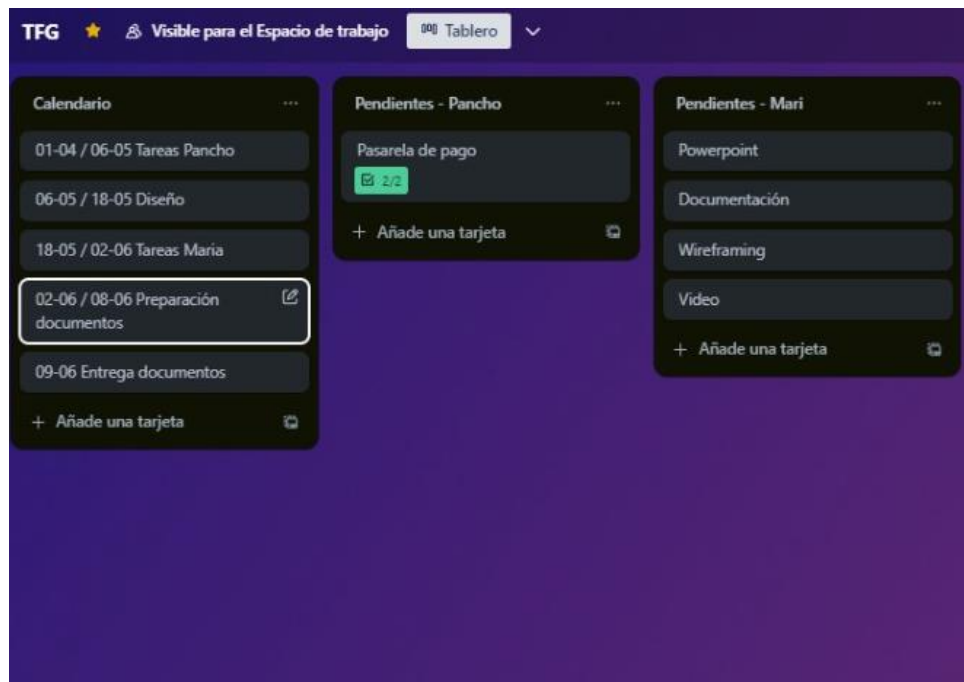
Antes



Después

Finalmente, entre mediados y finales de **mayo de 2025**, se elaboró toda la documentación técnica y formal del proyecto, así como los recursos necesarios para su presentación ante el tribunal. Esta fase concluyó con la preparación de esta memoria, la estructuración de los anexos y la organización de los materiales de entrega.

Para la planificación y organización del trabajo durante el desarrollo del proyecto, utilizamos la herramienta online Trello, una plataforma de gestión de tareas basada en el sistema de tableros y tarjetas. Esta herramienta nos permitió plasmar de forma visual y ordenada todas las tareas que teníamos pendientes, así como repartir responsabilidades entre los integrantes del equipo.



5. Desarrollo. Contenidos

Contenidos

A continuación, se detallan los contenidos del proyecto:

📁 public/assets

└─ Imágenes y recursos multimedia: banners, portadas, logotipos, vídeos demostrativos, iconos.

📁 Documentacion

└─ Documentos del proyecto: formalidades, estructura, documentación final.

📁 interface

└─ Recursos visuales: wireframes en Balsamiq, diagramas de interfaz, modelo entidad-relación.

📁 src

└─ assets/images (Recursos para la página)
| └─ author_photos: Fotos de autores.
| └─ covers: Portadas de libros.
└─ database
| └─ database.sql: script de creación y carga de base de datos.
└─ pages
| └─ Archivos PHP que corresponden a las vistas públicas del sistema.
└─ scripts
| └─ JavaScript para funcionalidades dinámicas, validaciones, sliders, etc.
└─ server
| └─ Lógica de servidor: controladores, consultas a la base de datos, gestión de usuarios y productos.
└─ styles
| └─ Hojas de estilo CSS separadas por módulos: panel de administración, métodos de pago, estilos globales.

└─ index.php Página principal de la web
└─ Read.me Usado en Github para dar información de la web

Frontend

La interfaz de usuario fue desarrollada utilizando HTML5, CSS3 y JavaScript, para facilitar la maquetación responsive y acelerar el diseño visual. Se diseñaron múltiples vistas adaptadas a distintos contextos de usuario.

