

  GENERALITAT VALENCIANA CONSELLERIA D' EDUCACIO, INVESTIGACIO, CULTURA I ESPORT	 Unió Europea Fons Social Europeu L'FSE inverteix en el teu futur	CFGS Desarrollo de Aplicaciones Web
		Tecnocraft
		Memoria - Proyecto Web Fin de Ciclo
		Junio 2025

Tecnocraft

Memoria del Proyecto

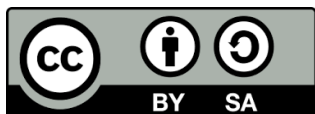
Autor: Manel Albella Galàn

Tutor: Jaume Rosa Matamoros

Vila-real, 19 de 05 de 2025

Licencia

La licencia elegida para la documentación del proyecto es la de **Creative commons Reconocimiento-Compartir Igual**.



Estas son las características de la licencia:

Usted es libre de:

- Copiar, distribuir y comunicar públicamente la obra.
- Generar obras derivadas.

Bajo las condiciones siguientes:

- Reconocimiento: Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
- Compartir bajo la misma licencia: Si altera o transforma esta obra, o genera una obra derivada, solo puede distribuir la obra generada bajo una licencia idéntica a esta.

Entendiendo que:

- Renuncia: alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.
- Dominio Público: Cuando la obra o alguno de sus elementos se halle en el dominio público según la ley vigente aplicable, esta situación no quedaría afectada por la licencia.
- Otros derechos: Los derechos siguientes no quedan afectados por la licencia de ninguna manera:
 - Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
 - Los derechos morales del autor.
 - Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo derechos de imagen o de privacidad.
- Aviso: Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Índice de contenidos

1. Presentación del proyecto	4
1.1. Introducción	4
1.2. Objetivos	4
1.3. Tecnologías y herramientas empleadas	4
1.4. Planificación temporal	5
2. Estudio de viabilidad	6
2.1. Establecimiento del alcance del sistema	6
2.2. Estudio de la situación actual	7
2.2.1. Nuestra aplicación (Siglas de tu aplicación)	7
2.3. Definición de los requisitos del sistema	7
2.4. Estudio de la solución adoptada	8
2.5. Presupuesto	8
2.6. Análisis de riesgos	9
3. Análisis del sistema	11
3.1. Definición del sistema	11
3.2. Establecimiento de requisitos	11
3.2.1. Subsistema 1	12
3.2.2. Subsistema 2	12
3.2.3. Subsistema n	12
3.3. Definición de interfaces de usuario	13
3.3.1. Página principal	13
3.3.2. Página de autenticación	14
3.3.3. Página de	14
3.3.4. Página de ayuda	14
3.4. Mapa de navegación	15
4. Diseño del sistema	15
4.1. Arquitectura	15
4.1.1. Identificación de subsistemas	16
4.1.2. Subsistema 1	17
4.1.3. Subsistema 2	18
4.1.4. Subsistema n	18
4.2. Diseño de la base de datos	18
4.3. Elección de alternativas de componentes y licencias más adecuadas	19
4.4. Especificaciones de desarrollo	20
4.4.1. Necesidades de equipos	20
4.4.2. Necesidades software	20

4.5. Especificación del plan de pruebas	21
4.5.1. Pruebas Unitarias	21
4.5.2. Pruebas de Implantación	21
5. Desarrollo	22
5.1. Desarrollo en el lado del cliente	22
5.2. Desarrollo en el lado del servidor	22
6. Conclusiones	22
6.1. Nivel de satisfacción general	22
6.2. Propuestas de mejoras	22
7. Bibliografía	22

NOTA: El texto marcado como “Ejemplo: ” o que está entrecomillado y subrayado, está como referencia para que, si os va bien, lo adaptéis a vuestro proyecto particular y que, una vez adaptado, sustituya al que hay actualmente (eliminando “Ejemplo: ” en el caso de los ejemplos, y el texto entrecomillado y subrayado).

NOTA: El resto del texto, si se adapta a vuestro proyecto, podéis dejarlo tal cual está, completando las partes que sean necesarias. Esto es un esquema típico de memoria, con los apartados que suelen incluirse, pero podéis redactar vuestra memoria de otro modo si lo consideráis conveniente.

NOTA: Si tomáis este documento como referencia, eliminado estas 3 notas.

1. Presentación del proyecto

1.1. Introducción

Este proyecto consiste en el desarrollo de una aplicación web enfocada en el comercio electrónico, específicamente en la venta de productos tecnológicos y de gaming / tiene su campo de actuación en el sector del comercio electrónico (e-commerce), la plataforma está diseñada para ofrecer una experiencia de usuario intuitiva, con un diseño atractivo y una navegación clara

1.2. Objetivos

El objetivo principal de este proyecto es crear una aplicación Web funcional, dinámica y fácil de usar que permita a los usuarios realizar compras de productos de informática y gaming desde cualquier lugar y dispositivo.

A continuación se describen los objetivos del proyecto en detalle:

Ejemplo:

- Mostrar información sobre productos disponibles, incluyendo nombre, descripción, precio, imagen y características técnicas.
- Permitir buscar información sobre productos mediante filtros como rango de precios, marcas y palabras clave, todo desde una sola vista simplificada.
- Permitir compartir el carrito de compra o lista de deseos
- Registrar clientes en el sistema para que puedan realizar compras, ver su historial de pedidos y gestionar su información personal.
- Dar de baja clientes del sistema si lo solicitan, eliminando su cuenta de forma segura mediante la base de datos.
- Enviar información a los clientes periódicamente mediante correos electrónicos sobre novedades, promociones y lanzamientos de nuevos productos.

1.3. Tecnologías y herramientas empleadas

A continuación se presenta un resumen de las herramientas y componentes software utilizados para este proyecto:

Ejemplo:

- Sistema operativo: GNU/Linux.
- Servidor Web: Express.js.
- Gestor de bases de datos: Sqlite3 & Db Browser.
- Lenguajes de programación: Node.js (JavaScript) ,HTML y CSS.
- Formato de intercambio de datos: JSON
- Gestión del proyecto: Dia.
- Documentación del proyecto: Google Drive Docx.
- Herramienta de control de versiones Git
- Frameworks: Express.js (para la gestión del servidor y rutas en Node.js)
- VS Code, Npm.

1.4. Planificación temporal

El coste temporal del proyecto es de 34 horas por hombre. El proyecto debe finalizar el 05 de 05 de 2025, para lo cual se proporcionarán los recursos necesarios para cumplir los plazos de entrega.

El siguiente diagrama de Gantt muestra el reparto de tiempo entre las diferentes fases y tareas del proyecto.

