



DEPLATEC - TIENDA WEB INFORMÁTICA

BORJA PATIÑO DEL VALLE

DAW2 – Desarrollo de
aplicaciones Web



ÍNDICE

1. INTRODUCCIÓN	2
1.1. PRESENTACIÓN DEL PROYECTO	2
1.2. OBJETIVOS DEL PROYECTO	3
1.3. JUSTIFICACIÓN DEL PROYECTO	4
2. ANÁLISIS DE REQUERIMIENTOS	5
2.1. IDENTIFICACIÓN DE NECESIDADES Y REQUERIMIENTOS	5
2.2. IDENTIFICACIÓN DE PÚBLICO	7
2.3. ESTUDIO DE MERCADO Y COMPETENCIA	8
3. DISEÑO Y PLANIFICACIÓN	10
3.1. DEFINICIÓN DE LA ARQUITECTURA DEL PROYECTO	10
3.2. DISEÑO DE LA INTERFAZ DE USUARIO	12
3.3. PLANIFICACIÓN DE LAS TAREAS Y LOS RECURSOS NECESARIOS	14
4. IMPLEMENTACIÓN Y PRUEBAS	16
4.1. DESARROLLO DE LAS FUNCIONALIDADES DEL PROYECTO	16
4.2. PRUEBAS UNITARIAS Y DE INTEGRACIÓN	19
4.3. CORRECCIÓN DE ERRORES Y OPTIMIZACIÓN DEL RENDIMIENTO	21
5. DOCUMENTACIÓN	23
5.1. DOCUMENTACIÓN TÉCNICA	23
5.2. DOCUMENTACIÓN DE USUARIO	25
5.3. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN	27
6. MANTENIMIENTO Y EVOLUCIÓN	29
6.1. PLAN DE MANTENIMIENTO Y SOPORTE	29
6.2. IDENTIFICACIÓN DE POSIBLES MEJORAS Y EVOLUCIÓN DEL PROYECTO	31
6.3. ACTUALIZACIONES Y MEJORAS FUTURAS	32
7. CONCLUSIONES	34
7.1. EVALUACIÓN DEL PROYECTO	34
7.2. CUMPLIMIENTO DE OBJETIVOS Y REQUISITOS	35
7.3. LECCIONES APRENDIDAS Y RECOMENDACIONES PARA FUTUROS PROYECTOS	37
8. BIBLIOGRAFÍA Y REFERENCIAS	39
8.1. FUENTES UTILIZADAS EN EL PROYECTO	39
8.2. REFERENCIAS Y ENLACES DE INTERÉS	40



1. INTRODUCCIÓN

1.1. PRESENTACIÓN DEL PROYECTO

El proyecto, denominado DEPLATEC INFORMÁTICA, se centra en la creación integral de una web para la gestión y venta de artículos informáticos. Básicamente, es una tienda online donde los usuarios pueden explorar un catálogo de productos, usar filtros por categorías, ver detalles y hacer sus pedidos. A su vez, el administrador cuenta con un panel para manejar todo en la tienda: productos, categorías, usuarios y pedidos.

La meta principal fue crear una aplicación práctica y que funcione bien, imitando un e-commerce real pero básico, donde aplicar los conocimientos aprendidos del curso de Desarrollo de Aplicaciones Web (DAW), como bases de datos relacionales, programación en ambos lados (backend y frontend), diseño adaptable, validación de formularios, seguridad y puesta en marcha local.

Para llevar a cabo este proyecto, he optado por Laravel como framework clave para el backend, Bootstrap para el diseño del frontend, MySQL para la gestión de la base de datos, y herramientas como JavaScript y AJAX para hacer más amigable la experiencia del usuario en ciertas funciones importantes.

Si bien el foco del proyecto es lo técnico, también he dedicado tiempo a la planificación, la documentación, el diseño de la interfaz y la parte visual. El proyecto ofrece funciones esenciales como la creación, edición, visualización y eliminación de productos (CRUD), gestión de pedidos, control de roles (usuario/administrador) y un filtrado dinámico por categorías usando AJAX.

Este proyecto muestra tanto el aprendizaje técnico logrado durante los dos años del ciclo como la habilidad para manejar por cuenta propia un proyecto más grande. Es un ejemplo real de cómo llevar una idea desde el análisis hasta su puesta en marcha, pasando por todas las fases del desarrollo profesional.



1.2. OBJETIVOS DEL PROYECTO

El objetivo principal de este proyecto ha sido desarrollar una aplicación web funcional que simule una tienda online de productos informáticos, donde se puedan aplicar de forma práctica los conocimientos aprendidos durante el ciclo de Desarrollo de Aplicaciones Web. Para ello, se han definido una serie de objetivos generales y específicos que han guiado el proceso de desarrollo.

Entre los objetivos generales se encuentra la creación de un sistema completo que permita la gestión de productos, categorías, pedidos y usuarios, tanto desde el punto de vista del cliente como desde el del administrador. Además, se busca que la aplicación tenga un diseño visual agradable, sea usable en distintos dispositivos (ordenador, tablet y móvil) y que cumpla unos mínimos requisitos de seguridad y organización del código.

Los objetivos específicos del proyecto han sido:

- Implementar un sistema de registro y login con roles diferenciados para usuario y administrador.
- Desarrollar CRUDs completos para productos, pedidos, categorías y usuarios, aplicando validaciones y relaciones entre tablas.
- Crear una interfaz clara y responsive usando Bootstrap.
- Integrar AJAX para ciertas funcionalidades (como el filtrado por categoría y la eliminación de registros) para mejorar la experiencia de usuario.
- Asegurar que el acceso a determinadas rutas esté protegido mediante middleware y control de roles.
- Organizar el proyecto según el modelo MVC (Modelo-Vista-Controlador), siguiendo las buenas prácticas de Laravel.
- Documentar de forma clara el funcionamiento del sistema y los pasos para su instalación.

Todos estos objetivos han sido planteados desde el principio del proyecto, y se han ido adaptando ligeramente según las necesidades y limitaciones surgidas durante el proceso.



1.3. JUSTIFICACIÓN DEL PROYECTO

Desde el inicio del segundo curso del ciclo de Desarrollo de Aplicaciones Web (DAW), tenía claro que para mi Trabajo de Fin de Ciclo quería desarrollar un proyecto que me permitiera aplicar de forma real todo lo aprendido durante estos dos años. Por eso, elegí crear una aplicación web de tipo tienda online, ya que combina muchas de las áreas que se trabajan en el ciclo: bases de datos, backend, frontend, diseño responsive, seguridad y usabilidad.