HTML y CSS

En concreto, se definieron tres bloques principales de **@media queries** para cubrir los distintos rangos de dispositivos:

- 1- @media para **tablets o móviles** con grandes resoluciones

Esta regla está orientada a dispositivos como tablets en posición horizontal o móviles de gama alta que disponen de pantallas amplias. Se realizaron ajustes específicos en tamaños de fuente, espaciados y distribución de elementos para que se vieran correctamente sin necesidad de hacer zoom ni scroll horizontal.

```
@media all and (min-width: 701px) and (max-width: 1200px) {
```

- 2- @media para **móviles**

Pensada para dispositivos de pantalla más pequeña (como smartphones estándar), esta media query reorganiza el contenido en columnas verticales, oculta elementos no esenciales y agranda zonas táctiles (botones y enlaces) para facilitar la interacción.

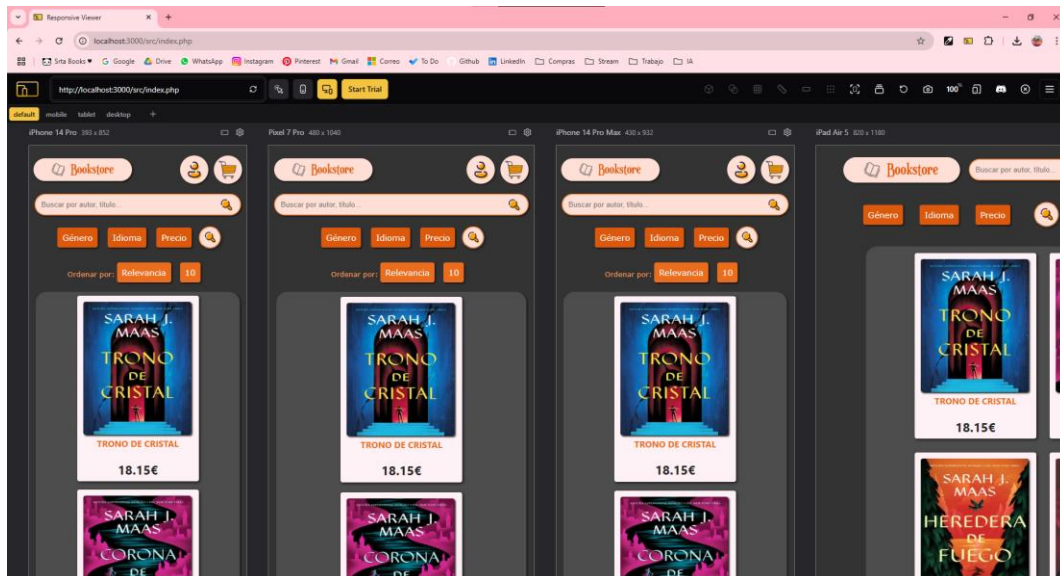
```
@media all and (max-width: 700px) {
```

- 3- @media para **ordenadores o tablets** con grandes resoluciones

Esta regla se aplica en pantallas a partir de resoluciones elevadas (como monitores Full HD o tablets en modo escritorio). En este contexto, se maximizan los márgenes laterales, se amplían los contenedores y se ajustan las imágenes o textos para que aprovechen el espacio disponible sin saturar la vista.

Para el diseño responsive se utilizó la **extensión de Google Chrome** llamada **Responsive Viewer**, una herramienta que permite simular de forma realista cómo se ve la web en diversos dispositivos y tamaños de pantalla simultáneamente. Esta extensión resultó especialmente útil para detectar desajustes, verificar la correcta

alineación de elementos, y comprobar que los menús, botones, formularios y textos mantenían su funcionalidad y estética en cualquier entorno.