Ejemplo:

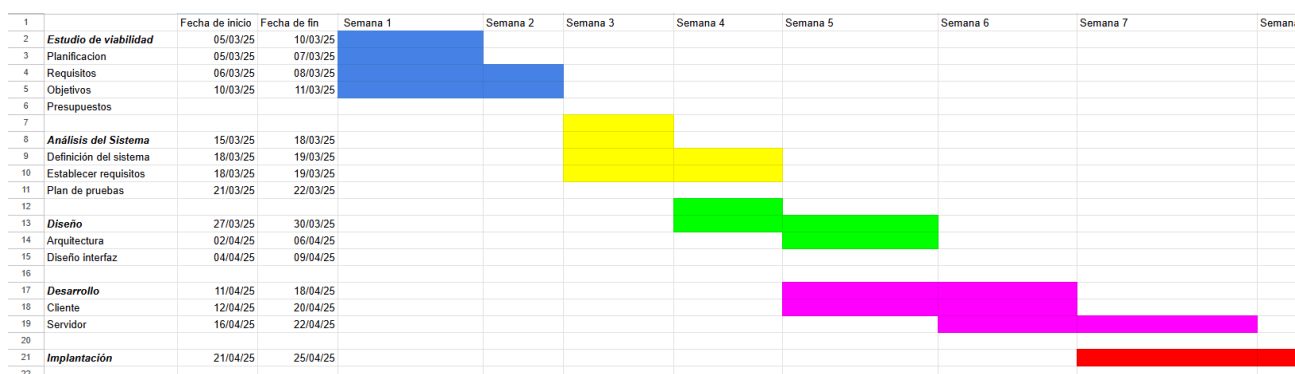


Figura. Planificación temporal del proyecto (Diagrama de Gantt)

En el diagrama destacan dos puntos de control (hitos):

- **El sistema es viable.** Este punto de control se hace después de realizar el estudio de viabilidad. Si el proyecto es viable con los requisitos, objetivos y presupuesto se continúa con el mismo. Si por el contrario el proyecto se considera no viable, hay que cambiarlo o desestimar.

Tras realizar el estudio de viabilidad, el sistema se considera viable.

- **La aceptación del sistema.** Al finalizar la fase de implantación se presenta el proyecto para su aprobación. Si el proyecto es aprobado, entra en producción y comienza su mantenimiento.

Tras presentar el proyecto para su aprobación, se considera aprobado.

Aunque en el diagrama de Gantt cada tarea tiene una fecha de comienzo y finalización concretas, esto no impide que se pueda flexibilizar el comienzo/finalización de las mismas. Se puede, por ejemplo, comenzar con el desarrollo una vez se conozcan los requisitos, para tener un prototipo cuanto antes.

En la siguiente tabla se especifican las fechas de comienzo, finalización y coste temporal de cada una de las fases del proyecto.

Fase	Comienzo	Final	Horas/Hombre
Estudio de viabilidad	05/03/2025	11/03/2025	5
Análisis	15/03/2025	22/03/2025	6
Diseño	27/03/2025	09/04/2025	7
Desarrollo	11/04/2025	18/04/2025	7
Implantación	21/04/2025	25/04/2025	9
Total			34

Tabla. Planificación temporal y coste del proyecto por fases

2. Estudio de viabilidad

Este proyecto es totalmente viable tanto a nivel técnico como económico y operativo, y con una buena organización del tiempo, se puede llevar a cabo de forma efectiva y con éxito.

2.1. Establecimiento del alcance del sistema

Ejemplo:

El proyecto se ha desarrollado utilizando tecnologías y recursos ampliamente disponibles y gratuitos, por lo que el coste total no resulta elevado. Al hacer uso de herramientas como Node.js, MySQL y frameworks libres, se evita la necesidad de adquirir licencias comerciales, lo que supone una ventaja tanto económica como de flexibilidad.

2.2. Estudio de la situación actual

Este proyecto se plantea como un sistema completamente nuevo, por lo que no se parte de ninguna solución previa ni se introducen mejoras sobre una plataforma ya existente. La tienda online será desarrollada desde cero, definiendo su estructura, diseño y funcionalidades conforme a los requisitos establecidos para cubrir las necesidades de venta de productos de informática y gaming.

2.2.1. Nuestra aplicación (Siglas de tu aplicación)

TecnoCraft es la aplicación web propuesta para la creación de una tienda online especializada en la venta de productos de informática y gaming. Aunque no se trata de una mejora o ampliación de un sistema existente, nace con el objetivo de cubrir una necesidad real: disponer de una plataforma digital moderna, accesible y funcional para facilitar la compra y gestión de artículos tecnológicos.

2.3. Definición de los requisitos del sistema

Se realiza una descripción de los requisitos del sistema teniendo en cuenta las necesidades del cliente y las restricciones temporales y de presupuesto.

Esta definición puede realizarse teniendo en cuenta diferentes requisitos por prioridad del 1 al 5 (siendo 5 la prioridad más alta).

- **Prioridad 5:** Acceso web desde cualquier dispositivo, base de datos relacional, funciones clave como catálogo de productos, registro de usuarios y carrito de compra.
- **Prioridad 4:** Diseño responsivo, accesibilidad básica, filtros de búsqueda, gestión de productos por el administrador.

- **Prioridad 3:** Uso de estándares abiertos, separación cliente-servidor, documentación con herramientas libres.
- **Prioridad 2:** Preparado para futuras integraciones (como pagos), despliegue multiplataforma, compartir productos.
- **Prioridad 1:** Valoraciones de productos, mejoras opcionales futuras.

2.4. Estudio de la solución adoptada

Se opta por una solución desarrollada a medida y libre por las características del proyecto, al no existir en el mercado una aplicación que cumpla con los requisitos del cliente.

En la siguiente tabla se describen los detalles del software necesario:

Aplicación	Notas
Aplicación Web	Desarrollada a medida utilizando PHP, JavaScript, HTML5, CSS3 y VS
GNU Linux	GNU/Linux (usado en el servidor para alojar la aplicación).
Apache	Servidor web
MySQL	Sistema de Gestión de Bases de datos
Frameworks y librerías	Express.js (framework de Node.js para gestionar rutas y middleware)
Control de versiones	Git y GitHub (para controlar el código fuente del proyecto y facilitar el trabajo colaborativo mediante el repositorio).

2.5. Presupuesto

El presupuesto disponible para este proyecto es de xxx euros, los cuales se repartirán del siguiente modo:

- **Compra de equipos (xx €).** Incluye la adquisición de servidores, estaciones de trabajo, etc.
- **Implantación del proyecto (xx €).** Incluye el coste necesario para la realización de las diferentes fases del proyecto.