He decidido centrarme en una tienda de productos informáticos porque es un sector que me resulta familiar y atractivo. Además, me ha permitido trabajar con ejemplos reales (productos, categorías, pedidos, etc.) que hacen que el proyecto tenga un sentido práctico y una utilidad clara, tanto como ejercicio académico como si en un futuro se quisiera llevar a producción o ampliar. Crear una tienda online también es una de las prácticas más completas que se pueden plantear en este contexto, porque implica trabajar con múltiples entidades relacionadas, formularios, validaciones, control de usuarios y gestión de datos.

Otro motivo por el que he elegido este proyecto es porque me permite demostrar autonomía, organización y capacidad técnica. A diferencia de los ejercicios guiados del aula, aquí he tenido que enfrentarme a decisiones de arquitectura, organización del código, diseño visual, etc., sin depender de un enunciado cerrado. Eso me ha obligado a investigar, leer documentación y buscar soluciones reales a problemas que han surgido durante el desarrollo, lo cual considero muy valioso.

Por último, este proyecto también me servirá como carta de presentación. Tener un desarrollo completo con documentación, código bien organizado y presentación oral es una buena forma de demostrar mis competencias a futuras empresas o incluso para seguir aprendiendo por mi cuenta. Además, Laravel y Bootstrap son herramientas utilizadas en el mundo profesional, lo cual hace que este proyecto tenga también una orientación hacia la inserción laboral.



2. ANÁLISIS DE REQUERIMIENTOS

2.1. IDENTIFICACIÓN DE NECESIDADES Y REQUERIMIENTOS

Para empezar con la aplicación web como es debido, fue clave empezar por un buen análisis de lo que el sistema debía hacer y cómo debía ser. Esta etapa inicial es súper importante porque pone los cimientos del proyecto y así evitamos meter la pata después.

Para empezar, algo básico es que los usuarios puedan ver los productos informáticos de forma fácil y que se entienda todo a la primera. Eso significa que el catálogo tiene que estar bien ordenado, con fotos, precios y que se pueda buscar por tipo de producto. Además, como en cualquier tienda online de verdad, el usuario necesita poder registrarse, entrar con su usuario y hacer pedidos. Así que, manejar usuarios y pedidos es algo que no puede faltar.

También tuve en cuenta que hacía falta un panel de administración para tener control sobre lo que hay en la tienda. El administrador tiene que poder añadir productos, cambiarlos, quitarlos, manejar las categorías y ver los pedidos que han hecho los usuarios. Esto es fundamental porque así mantenemos el sistema al día sin tener que meternos directamente en la base de datos.

Respecto a la parte técnica, el sistema tenía que hacerse con Laravel como base principal, ya que es lo que hemos estado usando durante el curso y permite usar el modelo MVC sin problemas. Además, el diseño tenía que ser adaptable a cualquier pantalla, así que use Bootstrap para que la web se vea bien en móviles, tablets y ordenadores.

Otro punto importante era que el sistema fuera rápido y que el usuario no se desesperara al usarlo. Para lograrlo, use AJAX en algunas partes, como al filtrar productos por categoría, para que la información se cargue al instante sin tener que



refrescar la página. También revise las validaciones tanto en la parte visible como en la interna para que los datos estén siempre correctos.

Por último, decidí que la aplicación debía estar organizada, que otros programadores pudieran entenderla, que tuviera manuales técnicos y para el usuario, y que fuera fácil instalarla en un ordenador local (XAMPP, etc.).

En pocas palabras, este análisis previo fue clave para decidir qué cosas tenía que tener la tienda online y cómo debía funcionar, tanto para el usuario como para el que la administra. Gracias a esta etapa, el desarrollo después fue más fácil y con las cosas claras.



2.2. IDENTIFICACIÓN DE PÚBLICO

Un aspecto clave al momento de construir cualquier sitio web es definir bien a quién va dirigido. Esto facilita las elecciones sobre el diseño, el idioma en la interfaz, las opciones que se añaden y cuán complejo será el sistema. En el caso de DEPLATEC INFORMÁTICA, podemos separar a los usuarios en dos grandes tipos: los usuarios comunes y los que administran el sistema.

Por un lado, los usuarios comunes serían personas que quieren comprar cosas de informática. Esta gente no tiene por qué saber mucho de tecnología, así que el sitio debe ser claro, fácil de usar y que entre por los ojos. La idea es que puedan hallar sin problemas lo que buscan, informarse bien y hacer sus compras sin líos. Por eso, se buscó que la navegación sea fácil, el diseño se adapte a cualquier pantalla y los formularios sean sencillos.

Ellos pueden mirar el catálogo y ver los productos, pero para comprar algo tienen que registrarse antes. Esto nos ayuda a tener un seguimiento de las compras y saber quién las hizo.

Por otro lado, los administradores del sistema son el segundo tipo de usuario. Este grupo sí necesita saber algo más de tecnología, porque deben encargarse de manejar la tienda: agregar productos, ponerlos en su lugar, cambiar la información, borrar cosas y revisar las compras hechas. Para ellos, creamos un panel de administración con acceso limitado y control de permisos con Laravel.

En cuanto a la edad de la gente a la que apuntamos, creo que va desde jóvenes interesados en tecnología que ya son mayores de edad, hasta adultos de cualquier edad que necesiten comprar algo de informática. En cualquier caso, no todos saben lo mismo de tecnología, así que la interfaz es simple y fácil de entender.



2.3. ESTUDIO DE MERCADO Y COMPETENCIA

Antes de proseguir con la elaboración de la aplicación, consideré oportuno realizar un breve análisis de la competencia y el mercado actual. Aunque el trabajo es académico, encontré útil conocer lo que las tiendas online actuales tienen que ofrecer y cómo dichas ideas se podrían trasladar a un proyecto como DEPLATEC INFORMÁTICA. Además, este análisis también me ha aportado ideas en lo que a diseño, funcionalidades, y distribución en el sitio web se refiere.

Actualmente, el mercado de ventas online de productos informáticos está muy consolidado. Hay grandes portales como PCComponentes, Amazon o Coolmod, que tienen miles de productos tecnológicos a precios ciertamente competitivos, con envío rápido y sistemas de compra muy eficientes y optimizados. Dichos sitios tienen filtros avanzados, comparadores de productos, opiniones de la comunidad de clientes y asistencia online. Claramente, sería muy difícil competir con este tipo de portales de manera profesional, pero sí se pueden sacar ideas valiosas de ellos.

