SISTEMAS DE LA INFORMACIÓN
3º INGENIERÍA INFORMÁTICA

846088 ABEL ROMEO LANCINA
841723 DIEGO ROLDÁN URUEÑA
841972 PABLO MORENO MUÑOZ

ÍNDICE

Introducción

El presente informe detalla el desarrollo de un proyecto para la cuarta práctica de la asignatura Sistemas de la Información, de 3º de Ingeniería Informática, enfocado en la creación de una plataforma web robusta y funcional, utilizando tecnologías clave como Node.js, Express, HTML y CSS. Este proyecto tiene como objetivo principal proporcionar una interfaz eficiente y atractiva para la gestión de usuarios, pedidos y productos, con la capacidad de persistir los cambios en una base de datos.

En un entorno digital cada vez más dinámico, la necesidad de soluciones web eficientes y escalables es fundamental. Este proyecto surge como respuesta a la demanda de una plataforma versátil que permita la gestión de usuarios, pedidos y productos de manera centralizada y accesible.

Los objetivos principales que guiaron el desarrollo de este proyecto han sido:

Desarrollo de una interfaz intuitiva y fácil de usar, que permita a los usuarios realizar operaciones de manera sencilla y eficiente.

Implementación de persistencia de datos para construir un backend sólido que se conecte a una base de datos, garantizando la integridad de los datos de usuarios, pedidos y productos.

Aseguración de la escalabilidad, de manera que pueda ser modificable fácilmente para manejar un crecimiento futuro en la cantidad de usuarios y datos.

Se ha desarrollado un plan de pruebas detallado para asegurar la estabilidad y funcionalidad de la aplicación antes de su implementación.

Establecimiento de un plan de mantenimiento para prever y planificar acciones que aseguren el correcto funcionamiento continuo de la aplicación una vez en un entorno de producción.

Para alcanzar estos objetivos, se han seleccionado cuidadosamente diversas tecnologías que facilitan su cumplimiento.

A lo largo de este informe, se detallan los aspectos clave del diseño, la metodología de trabajo, las dificultades encontradas y los planes de prueba y mantenimiento, proporcionando una visión integral del proceso de desarrollo de esta plataforma web.

Informe del Proyecto: Página Web

Diseños Realizados

Se ha diseñado una página web con las siguientes características:

- **Página de Inicio**:

Presenta una descripción general de la plataforma y acceso rápido a las funciones principales.

Encabezado:

Sección de contacto con número telefónico, correo electrónico y dirección.

Área de administrador con un ícono de candado para funciones administrativas.

Menú de navegación con categorías como Home, Hot Deals, Procesadores, Placas Base.

Contenido Principal:

Secciones organizadas por categorías, cada una con una imagen representativa y enlace "Shop now".

Barra de búsqueda para facilitar la exploración de productos.

Secciones Destacadas:

Hot Deal Section: Ofertas destacadas con temporizador de cuenta regresiva.

Nuevos Productos: Presentación de nuevos productos con pestañas para diferentes categorías.

Top Selling: Carrusel que destaca los productos más vendidos.

Pie de Página:

Información adicional sobre la tienda, incluyendo ubicación, categorías, enlaces y métodos de contacto.

Enlaces a redes sociales y logotipos de métodos de pago aceptados.

Funcionalidades Adicionales:

Scripts que gestionan la visibilidad de elementos según el rol del usuario.

Integración de bibliotecas como Bootstrap, FontAwesome y Google Fonts.

- **Registro de Usuarios**:

Formulario de registro con validación de campos y almacenamiento en la base de datos.

Secciones Principales:

Formularios interactivos para Login, Signup y Recuperación de Contraseña.

Diseño dividido con transiciones entre las secciones de Login, Signup y Recuperación.

Funcionalidades Adicionales:

Iconos interactivos para mostrar/ocultar contraseñas.

Scripts JavaScript para gestionar transiciones entre formularios y operaciones de usuario (login, signup, recuperación de contraseña).