- **Formación (xx €).** Formación a los responsables técnicos del mantenimiento del proyecto.
- **Servicios (xx €).** Gastos de alquiler del cuarto de servidores, gastos de electricidad, conexión a Internet, etc. ...

Se considera que el coste de trabajo por hora/hombre es de xx € de media.

En la siguiente tabla se detalla cada uno de los gastos y su valor:

Concepto	Cantidad	Precio Unitario (€)	Total (€)
Equipos	1	700 €	700 €
			...
			...
TOTAL Equipos			700 €
Implantación del proyecto			
Estudio de viabilidad	5 h.	10€/h.	50 €
Análisis	10 h.	10 €/h.	100 €
Diseño	7 h.	15€/h.	105 €
Desarrollo	11 h.	10€/h.	110 €
Implantación	10 h.	20€/h.	200 €
Formación	6 h.	25€/h.	150 €
TOTAL Implantación			715 €
Servicios			
Alquiler de hosting	1	120 €/año	120 €
Teléfono, ADSL, electricidad, agua.	1	50€/año	50 €
TOTAL Servicios	2		170 €
TOTAL			885 €

2.6. Análisis de riesgos

En todo proyecto, tanto si es uno que se desarrolla desde cero, como de uno existente en el que se quiere mejorar/ampliar algún aspecto, existen riesgos.

Los riesgos se clasifican normalmente en dos tipos:

- **Externos:** los originados por factores externos a la empresa, por ejemplo, cambio de políticas o legislación que deje a nuestro proyecto en un estado inestable.
- **Internos:** los que se crean en la propia empresa (técnicos, de gestión, de falta de infraestructura).

Además, cuando el proyecto se desarrolla mediante software libre hay que tener en cuenta los riesgos añadidos en este tipo de proyectos.

A continuación se describen los riesgos que pueden aparecer en el proyecto, su probabilidad de ocurrencia y las acciones necesarias o que pueden realizarse para evitarlos, corregirlos o mitigarlos.

Tipo	Riesgos	Probabilidad	Acciones de prevención y remedio
Externo	Cambios en las políticas de gestión de la información	Baja	Mantenerse actualizado con las normativas y políticas de protección de datos. Adaptarse rápidamente a cambios legislativos.

Interno	Complejidad de mantenimiento y configuración	Baja	Implementar una solución no compleja. Proporcionar suficiente documentación y ofrecer cursos de formación al personal técnico.
	Falta de soporte de la comunidad del software libre	Baja	Escoger soluciones libres populares que garanticen su permanencia y ampliamente

			conocidas para, dado el caso, poder contratar el soporte.
	Que el proyecto no tenga popularidad y su página Web no sea visitada.	Media	Realizar una aplicación Web fácil de usar y atractiva.

Riesgos	Acciones de prevención y remedio
No disponer de un encargado del mantenimiento de un determinado producto de Software Libre.	Al disponer del código fuente se pueden contratar servicios de mantenimiento.
Reticencia a adoptar soluciones libres por parte de empresas y usuarios.	Proporcionar una visión clara de las soluciones. Proporcionar suficiente formación y dar una solución preconfigurada, que requiera el mínimo mantenimiento.

3. Análisis del sistema

En esta fase se realiza un análisis detallado de la solución que se ha adoptado para la aplicación elegida.

3.1. Definición del sistema

El sistema desarrollado es una tienda online orientada a la venta de productos de informática y gaming, que permite a los usuarios explorar, buscar, comparar y adquirir productos de manera sencilla y rápida a través de una plataforma web.

La arquitectura del sistema está basada en una estructura de aplicación web dinámica, utilizando Node.js para el backend y Express.js como framework para la gestión de rutas y solicitudes HTTP. El frontend está desarrollado con HTML, CSS, y JavaScript, asegurando una interfaz interactiva y responsiva para adaptarse a diferentes tamaños de pantalla.

A continuación se definen los diferentes perfiles de usuarios que pueden interactuar con la aplicación (Siglas o nombre de tu aplicación):

- **Usuario.** Este accede a la parte de Puede acceder desde cualquier navegador con/sin necesidad de autenticarse.

- **Administrador.** Este perfil es el del encargado del mantenimiento del sistema. Esto incluye la ejecución de scripts, realizar actualizaciones, analizar los ficheros “log” y hacer las copias de seguridad. El administrador puede usar las interfaces propias del sistema operativo y de las aplicaciones como, por ejemplo, MySQL o “phpMyAdmin” para Apache. Podrá acceder desde cualquier navegador, pero necesita autenticación.

3.2. Establecimiento de requisitos

A continuación se realiza un refinamiento de los requisitos definidos previamente durante el estudio de viabilidad del sistema. Para ello se divide la aplicación en varios subsistemas.

Gestión de Productos: Permite añadir, editar, eliminar y visualizar productos, gestionando detalles como precio, descripción e inventario.

Gestión de Usuarios: Facilita el registro, autenticación, y administración de cuentas de usuario, así como la gestión de su historial de pedidos.

Carrito de Compras: Permite a los usuarios añadir y eliminar productos, calcular el total y proceder al pago.

Gestión de Pedidos: Administra los pedidos realizados, su estado y la información de envío.

Pago y Facturación: Integra pasarelas de pago para procesar pagos de forma segura y generar facturas electrónicas.

Seguridad y Protección de Datos: Implementa cifrado de contraseñas, HTTPS, y otras medidas de seguridad para proteger los datos de los usuarios.

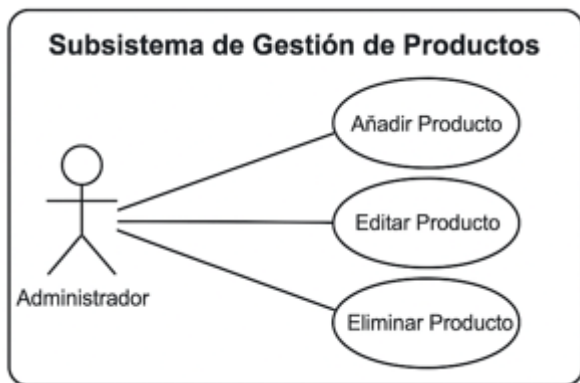
Marketing y Promociones: Gestiona códigos de descuento y promociones, y permite el envío de correos electrónicos con ofertas y novedades.

Administración: Ofrece un panel para la gestión de productos, usuarios, pedidos y la visualización de reportes.