También se ha realizado un uso estructurado y coherente de **IDs y clases en el código HTML y CSS**, lo que ha permitido acceder de forma precisa a cada elemento de la interfaz. Esta práctica es fundamental para mantener el proyecto **ordenado y escalable**, ya que facilita la aplicación de estilos específicos, la implementación de comportamientos dinámicos con JavaScript y la reutilización de componentes visuales en distintas secciones de la web.

```
#inicio {  
  height: 35px;  
  width: 228px;  
}
```

Se ha hecho uso de **variables de CSS** para albergar los colores, algo que facilita si se quiere modificar todo el color de la web:

```

:root {
  --sbg-color: #141414;
  --bg-color: #333;
  --bgl-color: rgb(77, 77);
  --lighter-color: #fff1e8;
  --ll-color: #ffdfd6;
  --l-color: #ffc5b2;
  --lm-color: #ff9e5e;
  --m-color: #f28847;
  --md-color: #ec6d1e;
  --d-color: #dc590e;
}

```

```

body {
  background-color: var(--bg-color);
}

```

Además del diseño visual y la estructura general de las páginas, se han incorporado una serie de **transiciones y efectos visuales** que mejoran notablemente la **interactividad y dinamismo** de la plataforma. Estos efectos tienen como objetivo aportar una experiencia más moderna, fluida y atractiva para el usuario, sin afectar al rendimiento ni sobrecargar visualmente el sitio.

```

.nav li a {
  display: flex;
  background-color: var(--ll-color);
  border-radius: 40px;
  padding: 8px;
  transition: background-color 0.3s;
  box-shadow: 2px 2px 3px var(--sbg-color);
}

```

Entre las animaciones más destacadas implementadas se encuentran:

- **Transiciones suaves en los botones:** al pasar el cursor sobre ellos, se produce un **cambio de color de fondo y de borde**, acompañado de una **ligera expansión** del botón, dando la sensación de que este “responde” a la interacción del usuario.
- **Efecto hover en las tarjetas de libros:** cuando se posiciona el ratón sobre la portada de un libro, esta **aumenta ligeramente su tamaño** y se aplica una

sombra sutil, lo que resalta visualmente el elemento activo y favorece la exploración del catálogo.

- **Animaciones en el menú desplegable y formularios:** algunos menús o formularios hacen uso de transiciones de opacidad y desplazamiento para que su aparición y desaparición sean suaves y agradables.
- **Iconos animados** en ciertos botones o acciones, como el carrito o los accesos rápidos, lo que mejora la visibilidad de las acciones disponibles.

JavaScript

Para dotar a la plataforma de una mayor **interactividad** y sensación de fluidez, se incorporaron diversos scripts en **JavaScript** que permiten generar comportamientos dinámicos sin necesidad de recargar la página constantemente. Estas funcionalidades mejoran significativamente la **experiencia del usuario**, haciendo que la navegación sea más intuitiva, atractiva y moderna.

Entre las implementaciones más destacadas se encuentran:

- **Menús desplegables tipo dropdown:** se desarrolló un sistema personalizado donde, al hacer clic sobre un menú, se despliega su contenido, y si otro estaba abierto, se cierra automáticamente. Esto mejora la organización visual y evita la saturación de información en pantalla.
- **Carrusel de libros recomendados:** al acceder a la ficha de un libro, se muestra un carrusel con productos similares que puede deslizarse de forma fluida, permitiendo al usuario explorar recomendaciones sin cambiar de página. Este carrusel proporciona continuidad visual y mantiene al usuario interesado en explorar el catálogo.
- **Modo oscuro en el panel de administración:** se incluyó un botón que activa o desactiva un modo oscuro para la interfaz del administrador. Esta funcionalidad, desarrollada con JavaScript, permite alternar los colores de fondo y texto dinámicamente, ofreciendo una mejor experiencia en entornos con poca luz o para usuarios que lo prefieran por comodidad.
- **Validación de formularios en tiempo real:** mediante JavaScript se implementaron comprobaciones automáticas en los formularios, mostrando mensajes en color rojo junto a los campos obligatorios cuando estos están vacíos o contienen datos incorrectos. Esto evita enviar formularios incompletos y reduce la frustración del usuario al no tener que esperar a una recarga para conocer los errores.

- **Detección del estado de compra:** el sistema es capaz de detectar si una compra ha sido cancelada o completada con éxito mediante los datos de retorno de la pasarela de pago (PayPal), y muestra mensajes informativos correspondientes en pantalla.
- **Alertas de confirmación en el panel de administración:** cada vez que el administrador intenta eliminar o modificar un producto o un usuario, se muestra una ventana de confirmación que pregunta si realmente desea ejecutar esa acción. Este tipo de validaciones previenen errores accidentales y refuerzan la seguridad operativa del sistema.