- **Datos de Usuarios**:

Sección de los datos:
Menú de opciones: "Mis Datos" y "Mis Pedidos".
Sección de Datos Personales:
Formulario para actualizar la información del usuario (correo electrónico, nombre, apellidos, dirección, contraseña).
Icono interactivo para mostrar/ocultar la contraseña.
Botones para guardar cambios y cerrar sesión.
Sección de Pedidos:
Lista desplegable de pedidos en proceso, enviados y entregados.
Los pedidos se presentan en listas ordenadas.
Scripts JavaScript:
Funciones asociadas a los botones y eventos del formulario.
Manejo dinámico de la visibilidad de las secciones "Datos Personales" y "Pedidos".
Operaciones de actualización de datos de usuario y cierre de sesión.

- Catálogo de Productos:

Barra de Navegación:
Menú de navegación con categorías de productos.
Barra de búsqueda con opciones desplegables (categorías).

Contenido Principal:
Breadcrumb que indica la ubicación actual del usuario en el sitio.
Filtros laterales por precio y marca.
Selección de cantidad de productos a mostrar y opciones de ordenamiento.

Productos:
Presentación dinámica de productos según los filtros aplicados.
Interfaz interactiva con imágenes y detalles de productos.

Scripts JavaScript:

- Función crearTarjeta(item):

Crea la estructura HTML de una tarjeta de producto.
Maneja visualmente la información del producto, incluyendo descuentos
Genera aleatoriamente una calificación de estrellas.

- Función foto(tipo):

Retorna la URL de la imagen según el tipo de producto.

- Función hot_deals():

Captura las selecciones del usuario (cantidad, orden, precios, marcas).
Llama a search_products con los parámetros adecuados.

- Función search_products:

Construye la URL de búsqueda codificando los parámetros.
Realiza una solicitud al servidor y procesa la respuesta.
Crea y agrega dinámicamente tarjetas de productos al contenedor.
Maneja casos especiales, como tarjetas vacías para mantener la cuadrícula.

Metodología de Trabajo

Planificación

Se ha llevado a cabo una fase inicial de planificación donde se identificaron y documentaron de manera detallada los requisitos del proyecto.
Se ha creado un plan de desarrollo que incluía una lista exhaustiva de funcionalidades a implementar, priorizando las tareas según su importancia y dependencias.
Se han establecido plazos para garantizar un progreso constante y cumplir con los requisitos establecidos.

Desarrollo Iterativo

Se ha adoptado un enfoque iterativo, dividiendo el desarrollo en ciclos cortos y manejables.
Durante cada iteración, se han implementado nuevas funcionalidades o se mejoraron las existentes.
Después de cada iteración, se han realizado pruebas exhaustivas para identificar y corregir errores de manera temprana.

Uso de Node.js y Express

Node.js ha sido seleccionado como entorno de ejecución del lado del servidor debido a su eficiencia y capacidad para manejar operaciones de entrada/salida de manera no bloqueante.
Express, un marco de aplicación web para Node.js, se ha utilizado para simplificar y organizar el manejo de rutas, middleware y solicitudes HTTP.
La elección de estas tecnologías permite la construcción de un backend eficiente y escalable.

HTML y CSS

HTML se ha empleado para estructurar el contenido de las páginas web, definiendo la jerarquía y disposición de los elementos.
CSS se ha utilizado para aplicar estilos y diseños atractivos, mejorando la presentación visual de la plataforma.
Se ha seguido una metodología de desarrollo web centrada en la responsividad, asegurando que la aplicación fuera accesible y estéticamente agradable en una variedad de dispositivos.

Manejo de Base de Datos

Se ha implementado una base de datos para almacenar información crítica, como datos de usuarios, pedidos y productos.
La interacción con la base de datos se gestionó mediante DAO (Data Access Objects) y VO (Value Objects), siguiendo patrones de diseño que facilitan la manipulación eficiente de datos.
Se establecieron mecanismos de seguridad, como la validación de datos antes de la inserción en la base de datos, para garantizar la integridad de la información almacenada.