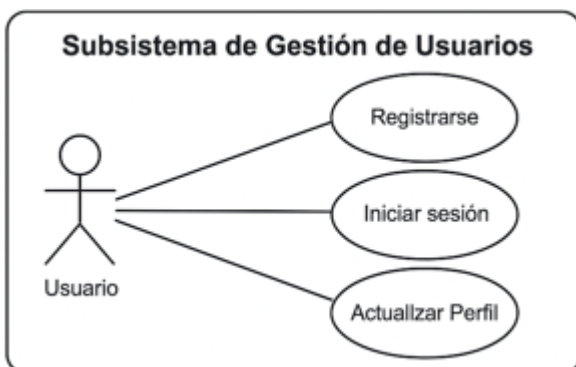
Herramientas para crear diagramas UML (podéis usar cualquier otra):

- YUML (online): <https://yuml.me/> (online)
- draw.io: <https://www.draw.io/> (online)
- ArgoUML: <http://argouml.tigris.org/> (descarga)
- startUML: <http://staruml.io/download> (descarga)

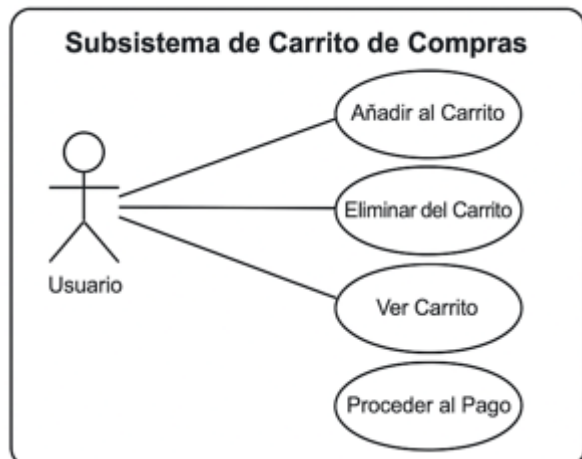
3.2.1. Subsistema 1



3.2.2. Subsistema 2



3.2.3. Subsistema n



3.3. Definición de interfaces de usuario

La aplicación web está diseñada para ser clara, intuitiva y visualmente atractiva, adaptada tanto a usuarios con conocimientos básicos como a los más avanzados.

Las principales interfaces son las siguientes:

- **Página de inicio:** muestra un resumen del catálogo de productos destacados, con botones visibles para acceder al carrito, iniciar sesión o registrarse.
- **Página de productos:** lista de artículos con imágenes, nombre, precio y botón de "Añadir al carrito". Se incluye un buscador y filtros por precio o marca (aunque no se usa el sistema de categorías).
- **Carrito de compra:** permite al usuario ver los productos añadidos, modificar cantidades o eliminar productos antes de confirmar la compra.

- **Atención al cliente:** se incluye una sección de **contacto** con un formulario para que los usuarios puedan realizar consultas o solicitar soporte técnico. También se facilita un correo electrónico y se contempla incluir un pequeño sistema de mensajes o respuestas automáticas en el futuro.

3.3.1 Página principal



3.3.2. Página de autenticación

Se accede desde la página principal (o será la página principal) para solicitar un nombre de usuario y contraseña.

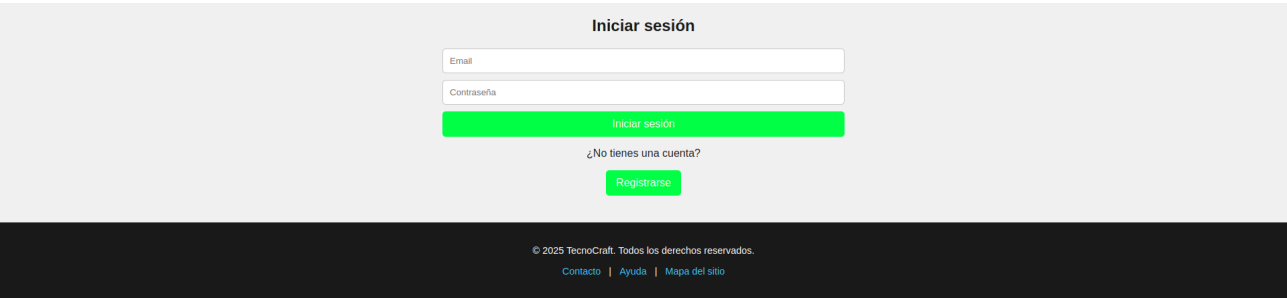
Consta de:

- Cuadro de texto **Nombre**: solicita el nombre de usuario.
- Cuadro de texto **Contraseña**: solicita la contraseña del usuario.
- Botón **Registrarse**: inicia la acción de Registrarse en el sistema.

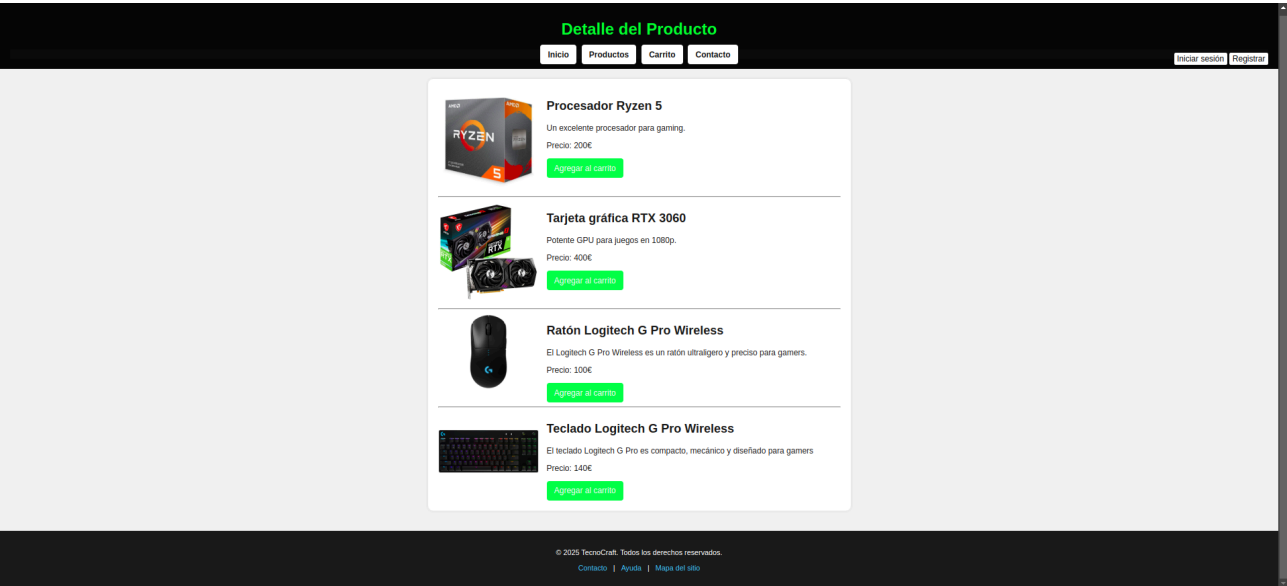


The screenshot displays a registration form titled "Registro" centered on a light gray background. The form consists of four stacked text input fields with labels "Nombre", "Apellidos", "Email", and "Contraseña" positioned to the left of each field. Below these fields is a prominent green button labeled "Registrarse". At the bottom of the page, a dark gray footer contains the copyright notice "© 2025 TecnoCraft. Todos los derechos reservados." and three links: "Contacto", "Ayuda", and "Mapa del sitio".