Por ejemplo, una de las características más habituales de las tiendas en estos lugares es la capacidad de filtrar los productos entre categorías (algunos ejemplos serían portátiles, monitores, ratones, etc.). Esto era algo que también decidí incorporar en mi aplicación con AJAX, ya que lo que hace es incrementar la interacción con el usuario y facilitar la navegación a la hora de buscar lo que queramos buscar. Otra de las ideas que sustraje es presentar los productos con una tarjeta gráfica con el título, la imagen, el precio y un botón de mostrar más información. Esto facilita que el usuario pueda obtener una vista rápida de lo que se encuentra en el catálogo.

A nivel de diseño, también me di cuenta de que la gran mayoría de las tiendas emplean una estructura muy sencilla: menú principal, categorías visibles y un footer con contenido adicional. Esta estructura facilita que el usuario no se extravíe y pueda navegar con soltura. Por ello traté de reproducir la misma lógica con Bootstrap, adecuando el diseño a varios tamaños de pantalla.



Además de las grandes tiendas, también estuve analizando algunas de las tiendas más pequeñas o locales, con páginas web más básicas pero operativas. A muchas de ellas se las hace con WordPress o Prestashop, y les permiten administrar productos y órdenes a un nivel básico. A estos tipos de webs he recurrido en busca de ejemplos reales, ya que un nivel técnicamente más similar a lo que puedo conseguir con Laravel en segundo de DAW es el que tienen ellas.

Por ello, este pequeño estudio de mercado me sirvió para entender mejor lo que un usuario espera cuando entra en una tienda online y cuáles son las características obligatorias para que la experiencia resulte satisfactoria.



3. DISEÑO Y PLANIFICACIÓN

3.1. DEFINICIÓN DE LA ARQUITECTURA DEL PROYECTO

Para construir una aplicación web sólida, fiable y que se pueda desarrollar con éxito, es necesario plantear una arquitectura sólida desde un primer momento. En mi caso, escogí Laravel como framework, por lo que la app sigue el patrón MVC (Modelo-Vista-Controlador), separando la lógica del negocio, la interfaz y el manejo de la información.

Laravel es una herramienta potente que vimos en el segundo año de DAW, y me ayudó a ordenar el código. El patrón MVC hace que cada parte esté clara: los modelos acceden a la base de datos, los controladores gestionan las peticiones y las vistas muestran los datos al usuario.

La aplicación tiene varias partes importantes:

- Área pública (productos): cualquiera registrado puede entrar, ver productos, buscar por tipo, ver datos básicos y gestionar sus pedidos.
- Área de usuario registrado: para que el usuario vea su perfil, haga pedidos y mire su historial.
- Área de administración: solo para administradores. Aquí se manejan productos, categorías, pedidos y usuarios.

La estructura de carpetas de Laravel también ayuda a mantener todo organizado. Las rutas están en `web.php`, los controladores en `app/Http/Controllers`, los modelos en `app/Models` y las vistas en `resources/views`. Para el diseño utilice Bootstrap, que me permitió tener estilos responsivos y un buen aspecto rápido.

Además, la app usa una base de datos relacional (MySQL), con relaciones entre productos, categorías, usuarios y pedidos. Para ello, definí migraciones y modelos con relaciones `hasMany` y `belongsTo`, usando Eloquent, el ORM de Laravel.



También usé middlewares para proteger rutas y evitar que usuarios sin permiso entren al panel de administración. Esto da seguridad de forma centralizada y evita problemas de acceso.

En resumen, la arquitectura del proyecto es profesional, basada en MVC y aprovechando todo lo que ofrece Laravel. Esto hace que la app sea fácil de mantener, crecer o mejorar en el futuro, y me ayudó a usar lo aprendido en el ciclo de forma práctica.



3.2. DISEÑO DE LA INTERFAZ DE USUARIO

Uno de los aspectos más importantes de una aplicación web es la experiencia del usuario (UX) y la forma en que se presenta la información (UI). Desde el principio del desarrollo de DEPLATEC INFORMÁTICA quise crear una interfaz clara, atractiva y funcional, que ofreciera una navegación fluida tanto en ordenador como en dispositivos móviles. Para ello, he utilizado Bootstrap como framework de diseño, ya que permite implementar rápidamente componentes responsivos y visuales de calidad.

La interfaz se ha diseñado teniendo en cuenta tres tipos de usuario:

- Usuario registrado: puede iniciar sesión, ver el catálogo, detalles de producto, realizar pedidos y consultar su historial.
- Administrador: tiene acceso a los paneles de gestión de productos, categorías, usuarios y pedidos.

La estructura general de la aplicación se organiza en varias vistas:

Página principal (Inicio):

- El menú superior permite navegar a otras secciones y cambia dinámicamente según el tipo de usuario.
- Se visualizan los productos mediante tarjetas que incluyen imagen, nombre, precio y botón de “Ver más”.
- Integra filtros dinámicos por categoría mediante AJAX, sin necesidad de recargar la página.
- Se utiliza un diseño de columnas adaptativo que responde al tamaño de la pantalla.

Detalle de producto:

- Muestra información ampliada del producto: descripción, imagen, precio y botón para pedido.
- Diseño centrado y limpio, para facilitar la lectura.



Área de usuario:

- Permite ver pedidos realizados, cancelar pedidos pendientes y actualizar sus datos personales.
- La interfaz se ha simplificado para que sea intuitiva incluso para usuarios con pocos conocimientos técnicos.

Panel de administración:

- Accesible solo por administradores mediante middleware.
- CRUDs para productos, categorías, pedidos y usuarios. Cada sección muestra una tabla con los registros y botones de acción (editar, borrar, ver).
- Formularios validados tanto en cliente (con JavaScript) como en servidor (Laravel).

Estéticamente, he seguido una línea minimalista y moderna, con colores neutros (blancos, grises, azules), tipografía clara y botones bien visibles. Todo el diseño está pensado para que el usuario no se pierda, con una navegación muy accesible. Además, gracias a Bootstrap, la web es totalmente responsive, por lo que se adapta automáticamente a móviles, tablets o monitores grandes.

A nivel técnico, he reutilizado plantillas base (layouts) para evitar duplicar código y mantener coherencia visual entre las diferentes páginas. También se ha separado el CSS en archivos específicos cuando ha sido necesario.



3.3. PLANIFICACIÓN DE LAS TAREAS Y LOS RECURSOS NECESARIOS

Para poder llevar a cabo el desarrollo del proyecto DEPLATEC INFORMÁTICA de forma organizada y dentro del tiempo establecido, ha sido fundamental realizar una buena planificación previa. Esta planificación me ha permitido distribuir el trabajo por fases, establecer tiempos aproximados y anticipar posibles dificultades. Aunque algunos plazos han tenido que ajustarse sobre la marcha, contar con una guía inicial me ha ayudado a avanzar de forma constante.

