#### C:\Users\Esmeralda\Pictures\umg.jpg

#### Universidad Mariano Gálvez de Guatemala Facultad en Sistemas

#### Curso: Programación WEB

PROYECTO:

Manual técnico

#### Autor: Asbel Ottoniell Sosa Dominguez Brandon Fernando Boror Mayen

#### Carné: 9490-08-36

#### 9490-18-6287

#### Guatemala noviembre de 2023

# Introducción

Este manual técnico proporciona una descripción detallada de la arquitectura y los componentes clave de eCommercePro React, un sistema de comercio electrónico desarrollado en React. Está diseñado para desarrolladores y equipos técnicos que deseen comprender la estructura y el funcionamiento del sistema. Para trabajar con éxito en el proyecto, es importante contar con conocimientos sólidos de React, JavaScript y herramientas relacionadas.

Contenido

[Introducción 2](#_Toc149939620)

[1. Arquitectura del Sistema 4](#_Toc149939621)

[2. Tecnologías Utilizadas 4](#_Toc149939622)

[3. Estructura de Directorios 4](#_Toc149939623)

[4. Componentes Principales 5](#_Toc149939624)

[5. Seguridad y Autenticación 5](#_Toc149939625)

[6. Gestión del Carrito de Compras 5](#_Toc149939626)

[7. Conexión a la Base de Datos 5](#_Toc149939627)

[8. Despliegue y Hosting 5](#_Toc149939628)

[9. Escalabilidad y Rendimiento 6](#_Toc149939629)

[Conclusiones 7](#_Toc149939630)

[Recomendaciones 8](#_Toc149939631)

[Recomendaciones 10](#_Toc149939632)

# 1. Arquitectura del Sistema

eCommercePro React sigue una arquitectura de una sola página (SPA) construida en torno a React. La aplicación se divide en componentes reutilizables y se comunica con un servidor backend para la gestión de datos y la autenticación de usuarios.

# 2. Tecnologías Utilizadas

React: El núcleo de la aplicación.

React Router: Para la navegación y la gestión de rutas.

Redux: Para el estado global de la aplicación y la gestión del carrito de compras.

Axios: Para realizar peticiones HTTP al servidor backend.

Node.js y Express.js: Utilizados en el servidor backend.

MongoDB: Base de datos para el almacenamiento de productos, usuarios y pedidos.

JSON Web Tokens (JWT): Utilizados para la autenticación y autorización de usuarios.

# 3. Estructura de Directorios

La estructura de directorios del proyecto eCommercePro React se organiza de la siguiente manera:

src/: Contiene los archivos fuente de la aplicación React.

server/: Contiene los archivos del servidor backend.

public/: Contiene los recursos estáticos, como imágenes y archivos CSS.

# 4. Componentes Principales

Los componentes clave de la aplicación incluyen:

App.js: El componente raíz que define la estructura general de la aplicación.

components/: Carpeta que contiene componentes reutilizables, como Product, Cart, UserDashboard, etc.

redux/: Contiene los archivos relacionados con la gestión del estado global de la aplicación.

# 5. Seguridad y Autenticación

La autenticación se maneja mediante JSON Web Tokens (JWT). Al iniciar sesión, se genera un token que se almacena en el lado del cliente y se envía con cada solicitud al servidor para verificar la autenticidad del usuario.

# 6. Gestión del Carrito de Compras

El estado del carrito de compras se gestiona utilizando Redux. Cada acción en la interfaz de usuario dispara eventos que modifican el estado del carrito.

# 7. Conexión a la Base de Datos

La aplicación se conecta a una base de datos MongoDB para almacenar productos, usuarios y pedidos. La comunicación con la base de datos se realiza a través del servidor backend.

# 8. Despliegue y Hosting

Para desplegar la aplicación en producción, recomendamos usar servicios de alojamiento web que sean compatibles con aplicaciones React, como Netlify o Vercel para el frontend, y servicios de alojamiento Node.js para el backend. Asegúrate de configurar las variables de entorno de manera segura.

# 9. Escalabilidad y Rendimiento

Para mejorar la escalabilidad y el rendimiento, considera el uso de técnicas como la optimización del código, el almacenamiento en caché, la compresión de recursos y la distribución de la carga en el servidor backend.

# Conclusiones

Este manual técnico ha ofrecido una visión general exhaustiva de eCommercePro React y su infraestructura subyacente. No obstante, es importante destacar que el desarrollo de aplicaciones es un proceso dinámico y que la documentación aquí presentada refleja el estado del sistema en el momento de su creación. Para una comprensión más profunda y detalles específicos sobre la implementación de cada componente, te recomendamos consultar el código fuente de la aplicación y los comentarios detallados en el proyecto.

La exploración del código fuente es esencial para quienes desean contribuir al desarrollo, realizar personalizaciones o solucionar problemas. Además, puede servir como una valiosa fuente de aprendizaje para aquellos que desean comprender en profundidad la lógica y la arquitectura del sistema.

# Recomendaciones

Dada la complejidad y la importancia de mantener actualizada y segura una aplicación como eCommercePro React, se sugiere encarecidamente seguir las siguientes recomendaciones:

Mantén una documentación actualizada: A medida que realices cambios en la aplicación, asegúrate de reflejarlos en la documentación técnica. Esto ayudará a futuros desarrolladores y equipos a comprender y mantener el sistema de manera eficiente.

Implementa pruebas unitarias y de integración: Las pruebas son esenciales para garantizar la estabilidad y el funcionamiento correcto de la aplicación. Incorpora pruebas de unidad y de integración en el proceso de desarrollo para detectar y solucionar problemas de manera proactiva.

Monitorea el rendimiento: Utiliza herramientas de monitoreo de rendimiento para identificar cuellos de botella, ineficiencias y posibles problemas de escalabilidad. Asegúrate de que la aplicación funcione de manera eficiente incluso bajo cargas elevadas.

Mantén actualizadas las dependencias: React, Redux y otras bibliotecas y módulos utilizados en la aplicación evolucionan con el tiempo. Asegúrate de mantener al día todas las dependencias y aplica parches de seguridad de manera regular.

Seguridad de la información: Protege la información de los usuarios y los datos sensibles. Implementa buenas prácticas de seguridad, como la validación de entrada, el cifrado de contraseñas y la gestión segura de tokens JWT.

Respaldo regular de la base de datos: Realiza copias de seguridad regulares de la base de datos y almacénalas de forma segura. Esto garantizará que los datos estén protegidos en caso de fallos o pérdidas.

Escalabilidad: Prevé el crecimiento de la aplicación y su capacidad para manejar un mayor número de usuarios y productos. Considera el uso de tecnologías de escalabilidad horizontal, como la implementación de sistemas de caché y el equilibrio de carga.

Seguimiento de errores y mejoras: Utiliza herramientas de seguimiento de errores para rastrear problemas y solicitudes de mejoras. Prioriza y aborda estos problemas de manera eficiente para mantener una aplicación confiable.

Mantén la seguridad del servidor: Asegúrate de que el servidor esté debidamente configurado y protegido contra amenazas comunes. Realiza auditorías de seguridad regularmente.

Colaboración y comunicación: Fomenta una comunicación abierta y una colaboración efectiva dentro del equipo de desarrollo. Mantén a todos los miembros informados sobre los cambios y actualizaciones importantes.