3.3.3. Página de inicio de sesión



3.3.4. Página de productos



3.3.5. Página de carrito



3.3.6. Página de contacto

Contacto

Inicio Productos Carrito Contacto Manel Cerrar sesión

Contáctanos

Si tienes alguna duda, por favor rellena el siguiente formulario y nos pondremos en contacto contigo.

Nombre completo:

Manel Albella

Correo Electrónico:

manelalbella@gmail.com

Mensaje:

Enviar

© 2025 TecnoCraft. Todos los derechos reservados.
[Contacto](#) | [Ayuda](#) | [Mapa del sitio](#)

3.3.4. Página de ayuda

Muestra una pantalla con texto de ayuda sobre las funcionalidades del sistema.

¿Qué métodos de pago aceptan?

Aceptamos tarjetas de crédito y débito (Visa, Mastercard), PayPal y transferencias bancarias.

¿Cuánto tarda en llegar mi pedido?

El tiempo estimado de entrega es de 24 a 72 horas laborables, dependiendo de la zona de envío.

¿Puedo devolver un producto si no estoy satisfecho?

Sí, puedes solicitar una devolución en un plazo de 14 días desde la recepción del pedido. Consulta nuestra política de devoluciones para más detalles.

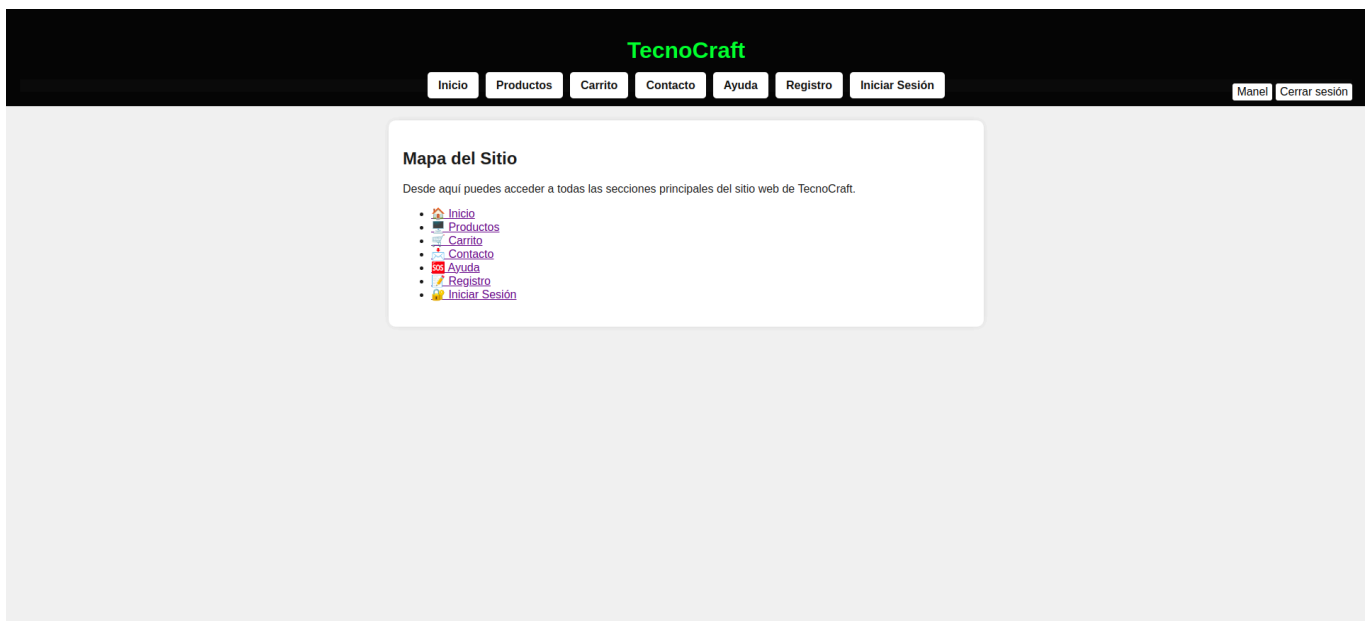
¿Cómo puedo seguir el estado de mi pedido?

Una vez confirmado el pedido, recibirás un correo con un enlace para hacer el seguimiento del envío en tiempo real.

¿Puedo pedir productos que no estén en la web?

Sí, puedes contactarnos a través del formulario de soporte o el chat para consultar disponibilidad de productos específicos.

3.4. Mapa de navegación



Herramientas para crear prototipos, mapas de navegación, etc. (podéis usar cualquier otra):

- FrameBox(online): <http://framebox.org/>
- Wireframe cc (online): <https://wireframe.cc/>
- Pencil (descarga): <http://pencil.evolus.vn/>
- Bubbl.us (mapas de navegación). <https://bubbl.us/>
- Cmap tools (descarga): <http://cmap.ihmc.us/>

4. Diseño del sistema

En este apartado se define “cómo” va a cumplir la aplicación los requisitos definidos previamente. Se describirán los componentes necesarios para cumplir esos requisitos y la

forma que tienen de interactuar (comunicarse) con otros componentes de forma que se obtenga un modelo de arquitectura.

El conjunto de estos componentes deben de satisfacer los casos de uso definidos anteriormente.

4.1. Arquitectura

Se define el sistema en bloques generales para ir poco a poco desglosándolos en subsistemas, componentes y módulos.