Estos elementos combinados forman una sólida base para el desarrollo de una aplicación web, asegurando no solo la funcionalidad deseada, sino también una estructura de código organizada y mantenible. Además, permiten adaptarse eficientemente a cambios y mejoras continuas en el proyecto.

Recursos y Herramientas Utilizadas

Node.js y Express

Node.js:

Se ha seleccionado Node.js como entorno de ejecución del lado del servidor debido a su naturaleza no bloqueante y su capacidad para manejar múltiples operaciones de manera eficiente. Esto resulta crucial para aplicaciones web que requieren interacciones rápidas y fluidas.

Express

Este marco de aplicación web para Node.js simplificó el desarrollo del backend. Express facilitó la creación de rutas, middleware y el manejo de solicitudes HTTP. Su estructura minimalista y su flexibilidad permiten una implementación rápida y eficiente de la lógica de negocio del servidor.

HTML y CSS

HTML

Ha sido utilizado para estructurar el contenido de las páginas web. Define la jerarquía de los elementos, desde encabezados hasta formularios, proporcionando la base estructural esencial para la presentación de la información.

CSS

Se emplea para estilizar y diseñar la interfaz de usuario. La separación de la estructura (HTML) y los estilos (CSS) permite un desarrollo más modular y facilita futuras modificaciones en el diseño sin afectar la estructura de la página.

Base de Datos

Persistencia de Datos

La base de datos almacena información crítica, incluyendo detalles de usuarios, pedidos y productos. Esto asegura que los datos sean persistentes, permitiendo que la aplicación recuerde y recupere información incluso después de reinicios o cambios en la ejecución del servidor.

DAO y VO

Para interactuar con la base de datos de manera eficiente, se han empleado patrones de diseño como DAO (Data Access Objects) y VO (Value Objects). Estos patrones facilitan la manipulación y el acceso a los datos, proporcionando una capa de abstracción que simplifica las operaciones de base de datos.

Control de Versiones y Colaboración

Git

Es utilizado para el control de versiones, permitiendo el seguimiento de cambios en el código a lo largo del tiempo. Además, Git facilita la colaboración entre miembros del equipo al gestionar de manera eficiente las ramas y fusiones de código.

Entorno de Desarrollo

VS Code

Se ha elegido Visual Studio Code como entorno de desarrollo principal. Su interfaz intuitiva, soporte extensivo para lenguajes de programación, y una amplia variedad de extensiones lo convierten en una herramienta poderosa para el desarrollo de aplicaciones web. La integración nativa con Git también simplifica la gestión del control de versiones directamente desde el entorno de desarrollo.
Estas tecnologías y herramientas se seleccionaron cuidadosamente para maximizar la eficiencia, la colaboración y la mantenibilidad del proyecto, garantizando una base sólida para el desarrollo y evolución continua de la aplicación web.

Distribución del Trabajo

Backend

Desarrollo de Rutas

Se han diseñado las rutas del servidor utilizando Express. Cada ruta corresponde a una funcionalidad específica de la aplicación, como registro de usuarios, gestión de pedidos, o consulta de productos.
Se han establecido métodos HTTP adecuados para cada ruta (GET, POST, DELETE, etc.) según los requisitos del proyecto.

Lógica de Negocio:

La lógica de negocio se implementó en el backend para procesar las solicitudes del cliente. Esto incluyó la validación de datos de entrada, la ejecución de operaciones en la base de datos y la preparación de respuestas adecuadas para el cliente.

Middleware

Se ha usado middleware de Express para ejecutar funciones antes o después del procesamiento de las rutas. Esto incluye middleware para la autenticación de usuarios, el manejo de errores y la gestión de sesiones.

Frontend

Creación de Páginas HTML

Cada página de la aplicación ha sido desarrollada utilizando HTML para definir la estructura y los elementos de la interfaz de usuario, tales como formularios, tablas y otros elementos HTML según los requisitos de cada página.