He dividido el desarrollo en varias fases:

Fase de análisis

- Definición del proyecto y objetivos.
- Estudio de necesidades y análisis de otros sitios similares.
- Diseño de los modelos de base de datos.

Fase de diseño

- Estructura de la base de datos en Laravel (migraciones).
- Maquetación inicial de las vistas con Bootstrap.
- Diseño del flujo de usuarios (registrado, admin).

Fase de desarrollo

- Implementación del backend en Laravel.
- Creación de controladores, modelos y rutas.
- CRUD de productos, categorías, pedidos y usuarios.
- Lógica de autenticación y roles (middleware).
- Filtrado de productos con AJAX.



Fase de pruebas y corrección

- Comprobación del correcto funcionamiento de cada funcionalidad.
- Pruebas con usuarios (alta, pedidos, restricciones).
- Revisión de la interfaz en distintos dispositivos.
- Solución de errores detectados.

Fase de documentación y presentación

- Redacción del documento técnico del proyecto.
- Elaboración del PowerPoint para la defensa.
- Preparación de la exposición oral.

A nivel de herramientas y recursos, he utilizado:

- Laravel 12 como framework principal de desarrollo backend.
- MySQL como sistema gestor de bases de datos.
- Bootstrap 5 para el diseño visual y la adaptación responsive.
- Visual Studio Code como editor de código.
- Git para el control de versiones.
- XAMPP para el entorno de desarrollo local.



4. IMPLEMENTACIÓN Y PRUEBAS

4.1. DESARROLLO DE LAS FUNCIONALIDADES DEL PROYECTO

Durante esta fase, se desarrollaron todas las funcionalidades principales que dan vida a la tienda DEPLATEC INFORMÁTICA. El proyecto fue implementado utilizando Laravel como framework backend, aprovechando su arquitectura MVC (Modelo-Vista-Controlador), y Bootstrap en el frontend, logrando así una interfaz adaptable y actual.

A continuación, detallo las funcionalidades más importantes y cómo las he ido implementando:

1. Gestión de productos

Esta es una de las funcionalidades clave del sistema. Los usuarios con rol administrador pueden crear, editar, ver y eliminar productos. Cada producto cuenta con detalles como el nombre, descripción, stock, precio, imagen y categoría.

- Se ha implementado un CRUD completo utilizando los métodos index, create, store, show, edit, update y destroy del controlador ProductoController.
- Las imágenes se almacenan físicamente en la carpeta /public/imagenes y se guarda la ruta en la base de datos.
- Se validan los datos tanto en el servidor (Request) como en el frontend.

2. Gestión de categorías

Las categorías permiten organizar los productos para facilitar la navegación del usuario. Solo los administradores pueden crear, modificar o eliminar categorías.

- Cada producto pertenece a una única categoría, definida mediante una relación de clave foránea.
- Al eliminar una categoría, se verifica si tiene productos asociados para evitar errores de integridad referencial.



3. Sistema de autenticación y roles

- El sistema distingue entre dos roles: usuario normal y administrador.
- Los middlewares protegen las rutas según el rol, por ejemplo, los usuarios normales no pueden acceder al panel de usuarios.
- Los usuarios registrados pueden acceder a sus pedidos, hacer nuevos pedidos o cancelar los existentes.

4. Gestión de pedidos

El usuario registrado puede seleccionar productos y realizar pedidos, los cuales quedan asociados a su cuenta.

- Cada pedido contiene tanto los datos del producto como los datos del pedido como los datos del usuario.
- Se ha implementado una relación de muchos a muchos entre pedidos y productos a través de una tabla intermedia.
- El usuario puede ver su historial de pedidos y cancelar los pedidos que haya realizado si así lo desea.

5. Filtrado dinámico de productos (AJAX)

Una de las funcionalidades clave del proyecto es el filtrado por categoría sin recargar la página.

- Utilizando jQuery y AJAX, al seleccionar una categoría se hace una petición a una ruta específica que devuelve los productos en formato JSON.
- Los productos se renderizan dinámicamente en la vista sin necesidad de recarga, mejorando la experiencia de usuario.

6. Panel de administración

El administrador tiene acceso a un panel donde puede gestionar todo el contenido de la tienda.



- Desde el panel puede crear, editar o eliminar productos, categorías y ver los pedidos de todos los usuarios.
- Se ha diseñado un layout específico para diferenciar visualmente la zona de administración.

7. Página “Sobre mí”

Se ha añadido una sección que presenta al desarrollador del proyecto, explicando brevemente su formación, experiencia y el propósito de esta tienda online.

- Esta página se puede visualizar desde el menú principal y refuerza el carácter académico del proyecto.



4.2. PRUEBAS UNITARIAS Y DE INTEGRACIÓN

Una vez desarrolladas las funcionalidades principales de la aplicación, se llevó a cabo un proceso de verificación para asegurar que todo funcionase correctamente. Dado que este proyecto ha sido desarrollado principalmente como práctica académica, se combinaron pruebas manuales con pruebas funcionales y, en algunos casos, pruebas automatizadas básicas propias del framework Laravel.

Pruebas manuales

La mayor parte del testeo se ha realizado de forma manual, comprobando el funcionamiento correcto de cada parte del sistema desde el punto de vista del usuario. Para ello, se creó una lista de comprobación con los siguientes objetivos:

- Verificar que un usuario puede registrarse, iniciar sesión y acceder a sus funcionalidades correspondientes.
- Comprobar que un usuario normal no puede acceder a rutas protegidas por middleware de administrador.
- Probar la creación, edición y eliminación de productos y categorías como administrador.
- Confirmar que el filtrado de productos mediante AJAX funciona correctamente sin recargar la página.
- Validar que un usuario puede realizar un pedido, ver sus pedidos y cancelar alguno si así lo desea.
- Comprobar que las imágenes se cargan correctamente desde la ruta guardada en la base de datos.



Pruebas de integración

Se realizaron pruebas completas recorriendo todo el proceso que seguiría un usuario real:

- Registrarse como nuevo usuario.
- Navegar por los productos disponibles.
- Filtrar por categoría usando AJAX.
- Seleccionar productos y generar un pedido.
- Consultar el historial de pedidos y cancelar alguno si es necesario.

Del mismo modo, se comprobó todo el recorrido de gestión por parte del administrador:

- Acceso al panel de administración.
- Creación y edición de productos.
- Gestión de categorías y usuarios.
- Gestión de pedidos de todos los usuarios.