Ejemplo de gráfico que describe la arquitectura de un sistema de tres capas:

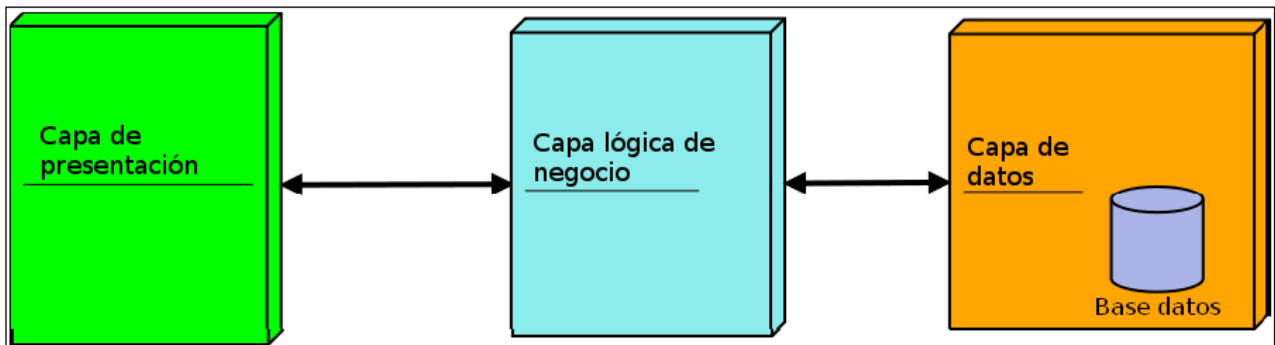


Imagen. Arquitectura de un sistema de tres capas

Como se muestra en la imagen anterior, el sistema tiene una arquitectura de tres capas, en la que cada una de las capas tiene una misión perfectamente delimitada e independiente de las demás.

- **Capa de Presentación:** Es la capa que interactúa con el usuario, en la cual puede introducir los datos y, a su vez, los datos del sistema son presentados en esta capa. En el caso de una aplicación Web, como es este caso, esta capa proporciona los contenidos de las páginas web a la que el usuario accede. Tendrá la página de presentación, autenticación, ayuda, etc. Normalmente se ejecuta en la parte cliente.
- **Lógica de negocio (Business):** Es la capa encargada de realizar la computación, es decir los diferentes procesos en la aplicación. Aquí encontraremos las funciones que toman decisiones para que el usuario pueda visualizar los datos que se necesitan y sobre los que introduce en el sistema. Sirve como puente entre las capas de presentación y de datos para mover los datos entre ellas, y de interfaz entre la base de datos y la presentación de los mismos al usuario.

- **Capa de Datos:** Es la capa en la que residen y se manipulan los datos. Todos los accesos a la base de datos se encuentran en esta capa.

4.1.1. Identificación de subsistemas

Para la división en subsistemas se consideran factores como:

- Funcionalidad.
- Datos y tipos a los que acceden.
- Interfaz de usuario común.

A continuación se incluye un diagrama de bloques donde se muestran los distintos subsistemas y cómo se comunican.

Ejemplo:

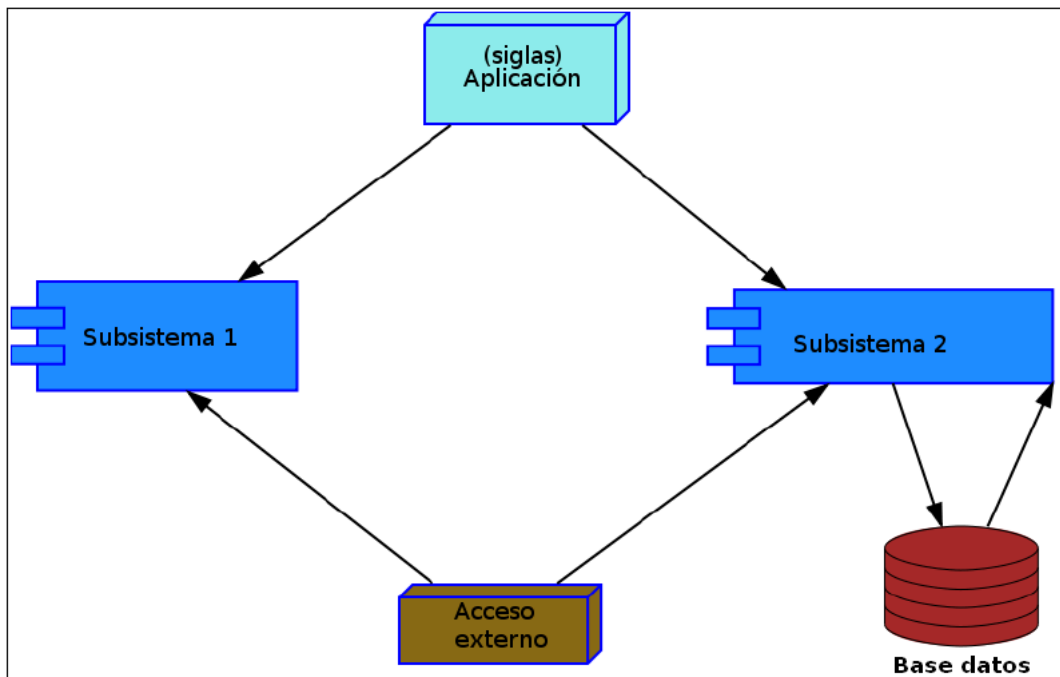


Imagen. Diagrama de bloques de los distintos subsistemas

Una vez definidos los subsistemas de la aplicación, se pasa a definir los componentes de cada uno de ellos en mayor detalle.

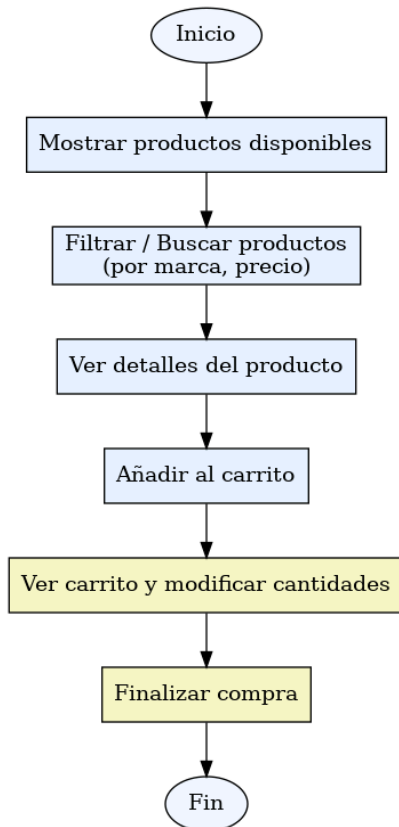
4.1.2. Subsistema 1

En este punto, se puede realizar un diagrama donde se aprecian los componentes o módulos que comprenden este subsistema. Se describe también la comunicación entre ellos y con los sistemas remotos (si es el caso), así como el tipo de interfaz que se usa.

A continuación, podéis detallar cada uno de los componentes.