Estilización con CSS

Se han aplicado estilos y diseño a las páginas HTML utilizando CSS, para definir colores, fuentes, márgenes, y otros aspectos visuales de la aplicación, con una metodología de diseño responsivo para garantizar que la aplicación fuera visualmente atractiva y funcional en una variedad de dispositivos y tamaños de pantalla.

Interactividad con JavaScript

JavaScript agrega interactividad a las páginas. Esto incluye la validación de formularios, actualización dinámica de contenidos y la manipulación del DOM para lograr una experiencia de usuario más fluida.

Base de Datos: Diseño y Implementación

Modelado de Datos

Se ha diseñado el esquema de la base de datos, identificando las entidades principales (usuarios, pedidos, productos…) y estableciendo relaciones entre ellas, así como un selección de tipos de datos apropiados para almacenar la información y restricciones para garantizar la integridad de los datos.

Implementación de la Estructura

Se utilizaron sentencias SQL o herramientas de mapeo objeto-relacional (ORM) para crear las tablas y relaciones en la base de datos.
Se implementaron índices y claves primarias/foráneas según las mejores prácticas para optimizar el rendimiento y la integridad de la base de datos.

Acceso a la Base de Datos

La interacción con la base de datos se gestiona mediante DAO (Data Access Objects) y VO (Value Objects). Estos objetos abstractos facilitan las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la base de datos, proporcionando una interfaz sencilla y consistente para la lógica de negocio.

Este enfoque detallado en el backend, frontend y diseño de la base de datos garantiza una aplicación web integral, con una arquitectura sólida y una interfaz de usuario atractiva y funcional. Además, la estructura de la base de datos se diseñó para garantizar la eficiencia y la integridad de los datos almacenados.

Dificultades Encontradas

Durante el desarrollo del proyecto, se encontraron las siguientes dificultades:

Dificultades en la Colaboración del Código

Al haber varios desarrolladores colaborando en el proyecto, surgen conflictos en el control de versiones y problemas de integración.

La solución ha consistido en establecer prácticas sólidas de ramificación y fusión en Git, utilizar herramientas de revisión de código para identificar y corregir problemas antes de la fusión y mantener una comunicación clara y documentación actualizada para facilitar la colaboración.

Inconsistencia en los Datos de la Base de Datos

Pueden ocurrir inconsistencias en los datos de la base de datos debido a errores durante las operaciones de escritura o actualización.

Asegurar la atomicidad de las operaciones en la base de datos y validar los datos antes de realizar operaciones de escritura para evitar problemas de integridad son las soluciones propuestas contra esta dificultad.

Dificultades en la Conexión con la Base de Datos

Pueden surgir problemas de conexión con la base de datos, como tiempos de espera excesivos, desconexiones inesperadas o errores de autenticación.

La solución recae en asegurarse de que las credenciales de conexión sean correctas e implementar manejo de errores y reconexión automática en el código para gestionar problemas temporales.

Plan de Pruebas de Usuario

Se han creado pruebas unitarias para cada componente individual de la aplicación, como las funciones de la lógica de negocio, rutas del servidor y operaciones de base de datos.
Esto permite verificar que cada parte del código funcione como se espera de manera aislada, identificando posibles errores y asegurando la calidad del código.
Pruebas de Integración:

También se han ejecutado pruebas de integración para garantizar que los distintos componentes de la aplicación trabajen correctamente juntos.
Estas son cruciales para identificar posibles problemas de comunicación o incompatibilidades entre módulos, asegurando la cohesión general de la aplicación.

Se han simulado casos de uso del usuario final para evaluar la aplicación desde la perspectiva del usuario.
Se busca confirmar que la aplicación cumple con los requisitos y expectativas del usuario, evaluando la experiencia del usuario y asegurando que todas las funcionalidades sean accesibles y efectivas.