Conclusiones de las pruebas

Gracias a las pruebas realizadas, tanto manuales como automatizadas, se detectaron y solucionaron errores como:

- Fallos en la validación de formularios al insertar productos.
- Acceso no autorizado a ciertas rutas por parte de usuarios normales.
- Problemas con la carga de imágenes cuando el archivo era demasiado grande.



4.3. CORRECCIÓN DE ERRORES Y OPTIMIZACIÓN DEL RENDIMIENTO

Durante el desarrollo, surgieron diversos errores y percances, algo bastante común en cualquier proyecto de fin de curso. Gracias a las pruebas realizadas me fue posible identificar estos errores a tiempo y corregirlos antes de la entrega final.

Principales errores corregidos

- Validación de formularios:
 - En las vistas de creación y edición de productos, inicialmente no se realizaban validaciones completas del lado del servidor. Se implementaron validaciones con Laravel utilizando reglas como required, numeric y image, lo que evitó la introducción de datos erróneos o malformados.
- Problemas con los permisos:
 - Al principio, los usuarios normales podían acceder a algunas rutas restringidas mediante URL directa. Este fallo se solucionó aplicando correctamente middlewares como auth y admin para proteger las rutas sensibles.
- Gestión de imágenes:
 - A veces, las imágenes no aparecían bien, ya fuera porque las rutas estaban mal hechas o por fallos al cambiar los ficheros de sitio. Se le echó un vistazo al sistema para subir y guardar imágenes, verificando que los archivos quedaran guardados en la carpeta public/imagenes y que las rutas se guardasen bien en la base de datos.
- Eliminación de pedidos:
 - Cuando se eliminaba un pedido, el stock del producto no volvía a su cantidad original. Se añadió un sistema para que al borrar un pedido, la cantidad pedida.



Optimización del rendimiento

Además de corregir errores, se realizaron mejoras orientadas a optimizar el rendimiento de la aplicación, tanto en velocidad de carga como en eficiencia de consultas:

- AJAX en el filtrado de productos:
 - Para mejorar la experiencia de usuario, se integró AJAX en el filtrado de productos por categoría. Esto permitió que la página no se recargara completamente al aplicar filtros, haciendo la navegación más rápida y fluida.
- Optimización de imágenes:
 - Se añadió una restricción en el tamaño de las imágenes subidas y se aconsejó el uso de formatos optimizados (.jpg o .webp), para agilizar el rendimiento de la página y que el usuario no suba otro formato.



5. DOCUMENTACIÓN

5.1. DOCUMENTACIÓN TÉCNICA

En este apartado se recoge la documentación técnica del proyecto DEPLATEC INFORMÁTICA. Esta documentación describe los aspectos técnicos más importantes del backend y frontend, así como las herramientas y tecnologías utilizadas.

Tecnologías empleadas

- Laravel 12: Framework PHP utilizado para el desarrollo del backend, incluyendo el sistema de rutas, controladores, modelos, vistas, migraciones y middleware.
- MySQL: Sistema de gestión de base de datos relacional para almacenar información sobre usuarios, productos, pedidos y categorías.
- Bootstrap 5: Framework CSS utilizado para el diseño y maquetación del frontend.
- JavaScript y AJAX: Lenguaje y tecnología utilizados para mejorar la experiencia de usuario en acciones como el filtrado dinámico y la eliminación de productos sin recargar la página.
- Git: Sistema de control de versiones usado durante todo el desarrollo para mantener un historial del proyecto y trabajar de forma ordenada.

Estructura del proyecto

El proyecto Laravel está organizado con la estructura típica del framework. Los directorios principales son:

- app/
 - Models/: contiene los modelos como Producto, Pedido, Categoria y User.
 - Http/Controllers/: controladores como ProductoController, PedidoController, CategoriaController.
 - Http/Middleware/: archivos que gestionan la autenticación y autorización.



- resources/views/: contiene las vistas Blade del frontend organizadas en carpetas (productos, pedidos, categorías, auth, etc.).
- routes/web.php: define las rutas principales de la aplicación.
- database/migrations/: contiene las migraciones que crean las tablas de la base de datos.

Migraciones y modelos

Para cada entidad principal (Producto, Pedido, Categoría, Usuario) se ha creado su respectivo modelo y migración, estableciendo relaciones entre ellos mediante funciones como belongsTo(), hasMany() y belongsToMany() cuando ha sido necesario.

Middleware y roles

Se ha implementado un middleware para controlar el acceso a ciertas rutas, garantizando que solo los usuarios con rol admin puedan acceder a funciones como crear, editar o eliminar productos, pedidos y categorías. El middleware se registra en Kernel.php y se aplica a rutas específicas en web.php.

Rutas y controladores

Las rutas están organizadas en el archivo routes/web.php. Se han agrupado por tipo de usuario (normal o admin) y protegidas con los middleware necesarios. Cada controlador contiene los métodos típicos del CRUD (index, create, store, edit, update, destroy) para cada entidad.

AJAX y funcionalidades dinámicas

Se han implementado funcionalidades dinámicas mediante JavaScript y AJAX, como:

- Filtrado de productos por categoría sin recargar la página (productos/index.blade.php).
- Eliminación de productos, categorías, pedidos mediante botón con confirmación y petición AJAX.



5.2. DOCUMENTACIÓN DE USUARIO

Esta documentación está dirigida al usuario final de la aplicación DEPLATEC INFORMÁTICA. Su objetivo es explicar, de forma sencilla y clara, cómo utilizar las funcionalidades principales de la plataforma. La web está pensada para ser intuitiva, responsive y fácil de usar, tanto en ordenadores como en dispositivos móviles.

Tipos de usuario

- Usuario registrado (cliente): puede navegar por el catálogo de productos, realizar pedidos y consultar su historial.
- Usuario administrador: además de las funciones anteriores, puede gestionar productos, usuarios, categorías y pedidos de todos los clientes desde el panel de administración.

Acceso y registro

- El usuario puede registrarse desde el enlace “Registrarse” disponible en el menú superior. Se solicitan datos básicos como nombre, email y contraseña.
- Si ya está registrado, puede iniciar sesión desde “Iniciar sesión”.
- Una vez autenticado, se muestra un menú adaptado según su rol.

Funcionalidades para el usuario cliente

- Ver productos: desde la página principal o el apartado “Productos”, el usuario puede ver todos los artículos disponibles con su nombre, imagen, precio y categoría.
- Filtrar productos: puede seleccionar una categoría para filtrar los productos sin recargar la página (función AJAX).
- Hacer un pedido: cada producto tiene un botón “realizar pedido”. Al pinchar se puede realizar un pedido del producto.
- Consultar pedidos: desde el menú “Mis pedidos”, el cliente puede ver un listado con los pedidos realizados, incluyendo sus detalles.
- Cancelar pedido: el cliente puede cancelar un pedido desde su listado.