Backend y bases de datos (PHP y MySQL)

El backend del proyecto ha sido desarrollado íntegramente utilizando **PHP**, lo que ha permitido implementar la lógica de servidor y gestionar la comunicación con la **base de datos MySQL**. A través de múltiples scripts y funciones, se han podido llevar a cabo operaciones esenciales como la inserción, eliminación, actualización y consulta de datos en tiempo real.

Uno de los aspectos clave ha sido la correcta **conexión entre el frontend y la base de datos**, lo que se logra gracias a archivos PHP específicos encargados de realizar las consultas y devolver los resultados que luego se muestran dinámicamente en la interfaz de usuario.

Entre las funcionalidades desarrolladas destacan:

- **Inserciones en la base de datos:** como el registro de nuevos usuarios, la realización de pedidos, o la creación de nuevos libros desde el panel de administración.

```

k?php
include_once('getConnection.php');

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $titulo = $_POST['titulo'] ?? '';
    $genero = $_POST['genero'] ?? '';
    $editorial = $_POST['editorial'] ?? '';
    $n_pag = $_POST['n_pag'] ?? 0;
    $idioma = $_POST['idioma'] ?? '';
    $fecha_publ = $_POST['fecha_publ'] ?? '';
    $encuadernacion = $_POST['encuadernacion'] ?? 'Tapa dura';
    $precio = $_POST['precio'] ?? 0.00;
    $descripcion_libro = $_POST['descripcion_libro'] ?? '';
    $serie = $_POST['serie'] ?? null;
    $numero = $_POST['numero'] ?? null;
    $codigo_autor = $_POST['codigo_autor'];
    $activado = intval($_POST['activado'] ?? 1);

    $tipos_encuadernacion = ['Tapa dura', 'Tapa blanda'];
    if (in_array($encuadernacion, $tipos_encuadernacion)) {
        echo json_encode(['success' => false, 'message' => 'Tipo de encuadernación no válido.']);
        exit;
    }

    try {
        $sql = "INSERT INTO libro (
            titulo, genero, editorial, n_pag, idioma,
            fecha_publ, encuadernacion, precio,
            descripcion_libro, serie, numero, codigo_autor, activado
        ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

        $stmt = $conn->prepare($sql);
    }
}

```

- **Validación previa en el registro de usuarios:** antes de insertar un nuevo usuario en la base de datos, el sistema comprueba automáticamente si ya existe una cuenta asociada al correo electrónico introducido. Si es así, se muestra un mensaje de advertencia y no se realiza el registro, evitando duplicidades. Esta comprobación se realiza mediante una consulta previa en PHP que busca coincidencias en la tabla usuarios.

```

if (isset($_POST['registrarse'])) {
    $client_nombre = $_POST['nombre'];
    $client_apellidos = $_POST['apellidos'];
    $client_email = $_POST['email'];
    $client_pass = $_POST['password'];
    $client_pass_conf = $_POST['password_conf'];

    $regex = '/^[A-z0-9_-]+@[A-z0-9]{1,3}[A-z0-9-]*\.[A-z0-9-]{2,6}$/i';

    if (strlen($client_nombre) == 0 || strlen($client_apellidos) == 0 || strlen($client_email) == 0 || strlen($client_pass) == 0 || strlen($client_pass_conf) == 0) {
        header('location: registro.php?error=Rellena los campos vacíos');
        exit();
    } else if (!preg_match($regex, $client_email)) {
        header('location: registro.php?error_email=Introduce una dirección de correo electrónico válida');
        exit();
    } else if (strlen($client_pass) < 6) {
        header('location: registro.php?error_pass=La contraseña debe contener 6 caracteres o más');
        exit();
    } else if ($client_pass != $client_pass_conf) {
        header('location: registro.php?error_pass_conf=Las contraseñas no coinciden');
        exit();
    } else {
        $stmt1 = $conn->prepare("SELECT * FROM cliente WHERE email=?");
        $stmt1->bind_param('s', $client_email);
        $stmt1->execute();
        $stmt1->store_result();

        if ($stmt1->num_rows != 0) {
            header('location: registro.php?error_email=Ya existe una cuenta registrada con este email');
            exit();
        }
    }
}