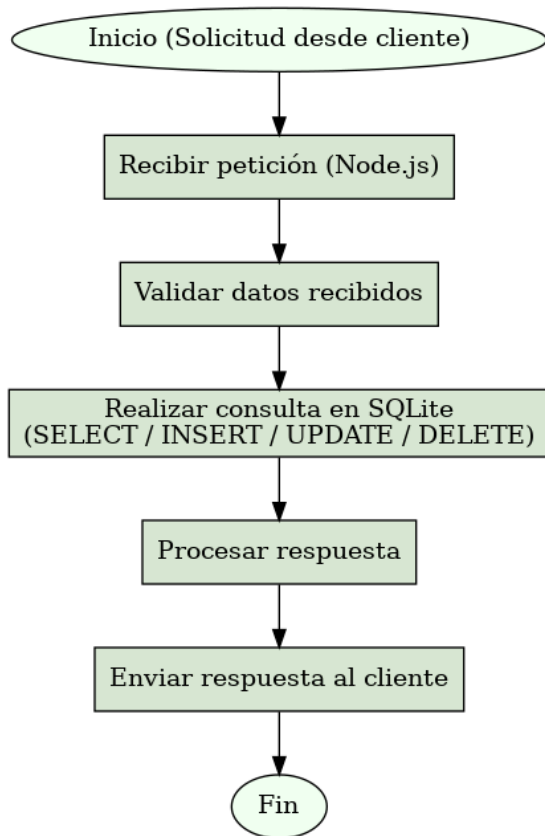
Ejemplo:

- **Módulo Principal:** Es el que proporciona la información al cliente que se conecta a la Web. Está escrito en JavaScript, HTML5, CSS3, etc. A continuación, podéis aportar un diagrama de flujo de este módulo. Como ejemplo se adjunta la siguiente figura:

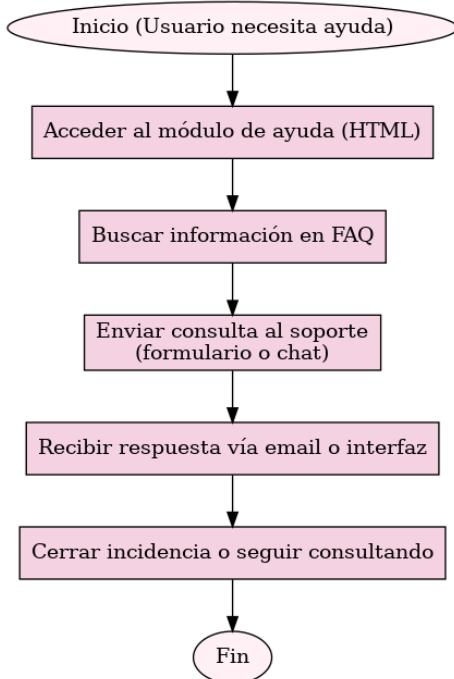


- **Módulo de Ayuda:** Es el encargado de presentar la información sobre el uso y funciones de la aplicación a los usuarios que lo soliciten. Está escrito en HTML5, etc.

4.1.3. Subsistema 2



4.1.4. Subsistema n



4.2. Diseño de la base de datos

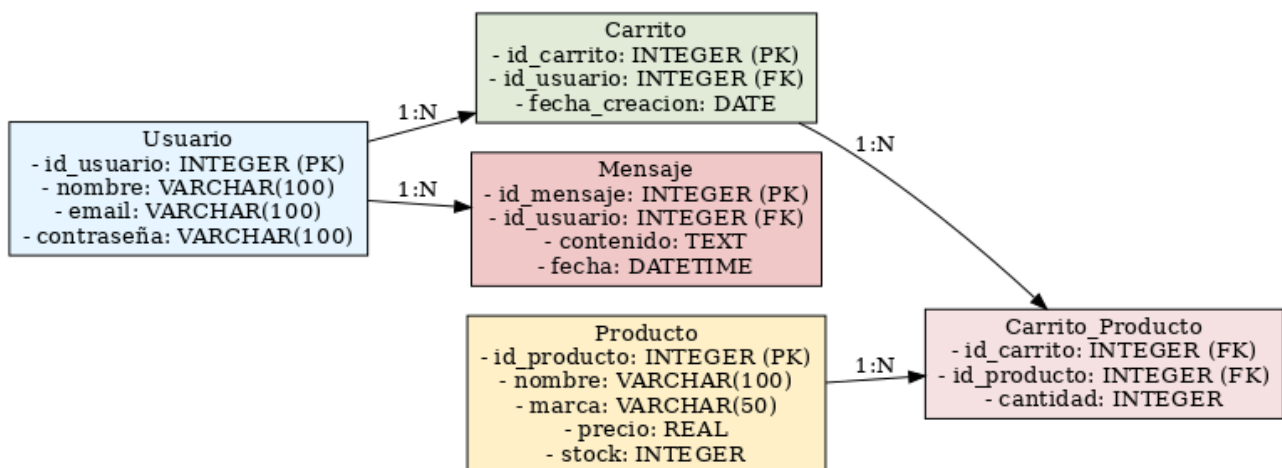
Para la base de datos se ha optado por un modelo relacional. A continuación, se adjunta el modelo Entidad-Relación o esquema relacional de la misma.

Ejemplo:

Figura. Esquema relacional de la base de datos

4.3. Elección de alternativas de componentes y licencias más adecuadas

“[En este apartado se define el software que se utiliza para cada componente del sistema y su licencia.]”



Para cada componente se ha usado software libre con una licencia que no permita incompatibilidades entre ellas.

La siguiente tabla describe en detalle el software empleado en cada parte del sistema así como su licencia. Esta combinación nos permite que la aplicación final (Siglas o nombre de la aplicación) se licencie como tipo GPL.

Componente	Software	Licencia
Aplicación Web	Lenguajes: JavaScript, HTML5, CSS	MIT / Estándares
Backend del servidor	Node.js + Express	MIT
Base de datos	SQLite 3	Public Domain
Sistema operativo	GNU/Linux Ubuntu	GPL
Visualización base de datos	DB Browser for SQLite	GPL
Servidor Web	Express.js	MIT
Librerías adicionales	dotenv, body-parser, JWT	MIT
Herramienta de diagramas	Dia	GPL
Planificación	Gantt Project	GPL
Suite ofimática	LibreOffice	GPL
Editor de código	Visual Studio Code	MIT
Control de versiones	Git	GPL

Los documentos, manuales y los materiales de formación se licenciará como Reconocimiento-Compartir Igual de Creative Commons.

Para el código fuente se usará la licencia GPL.

4.4. Especificaciones de desarrollo

En este apartado se describen las necesidades tanto de equipos como de software necesarios para realizar el desarrollo del proyecto.