Funcionalidades para el usuario administrador

- Panel de administración: visible tras iniciar sesión con un usuario con rol admin.
- Gestión de productos:
 - Ver listado de productos con opciones para ver, realizar pedido ,editar y eliminar (eliminación vía AJAX).
 - Crear nuevos productos indicando nombre, descripción, imagen, precio y categoría.
- Gestión de categorías:
 - Crear, editar o eliminar categorías desde el menú correspondiente.
- Gestión de pedidos:
 - Ver todos los pedidos realizados por los clientes.
 - Ver detalles de pedidos.
 - Cancelar cualquier pedido si es necesario.

Usabilidad y diseño

- Toda la web está diseñada con Bootstrap 5, lo que permite un diseño responsive adaptable a móviles, tablets y ordenadores.
- Se ha cuidado la legibilidad, colores, espaciado y estructura para facilitar la navegación.



5.3. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

Este manual está destinado a cualquier persona que necesite desplegar, instalar o configurar el proyecto en un entorno de desarrollo o producción. A continuación, se describen los pasos necesarios para poner en funcionamiento la aplicación, tanto el backend (Laravel) como el frontend (Blade con Bootstrap y AJAX).

Requisitos del sistema

Para instalar el proyecto localmente, se necesita:

- PHP > 8.1
- Composer
- Laravel 12
- MySQL
- Servidor web (Apache)
- Git (opcional para clonar el repositorio)

1. Clonar el repositorio

Si se desea trabajar con control de versiones, clonar el proyecto desde Git:

- `git clone https://github.com/usuario/proyectoTFG-Borja-Patino.git`
- `cd proyectoTFG-Borja-Patino`

2. Instalar dependencias

Instalar las dependencias de Laravel con Composer:

`composer install`

`npm run dev`



3. Configurar variables de entorno

Editar las siguientes variables:

APP_NAME="Deplatec Informática"

APP_URL=http://localhost

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=productostfg

DB_USERNAME=root

DB_PASSWORD=

4. Crear y migrar la base de datos

Crear la base de datos con el nombre definido en .env (por ejemplo, productostfg) y ejecutar las migraciones o si lo desea importar la base de datos proporcionada:

```
php artisan migrate
```

5. Iniciar el servidor local

```
php artisan serve
```

La aplicación estará disponible en <http://localhost:8000>

6. Crear usuarios admin y cliente

Puedes registrar usuarios desde la propia web o crearlos manualmente desde localhost/phpmyadmin.

Para acceder con usuario administrador, crear usuario y cambiar rol en base de datos.



6. MANTENIMIENTO Y EVOLUCIÓN

6.1. PLAN DE MANTENIMIENTO Y SOPORTE

El mantenimiento de una aplicación web es fundamental para garantizar su correcto funcionamiento a largo plazo, así como para adaptarse a posibles cambios tecnológicos, errores que puedan surgir o nuevas necesidades del usuario. En el caso del proyecto, se ha planteado un plan básico de mantenimiento dividido en tres niveles: correctivo, adaptativo y evolutivo.

Mantenimiento correctivo

Este tipo de mantenimiento tiene como objetivo resolver errores detectados por los usuarios o durante las pruebas posteriores a la puesta en producción. Las acciones incluidas en este nivel son:

- Corrección de errores en el frontend (como fallos en formularios, enlaces o validaciones).
- Reparación de errores del backend (como consultas mal definidas, problemas en controladores o errores de acceso a la base de datos).
- Revisión de funcionalidades que no respondan como se espera.

Mantenimiento adaptativo

Conforme evolucionen los navegadores, sistemas operativos o versiones del propio framework Laravel, será necesario adaptar el código para asegurar la compatibilidad. Esto incluye:

- Actualización de Laravel, PHP y sus dependencias mediante Composer.
- Revisión de posibles funciones obsoletas o depreciadas.
- Ajuste del frontend para que siga siendo compatible con nuevos navegadores y dispositivos.



Mantenimiento evolutivo

Este tipo de mantenimiento busca mejorar y ampliar las funcionalidades del sistema según las nuevas necesidades de los usuarios o el negocio. Algunas ideas que podrían implementarse a futuro son:

- Incorporación de pasarelas de pago para permitir compras directas desde la web.
- Implementación de un sistema de valoraciones y comentarios para productos.
- Mejora del sistema de gestión de pedidos, incluyendo estados más detallados.
- Sistema de notificaciones por correo electrónico para avisar al cliente del estado del pedido.
- Mejora del panel de administración con estadísticas y gráficos sobre pedidos y productos más vendidos.

Soporte

- Crear un apartado de “Preguntas frecuentes” en la web.
- Habilitar un formulario de contacto para incidencias técnicas.
- Asignar un correo de soporte técnico (por ejemplo: soporte@deplatec.com).
- Crear documentación técnica clara para que cualquier desarrollador pueda intervenir sin depender exclusivamente del autor original.

Periodicidad

- Revisión mensual de logs y posibles incidencias técnicas.
- Actualización de dependencias cada 2-3 meses (Laravel, Bootstrap, PHP).
- Valoración anual de nuevas funcionalidades o rediseño de la interfaz.



6.2. IDENTIFICACIÓN DE POSIBLES MEJORAS Y EVOLUCIÓN DEL PROYECTO

Si bien el proyecto DEPLATEC INFORMÁTICA cumple con los objetivos y funciona a la perfección, siempre hay margen para mejorar y así enriquecer tanto al usuario como la gestión interna. A continuación, se presentan ideas y caminos a seguir que salieron a la luz durante el desarrollo y las pruebas:

Mejoras funcionales

- Sistema de pagos online: Ahora mismo, la compra se completa sin pago electrónico. Lo ideal sería integrar una pasarela de pago segura (tipo Redsys o PayPal), para que los usuarios paguen sus pedidos de manera cómoda y fiable desde la plataforma.
- Cesta de productos: Se podría añadir un carrito para que los usuarios registrados pidan varios productos a la vez.
- Gestión de stock automática: Implementar un sistema que actualice el stock tras cada pedido, evitando errores o ventas de productos agotados.
- Sistema de valoraciones: Poner un sistema de comentarios y puntuaciones de los clientes sobre los productos, lo que da confianza y anima a la interacción entre usuarios.