```


- **Eliminación de datos:** se permiten acciones como la eliminación de productos, cancelación de pedidos o eliminación de cuentas de usuarios desde el panel de administrador.

```
<?php
include('getConnection.php');

if ($_SERVER['REQUEST_METHOD'] === 'DELETE') {

    if (isset($_GET['action']) && $_GET['action'] === 'deleteUser' && isset($_GET['userId']) && isset($_GET['email'])) {
        $userId = intval($_GET['userId']);
        $userEmail = $_GET['email'];

        if (deleteUserByIdAndEmail($userId, $userEmail)) {
            echo json_encode(['success' => true]);
        } else {
            echo json_encode(['success' => false, 'error' => 'No se pudo eliminar el usuario']);
        }
    } else {
        echo json_encode(['success' => false, 'error' => 'Parámetros incorrectos']);
    }
    exit();
}

function deleteUserByIdAndEmail($userId, $userEmail)
{
    global $conn;
    $stmt = $conn->prepare("DELETE FROM cliente WHERE codigo_cliente = ? AND email = ?");
    $stmt->bind_param("is", $userId, $userEmail);
    return $stmt->execute();
}
```

- **Actualización de registros:** el sistema permite modificar los datos de productos existentes, incluyendo su asociación con autores y categorías. También es posible actualizar la información de usuario desde su perfil personal.

```
<?php
include_once('getConnection.php');

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $codigo_libro = $_POST['codigo_libro'] ?? null;
    $titulo = $_POST['titulo'] ?? '';
    $genero = $_POST['genero'] ?? '';
    $editorial = $_POST['editorial'] ?? '';
    $n_pag = $_POST['n_pag'] ?? 0;
    $idioma = $_POST['idioma'] ?? '';
    $fecha_publ = $_POST['fecha_publ'] ?? '';
    $encuadernacion = $_POST['encuadernacion'] ?? '';
    $precio = $_POST['precio'] ?? 0.00;
    $descripcion_libro = $_POST['descripcion_libro'] ?? '';
    $serie = $_POST['serie'] ?? null;
    $numero = $_POST['numero'] ?? null;
    $codigo_autor = $_POST['codigo_autor'] ?? null;
    $activado = intval($_POST['activado'] ?? 1);

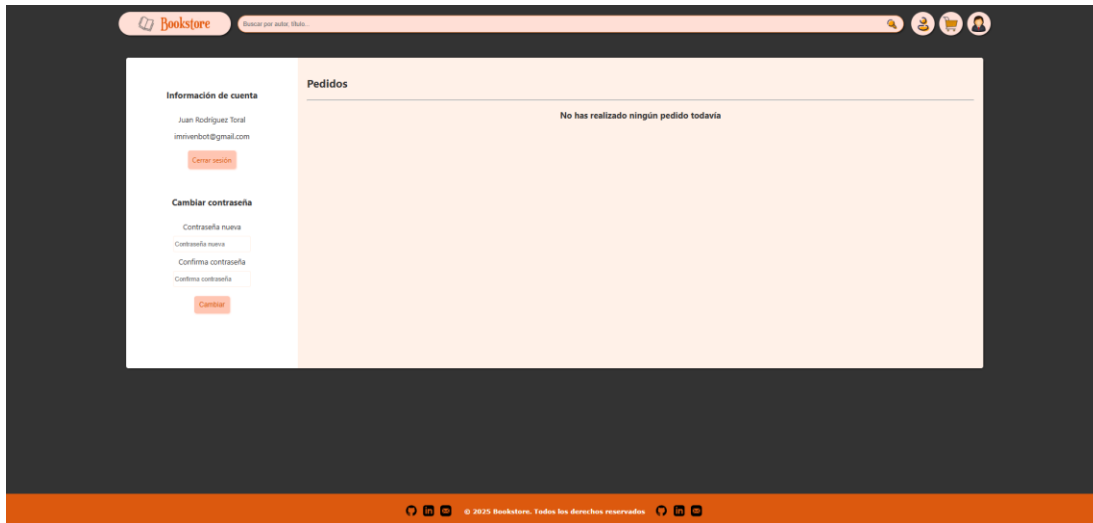
    if (!$codigo_libro) {
        echo json_encode(['success' => false, 'message' => 'Código del libro no proporcionado.']);
        exit();
    }

    try {
        $sql = "UPDATE libro SET
            titulo = ?,
            genero = ?,
            editorial = ?,
            n_pag = ?,
            idioma = ?,
            fecha_publ = ?,
            encuadernacion = ?,
            precio = ?,
            descripcion_libro = ?,
            serie = ?,
            numero = ?,
            codigo_autor = ?,
            activado = ?";
```