Plan de Mantenimiento

El plan de mantenimiento es esencial para garantizar el correcto funcionamiento y la eficiencia continua de la aplicación web en un entorno de producción. A continuación, se detallan cada uno de los componentes del plan:

Monitoreo Continuo

Se establecerá un sistema de monitoreo constante para supervisar la salud y el rendimiento de la aplicación.
Se vigilarán métricas clave, como el uso de recursos del servidor, tiempos de respuesta, tasas de error y disponibilidad.

Se implementarán herramientas de monitoreo automatizado para recibir alertas en tiempo real sobre posibles problemas y revisiones periódicas de registros y métricas para identificar patrones y áreas de mejora.

La detección temprana de posibles problemas permite una respuesta rápida y la mitigación de riesgos.

Actualizaciones de Seguridad

Se establecerá un protocolo para aplicar parches y actualizaciones de seguridad de manera regular, así como mantener todas las dependencias, bibliotecas y el sistema operativo actualizados.

Una programación regular de mantenimientos preventivos para aplicar actualizaciones críticas y la implementación de procesos automatizados para evaluar y aplicar parches de seguridad provoca beneficios de reducción del riesgo de vulnerabilidades de seguridad y un aseguramiento de que la aplicación esté protegida contra amenazas emergentes.

Soporte Técnico

Se establecerá un sistema de soporte técnico para abordar problemas reportados por los usuarios y para responder a sus consultas y se proporcionará un canal claro y eficiente para que los usuarios informen problemas y soliciten asistencia.

Se designará un equipo de soporte técnico con roles y responsabilidades claramente definidos para una resolución rápida y eficiente de problemas, mejorando la satisfacción del usuario.

El plan de mantenimiento no solo garantiza la estabilidad operativa de la aplicación en producción, sino que también refleja un compromiso continuo con la seguridad y la satisfacción del usuario. La combinación de monitoreo constante, actualizaciones de seguridad y soporte técnico proactivo contribuye significativamente a la longevidad y éxito continuo de la aplicación web.

Conclusión

En este proyecto, la elección de tecnologías como Node.js, Express, HTML y CSS ha resultado en una aplicación robusta y eficiente. La metodología de desarrollo iterativo, respaldada por pruebas exhaustivas ha asegurado la funcionalidad y usabilidad.

La implementación de un plan de mantenimiento sólido, que incluye monitoreo continuo, actualizaciones de seguridad y soporte técnico, refleja un compromiso constante con la calidad y la seguridad de la aplicación en producción.

En resumen, el proyecto no solo ha alcanzado sus objetivos de desarrollo, sino que también establece las bases para la sostenibilidad y mejora continua de la plataforma web. La combinación de tecnologías modernas, enfoques de desarrollo cuidadosos y un plan de mantenimiento integral contribuye al éxito a largo plazo de la aplicación.

Gestión de horas

Actividad	Roldán Urueña, Diego	Romeo Lancina, Abel	Moreno Muñoz, Pablo	Total
Creación de VO y DAO	4 horas	11 horas	3 horas	**18 horas**
Gestión de la base de datos	3 hora	3 horas	9 horas	**15 horas**
Creación de HTML y CSS	16 horas	10 horas	13 horas	**39 horas**
Códigos JavaScript	15 horas	10 horas	11 horas	**36 horas**
Realización de pruebas de usuario	8 horas	8 horas	8 horas	**24 horas**
Realización de la memoria	1 hora	5 horas	3 horas	**9 horas**
Total	**47 horas**	**47 horas**	**47 horas**	**141 horas**

Bibliografía

Documentación oficial de Node.js:
https://nodejs.org/en/docs
Documentación oficial de Express.js:
https://expressjs.com/
Documentación oficial de HTML y CSS
https://developer.mozilla.org/en-US/docs/Web/CSS
https://developer.mozilla.org/en-US/docs/Web/HTML
Documentación oficial de Visual Studio Code:
https://code.visualstudio.com/docs
Documentación del curso Sistemas de la Información 3º Ing. informática:
https://moodle.unizar.es/