Mejora de accesibilidad y experiencia de usuario

- Diseño responsive mejorado: Aunque la interfaz se adapta a las resoluciones, se puede usar un enfoque mobile-first más avanzado, con menús sencillos y navegación táctil optimizada.
- Modo oscuro: Poner una opción de modo oscuro para adaptarse a los gustos visuales de los usuarios y evitar la fatiga visual, sobre todo en sitios con poca luz.
- Filtros avanzados de productos: Añadir filtros como marca, rango de precio, valoración o tipo de producto, para facilitar la búsqueda en el catálogo.
- Interfaz de usuario más atractiva: Se podría renovar el diseño con componentes modernos, animaciones sutiles y un diseño más minimalista y limpio que mejore la usabilidad.
- Sistema multidioma: Traducir la aplicación a idiomas como inglés o francés facilitaría su expansión y la haría accesible a más gente.



6.3. ACTUALIZACIONES Y MEJORAS FUTURAS

Una faceta clave en la trayectoria de cualquier aplicación es cómo evoluciona. Si bien DEPLATEC INFORMÁTICA se creó para cumplir requisitos específicos, es vital considerar su futuro, su soporte y cómo crecerá. A continuación, se detallan varias acciones de soporte y mejoras, clasificadas según el tiempo estimado: corto, medio o largo plazo. Esta planificación facilitará la adaptación del proyecto a nuevas necesidades, potenciará el desempeño y refinará la experiencia de los usuarios con el paso del tiempo.

Actualizaciones a corto plazo (entre 1 y 6 meses)

- Actualizar el framework Laravel y sus dependencias: Mantener Laravel al día con su versión estable más reciente es clave para asegurar un desempeño óptimo, arreglos de seguridad y buena relación con nuevas tecnologías.
- Mejorar el sistema de gestión de pedidos: Sumar estados como "Preparando", "Enviado", "Entregado" o "Cancelado", visibles desde el perfil del cliente, facilitaría un seguimiento más preciso del pedido.
- Optimizar el código: Refactorizar controladores y consultas para pulir la claridad, eficiencia y facilidad de mantenimiento del backend.
- Revisar el diseño adaptable y la accesibilidad: Ajustar la interfaz para refinar la experiencia en móviles, realzar el contraste visual, añadir etiquetas ARIA y facilitar la navegación con teclado, haciéndola más inclusiva.
- Cesta de productos: Se podría instaurar el sistema de carrito para que los usuarios registrados puedan tramitar pedidos de varios productos a la vez.
- Mejorar el diseño de la interfaz de usuario: Hacer un rediseño más moderno y coherente usando componentes visuales más atractivos y animaciones suaves, con un enfoque centrado en móviles.

Mejoras a medio plazo (entre 6 meses y 1 año)

- Integrar una pasarela de pago: Activar el pago en línea con opciones como tarjeta bancaria, PayPal o Bizum sería un gran paso en la automatización de la compra y elevaría la percepción de profesionalidad.



- Mejorar el panel de administración: Sumar estadísticas de ventas, productos estrella, ingresos mensuales y permitir exportar pedidos en formatos tipo CSV o PDF para simplificar la gestión empresarial.
- Ampliar el catálogo y categorías: Sumar nuevas secciones como “accesorios móviles”, “periféricos gaming” o “componentes reacondicionados”, adaptándose así a las tendencias del mercado.
- Traducir la web a varios idiomas: Incorporar versiones en inglés y francés para alcanzar un público más amplio.



7. CONCLUSIONES

7.1. EVALUACIÓN DEL PROYECTO

Durante el tiempo que dediqué a crear DEPLATEC INFORMÁTICA, pasé por cada etapa, desde entender qué se necesitaba hasta poner el sistema en marcha. Esta travesía fue muy valiosa para mí, tanto en lo que aprendí de informática como en lo personal. Me dio la oportunidad de usar todo lo que había aprendido en mi curso de Desarrollo de Aplicaciones Web.

Desde el punto de vista técnico, el proyecto alcanzó lo que se quería al principio. Se diseñó y construyó una aplicación web que funciona bien, con un sistema para manejar productos, pedidos, categorías de productos y usuarios. Se usó la estructura MVC con Laravel, se empleó herramientas modernas como Bootstrap y AJAX para que la gente disfrute más al usarla, y se trabajó bien con bases de datos y cómo se relacionan entre sí.

Además, se añadieron formas de controlar quién puede acceder a qué, usando diferentes roles, comprobaciones tanto en la parte visible como en la parte interna, y se organizó el código de manera clara, lo que hace que sea más fácil mantenerlo y mejorarlo en el futuro.

En cuanto a cómo fue crear todo esto, me enfrenté a problemas reales de cómo diseñar, pensar lógicamente y planificar. También aprendí a dividir el trabajo en partes, decidir qué es más importante, arreglar errores rápidamente y escribir todo el proceso de manera clara, lo cual es clave en cualquier proyecto profesional. En resumen, creo que el proyecto salió muy bien. La aplicación funciona sin problemas, se ve bien, es útil en la vida real y cumple con lo que se esperaba. Además, se puede ampliar en el futuro, lo que muestra que se pensó en grande desde el principio.



7.2. CUMPLIMIENTO DE OBJETIVOS Y REQUISITOS

El principal objetivo de este proyecto era desarrollar una tienda online funcional para una empresa ficticia llamada DEPLATEC INFORMÁTICA. Esta debía incluir un sistema de gestión de productos, pedidos, usuarios y categorías, diferenciando entre administradores y usuarios normales, todo ello con un diseño moderno, una buena estructura de código y funcionalidades realistas propias de un entorno profesional. A continuación, detallo cómo se han cumplido cada uno de los objetivos establecidos:

Objetivos generales:

- **Desarrollar una aplicación web completa con frontend y backend:**
- Se ha implementado utilizando Laravel como framework backend, junto a Blade, Bootstrap y JavaScript para el frontend. Se ha seguido la arquitectura MVC.
- **Incorporar funcionalidades CRUD completas para productos, pedidos y categorías (solo para administradores):**
- El rol de administrador tiene acceso completo para crear, editar, eliminar y ver productos, pedidos y categorías.
- **Permitir a los usuarios registrarse, iniciar sesión y realizar pedidos:**
- Implementado con éxito usando el sistema de autenticación propio de Laravel. El usuario puede ver productos y gestionar sus propios pedidos.

Requisitos técnicos:

- **Uso de relaciones entre tablas:**
- Se han creado relaciones entre usuarios, pedidos, productos y categorías mediante Eloquent ORM.
- **Implementación de middleware:**
- Se han utilizado middlewares para controlar el acceso a determinadas rutas según el rol del usuario.
- **Manejo de imágenes de productos:**
- Las imágenes se suben correctamente y se almacenan con su ruta en la base de datos.