- **Consultas relacionales complejas:** se aprovecha el modelo entidad-relación diseñado para recuperar datos vinculados. Por ejemplo, se pueden obtener los libros asociados a un autor, los detalles de cada pedido con los libros comprados, y reconstruir el historial de compras completo desde la relación entre las tablas libro, detalle_compra y compra.



- **Visualización dinámica de datos:** los datos almacenados en la base de datos se muestran directamente en la web gracias a PHP. Cada página obtiene la información necesaria mediante consultas SQL y la refleja en tiempo real en la interfaz, ya sea al mostrar el catálogo, el perfil del usuario, el contenido del carrito o el panel de administración.
- **Gestión de sesiones y permisos:** PHP también se utiliza para iniciar sesión, mantener la sesión activa y controlar el acceso a ciertas secciones dependiendo del tipo de usuario (cliente o administrador). Con estas sesiones, también ha sido posible almacenar los productos del carrito en la cesta de la compra.



Además, se desarrolló una estructura limpia de funciones en PHP que permiten **mostrar dinámicamente los datos** en las distintas secciones de la web. Por ejemplo, al acceder a un libro, sus datos se obtienen en tiempo real desde la base de datos, y lo mismo ocurre con el carrito de compra, el perfil del usuario o el panel de administrador.

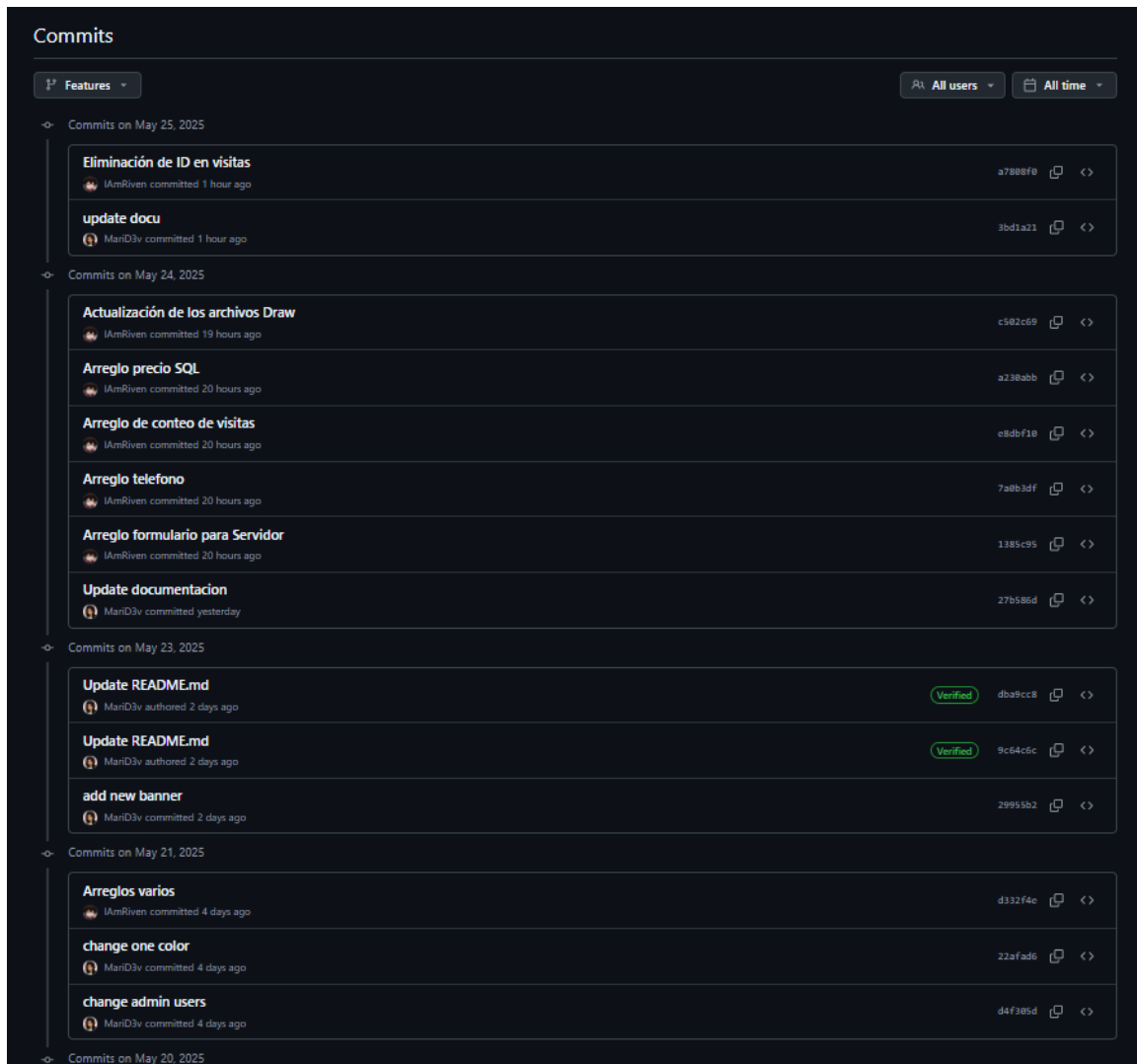
Todo este sistema de backend actúa como **punto entre la interfaz visual y la lógica de datos**, permitiendo que la plataforma funcione de forma dinámica, coherente y conectada. También se utilizó PHP para validar sesiones, controlar permisos según el tipo de usuario (cliente o administrador), y gestionar rutas y acciones críticas como el login, el registro y el procesamiento de pedidos.