4.4.1. Necesidades de equipos

Equipos clientes

Los equipos clientes necesitan disponer de las siguientes características mínimas para poder utilizar correctamente la aplicación web:

- **Procesador:** Intel Core i3 o equivalente
- **Memoria RAM:** 4 GB
- **Almacenamiento:** 1 GB libre (para caché del navegador y archivos temporales)
- **Navegador web actualizado:** Google Chrome, Firefox, Edge o similar
- **Conexión a Internet estable**
- **Resolución de pantalla:** 1366x768 o superior

Equipo(s) servidor(es).

El equipo servidor utilizado para alojar la aplicación y gestionar las peticiones necesita unas características algo más avanzadas para garantizar estabilidad y buen rendimiento:

- **Procesador:** Intel Core i5 o superior (cuatro núcleos)
- **Memoria RAM:** 8 GB mínimo
- **Almacenamiento:** 50 GB disponibles

- **Sistema operativo:** Linux Ubuntu 20.04 o superior
- **Software necesario:** Node.js, SQLite3, Express.js, Git
- **Conexión a red:** Acceso permanente a Internet (preferiblemente por cable)

4.4.2. Necesidades software

Se incluyen tanto las utilidades necesarias para el desarrollo en sí, como las aplicaciones que soportan el sistema y las aplicaciones para generar la documentación.

Utilidades de desarrollo

- **Sistema operativo:** Linux Ubuntu 20.04 (u otro compatible)
- **Editor de código:** Visual Studio Code
- **Node.js:** v 18.x
- **SQLite3:** como sistema de gestión de base de datos
- **DB Browser for SQLite:** para la visualización y gestión gráfica de la base de datos
- **Git:** para el control de versiones

Aplicaciones usadas en el sistema

- **Servidor local:** Express.js (ejecutado desde Node.js)
- **Librerías adicionales:** JSON Web Token, dotenv, body-parser, etc.

Aplicaciones para la documentación

- **Procesador de textos:** LibreOffice Writer
- **Diagramas:** Dia
- **Planificación:** Gantt Project

4.5. Especificación del plan de pruebas

En este apartado se especifica un plan de pruebas que permita la detección temprana de errores.

Empezando por cada componente de manera individual y progresivamente, se irá probando la integración con otros componentes. Para finalizar, se planificarán las pruebas que deben hacer los usuarios para validar y aceptar el sistema.

Durante el desarrollo se realizaron pruebas unitarias para verificar el correcto funcionamiento de los distintos componentes de forma individual. Algunas de las pruebas más destacadas fueron:

- **Páginas web:** Se comprobó que cada vista mostraba correctamente la información según la acción del usuario (por ejemplo, al añadir productos al carrito o al registrarse).
- **Formularios:** Se probó que los formularios de registro, login y contacto validaron correctamente los datos introducidos y mostrarán los mensajes adecuados en caso de error.
- **Funciones del servidor:** Se testean las rutas del servidor (por ejemplo, obtener productos o añadir al carrito) para comprobar que devuelven los datos esperados.
- **Base de datos:** Se verificó que las operaciones CRUD (crear, leer, actualizar, eliminar) funcionaran correctamente en la base de datos SQLite.

4.5.2. Pruebas de Implantación

Una vez finalizado el desarrollo, se llevaron a cabo diferentes pruebas para asegurar que la aplicación funciona correctamente en su entorno de producción. Las principales pruebas realizadas fueron:

- **Acceso desde distintos navegadores y dispositivos** para verificar la compatibilidad y el correcto funcionamiento del sitio web tanto en escritorio como en móviles.
- **Pruebas de carga** simulando múltiples usuarios realizando búsquedas y añadiendo productos al carrito al mismo tiempo, para comprobar el rendimiento del servidor y la base de datos SQLite.
- **Verificación del sistema de login y autenticación** con JWT para asegurar que solo los usuarios autorizados pueden acceder a determinadas funcionalidades.

Estas pruebas permitieron validar el correcto funcionamiento de la aplicación y detectar pequeños fallos que fueron corregidos

5. Desarrollo

5.1. Desarrollo en el lado del cliente

El desarrollo del lado del cliente se ha centrado en crear una interfaz clara, fácil de usar y visualmente atractiva. Se ha trabajado con HTML, CSS y JavaScript para que la web sea responsive y funcione bien en cualquier dispositivo. Uno de los retos fue hacer que el carrito y los filtros funcionarán sin recargar la página, pero se resolvió usando JavaScript y localStorage.

5.2. Desarrollo en el lado del servidor

El backend se desarrolló con Node.js y Express, gestionando las rutas, usuarios, productos y pedidos.

Se utilizó SQLite3 como base de datos por su facilidad de uso en proyectos pequeños, y se visualiza gráficamente con DB Browser for SQLite para facilitar la gestión de tablas y datos.

Para la autenticación se usó JWT, y se estructuró el código en módulos para hacerlo más limpio y mantenible. También se añadió un sistema de manejo de errores centralizado.

6. Conclusiones

Este proyecto ha sido una experiencia muy útil para poner en práctica lo aprendido sobre desarrollo web. Me ha servido para entender mejor cómo se conectan el frontend y el backend, y cómo organizar una aplicación real desde cero. Aunque al principio surgieron algunos problemas, poco a poco fui encontrando soluciones buscando información y mejorando el resultado final.

Me ha resultado interesante trabajar con Node.js y SQLite3, y ver cómo con herramientas como DB Browser se puede gestionar fácilmente la base de datos.

Teniendo en cuenta que he intentado hacer la “competencia” a la empresa tan conocida en España llamada Pc Componentes.

6.1. Nivel de satisfacción general

En general, estoy satisfecho con el resultado del proyecto. He podido cumplir con los objetivos principales y la aplicación funciona de forma estable.

Ha sido un proceso en el que he aprendido mucho, sobre todo al enfrentarme a errores y buscar soluciones por mi cuenta.

Aunque siempre se pueden mejorar cosas, creo que el resultado final es bueno

6.2. Propuestas de mejoras

Aunque el proyecto funciona correctamente, hay varias mejoras que podrían aplicarse en el futuro:

Diseño más avanzado: Implementar un diseño más profesional utilizando frameworks como Bootstrap o Tailwind CSS.

Sistema de pago real: Integrar una pasarela de pago como PayPal.

Sistema de reseñas y valoraciones: Permitir que los usuarios puedan dejar opiniones y puntuaciones sobre los productos.

Multiidioma: Incluir soporte para varios idiomas, por ejemplo español e inglés.

Favoritos o wishlist: Permitir a los usuarios guardar productos que les interesen para futuras compras.

7. Bibliografía

- Node.js Oficial – <https://nodejs.org/>

- Express.js – <https://expressjs.com/>
- SQLite – <https://sqlite.org/>
- DB Browser for SQLite – <https://sqlitebrowser.org/>
- W3Schools: HTML, CSS, JavaScript – <https://www.w3schools.com/>