- **Funcionalidades con AJAX:**
- Se ha implementado AJAX para filtrar productos por categoría y para eliminar productos sin recargar la página, mejorando la experiencia del usuario.
- **Diseño responsive y uso de Bootstrap:**
- Toda la interfaz se adapta correctamente a distintos dispositivos, utilizando Bootstrap y estructura grid/flex.

Otros elementos destacados:

- **Sección “Sobre mí”:**
- Añadida para cumplir con los requisitos del proyecto, explicando mi formación y motivaciones.
- **Código estructurado, comentado y organizado:**
- Toda la aplicación ha sido desarrollada con buenas prácticas, facilitando su mantenimiento futuro.



7.3. LECCIONES APRENDIDAS Y RECOMENDACIONES PARA FUTUROS PROYECTOS

Este proyecto ha sido una experiencia asombrosa, tanto para adquirir conocimientos como para crecer en el ámbito personal y profesional. Al desarrollar DEPLATEC INFORMÁTICA, me enfrenté a retos que me sacaron de mi zona de confort, me hicieron aplicar lo aprendido durante el ciclo y, más importante aún, me motivaron a descubrir nuevas herramientas y métodos por mi cuenta.

Una de las enseñanzas más esenciales fue la importancia de una buena planificación desde el inicio. Invertir tiempo en definir con claridad los modelos, las relaciones entre las tablas y el flujo de trabajo en las etapas iniciales del proyecto me evitó muchos inconvenientes más adelante. Asimismo, comprendí que no todo sale bien en el primer intento, y que es fundamental saber adaptarse y encontrar soluciones durante el proceso de desarrollo web.

Otra lección significativa fue la relevancia de gestionar las versiones y mantener el código organizado. Aunque este proyecto no fue realizado en equipo, estructurar el código de forma clara, empleando buenas prácticas, comentarios y una adecuada organización de las carpetas, resultó vital para no perderme a medida que el proyecto crecía.

Además, entendí lo importante que es mejorar la experiencia del usuario utilizando tecnologías como AJAX, que hacen la aplicación más fluida e interactiva. El uso de Bootstrap y el diseño responsive también me permitieron desarrollar una interfaz más profesional que se adapta a diversos dispositivos.

Si tuviera que ofrecer algunas sugerencias para futuros proyectos, diría:

- Planificar adecuadamente desde el inicio, especialmente la base de datos y el diseño general.
- Implementar y probar en partes, para prevenir errores que sean complicados de detectar después.
- No dejar la documentación ni los detalles menores para el final.
- Siempre buscar maneras de mejorar la experiencia del usuario, tanto en funcionalidad como en diseño.
- No dudar en consultar la documentación oficial, foros o recursos en línea.



- En conclusión, este proyecto me ha preparado de manera más efectiva para enfrentar desarrollos más complejos en el futuro y me ha motivado a seguir aprendiendo sobre nuevas tecnologías y métodos de desarrollo.



8. BIBLIOGRAFÍA Y REFERENCIAS

8.1. FUENTES UTILIZADAS EN EL PROYECTO

Para el desarrollo de este proyecto, he consultado múltiples fuentes de información que me han servido tanto para resolver dudas técnicas como para mejorar la calidad del código y del diseño. Entre las más destacadas se encuentran:

- <https://laravel.com/docs> (laravel)
- <https://getbootstrap.com/docs> (bootstrap)
- <https://stackoverflow.com> (stackoverflow)
- <https://stackoverflow.com/questions/tagged/laravel> (stackoverflow)
- <https://www.w3schools.com> (w3schools)
- <https://kinsta.com/es/blog/laravel-crud/> (kinsta)
- <https://laravel.com/docs/12.x/controllers> (laravel)
- <https://dcreations.es/blog/laravel/usar-laravel-ajax-jquery> (dcreations)
- https://www.w3schools.com/bootstrap/bootstrap_ver.asp (w3schools)
- <https://www.w3schools.com/mysql/default.asp> (w3schools)
- <https://www.w3schools.com/php/default.asp> (w3schools)
- https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/Network_requests (ajax)
- <https://dcreations.es/blog/laravel/como-hacer-login-en-laravel> (dcreations)
- <https://jquery.com/> (jQuery)
- <https://api.jquery.com/jQuery.ajax/> (jQuery)
- <https://kinsta.com/es/blog/autenticacion-laravel/> (kinsta)
- <https://es.stackoverflow.com/questions/323546/c%C3%B3mo-crear-formulario-log-in-laravel-b%C3%A1sico> (stackoverflow)
- <https://documentacionlaravel.com/docs/11.x/validation> (laravel)
- <https://kinsta.com/es/blog/laravel-validacion/> (kinsta)



8.2. REFERENCIAS Y ENLACES DE INTERÉS

A continuación, incluyo algunos enlaces de interés que me han ayudado a desarrollar mi proyecto y/o que pueden ser útiles :

- <https://www.php.net/manual/es/> (php)
- <https://getbootstrap.com/docs/5.3> (bootstrap)
- <https://www.postman.com/> (postman)
- <https://code.visualstudio.com/docs> (visual studio code)
- <https://github.com> (github)
- <https://aulasciclos2425.castillalamancha.es/> (educamos)
- <https://www.apachefriends.org/docs/> (xampp)
- <https://laravel.com/docs/4.2/quick> (laravel)
- <https://getcomposer.org/> (composer)
- <https://getcomposer.org/doc/> (composer)
- <https://www.w3schools.com/html/default.asp> (w3schools)
- <https://www.w3schools.com/css/default.asp> (w3schools)
- <https://www.w3schools.com/js/default.asp> (w3schools)
- <https://laravel-docs.com/es/docs/10.x/views> (laravel)
- <https://www.npmjs.com> (npm)
- <https://stackoverflow.com/questions/32498328/how-to-add-jquery-in-laravel-project> (stackoverflow)
- <https://www.reddit.com/r/laravel/> (Reddit)
- https://www.youtube.com/results?search_query=laravel (youtube)
- <https://tailwindcss.com/> (tailwind)
- <https://kinsta.com/es/blog/laravel-breeze/> (kinsta)
- <https://www.cursosdesarrolloweb.es/blog/laravel-breeze-la-solucion-moderna-de-autenticacion-para-laravel> (cursosdesarrolloweb)
- <https://www.cursosdesarrolloweb.es/blog/crud-laravel> (cursosdesarrolloweb)