6. Seguimiento de Fases del Proyecto

El seguimiento del proyecto se llevó a cabo de forma continua mediante herramientas digitales y revisión directa con el tutor. Desde el inicio se utilizaron **tablas de planificación** y la herramienta **Trello** para organizar y asignar tareas por fases.

Cada fase del proyecto fue revisada al completar sus objetivos principales, ajustando plazos y corrigiendo errores conforme se avanzaba. También se mantuvo un **registro de avances** a través del control de versiones con **GitHub** (<https://github.com/MariD3v/BookStore>), lo que permitió documentar cambios y facilitar la colaboración entre los participantes.

En **GitHub** hemos almacenado más de 150 commits (cambios) a lo largo de estos meses donde se ve reflejado el trabajo realizado:



The screenshot displays the 'Commits' page on GitHub, showing a list of commits grouped by date. The interface includes a search bar, filters for 'All users' and 'All time', and a list of commits with their messages, authors, and commit hashes.

Commits on May 25, 2025
Eliminación de ID en visitas IAMRiven committed 1 hour ago a7808f0
update docu MariD3v committed 1 hour ago 3bd1a21

Commits on May 24, 2025
Actualización de los archivos Draw IAMRiven committed 19 hours ago c502c69
Arreglo precio SQL IAMRiven committed 20 hours ago a238abb
Arreglo de conteo de visitas IAMRiven committed 20 hours ago e8dbf18
Arreglo telefono IAMRiven committed 20 hours ago 7a0b3df
Arreglo formulario para Servidor IAMRiven committed 20 hours ago 1389c95
Update documentacion MariD3v committed yesterday 27b586d

Commits on May 23, 2025
Update README.md MariD3v authored 2 days ago Verified dba9cc8
Update README.md MariD3v authored 2 days ago Verified 9c64c6c
add new banner MariD3v committed 2 days ago 29955b2

Commits on May 21, 2025
Arreglos varios IAMRiven committed 4 days ago d332f4e
change one color MariD3v committed 4 days ago 22afad6
change admin users MariD3v committed 4 days ago d4f385d

Commits on May 20, 2025

7. Resultados. Conclusiones. Propuestas Futuras

El desarrollo de **BookStore** ha supuesto un gran reto y, al mismo tiempo, una experiencia muy enriquecedora a nivel técnico y personal. A lo largo del proyecto hemos dedicado muchas horas, esfuerzo y concentración para dar forma a una aplicación web funcional, visualmente atractiva y alineada con los objetivos del ciclo formativo.

Durante el proceso nos hemos tenido que enfrentar a **muchos errores, pruebas, validaciones y correcciones**, sobre todo en cuanto a la gestión de usuarios, pagos y estructura del catálogo. También hemos aprendido a prever cómo puede interactuar el usuario con la plataforma, anticipándonos a posibles fallos para ofrecer una experiencia más robusta y controlada.

A pesar de ser un proyecto desarrollado por solo **dos personas y en un tiempo limitado**, los resultados obtenidos han sido muy satisfactorios. La plataforma es estable, dinámica y cumple con los requisitos fundamentales de una tienda online: gestión de productos, usuarios, pedidos, pagos y roles administrativos.

Pero, como todo proyecto real, aún hay margen de mejora. Algunas de las **propuestas futuras que hemos identificado** para seguir evolucionando la web son:

- Aplicar el **modo oscuro a toda la web**, no solo al panel de administración, permitiendo que el usuario elija su preferencia de visualización.
- Añadir **múltiples métodos de pago**, como tarjeta bancaria o Bizum, además de PayPal.
- Incluir más funcionalidades en el **panel de administración**, como gráficas estadísticas, logs de actividad o filtros avanzados.
- Rediseñar la **página principal** para que no sea simplemente el catálogo, sino que muestre secciones como "Novedades", "Libros más populares" o "Recomendados para ti".
- Aumentar el catálogo de libros o conectarse a **APIs públicas de libros**, como Google Books o Open Library.
- Añadir **opiniones o valoraciones** de usuarios para cada libro.
- Implementar un **sistema de favoritos o lista de deseos**.
- Mejorar el **sistema de búsqueda** con sugerencias en tiempo real y filtros combinados.
- Integrar un sistema de **notificaciones por correo electrónico** para confirmar pedidos o enviar avisos importantes.

- Optimizar el rendimiento y seguridad del sitio con medidas como validaciones del lado servidor, uso de HTTPS, protección CSRF/XSS y almacenamiento de sesiones de forma más segura.
- Implementar un sistema de **backup de la base de datos** automatizado y gestión de logs.

Estas ideas representan solo una parte del potencial que aún tiene BookStore como plataforma web. Lo más importante es que, más allá del resultado final, el proyecto nos ha permitido **disfrutar del proceso de construir algo propio desde cero**, aprender de los errores, resolver problemas reales y aplicar conocimientos que ahora sabemos usar de forma práctica.

8. Bibliografía

- Documentación oficial de PHP

<https://www.php.net/manual/es/>

- Documentación oficial de JavaScript (MDN Web Docs)

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

- MySQL Reference Manual

<https://dev.mysql.com/doc/>

- Manual oficial de CSS (W3C)








<https://www.w3.org/Style/CSS/Overview.en.html>

- GitHub Docs – Uso básico y gestión de repositorios

<https://docs.github.com/>

9. Anexos

A continuación, se detalla el contenido adicional que acompaña este proyecto y que se entrega como parte del conjunto completo:

-  **README del proyecto**
Documento inicial con una descripción general del sistema, funcionalidades, tecnologías utilizadas y versiones desarrolladas.
-  **Diagrama Entidad–Relación (E-R)**
Representación visual de la estructura de la base de datos y relaciones entre tablas, elaborado con Draw.io.
-  **Modelo de interfaz inicial (Paint)**
Boceto rápido de las pantallas principales de la web, utilizado como primer paso de análisis visual.
-  **Wireframes en Balsamiq**
Prototipos de baja fidelidad para definir la estructura y distribución de contenidos antes del desarrollo real.
-  **Este documento**
Memoria completa del proyecto BookStore, versión imprimible en PDF.
-  **Repositorio GitHub**
Código fuente completo del proyecto, accesible en línea:
 <https://github.com/MariD3v/BookStore>