Descripción: X es una empresa que ofrece servicios de transporte y logística, especializada en la entrega de productos de empresas de comercio electrónico. Su enfoque es la sostenibilidad: utiliza una flota de vehículos eléctricos y bicicletas para minimizar su huella de carbono. Aunque está bien posicionada en el mercado y cuenta con una amplia base de clientes, enfrenta dificultades en la gestión de sus operaciones diarias, lo que afecta la eficiencia del servicio.

Problemática:

enfrenta problemas en la **gestión de rutas de entrega** y en la comunicación con los clientes. A menudo, los pedidos se retrasan porque la empresa no tiene un sistema optimizado para asignar y organizar los paquetes en las **rutas más eficientes.** Además, los clientes no tienen visibilidad en **tiempo real de la ubicación de sus paquetes y del tiempo estimado de llegada**. Esto genera insatisfacción y baja fidelización.

Requerimientos del Sistema

Objetivo: Crear una plataforma de gestión y seguimiento de entregas que permita a **optimizar rutas** y proporcionar a los clientes actualizaciones en tiempo real sobre el estado de sus entregas.

Módulos del Sistema

1. Portal de Administración (Back-office para )

- Gestión de Pedidos: Una sección para que los administradores puedan ver los pedidos y asignarlos a los conductores disponibles.

- Optimización de Rutas: Integración con un servicio de mapas para calcular rutas óptimas, considerando el tráfico y la cercanía de los pedidos.

- Gestión de Conductores: Registro y seguimiento de los conductores activos, sus datos, y el estado de sus vehículos.

- Panel de Monitoreo en Tiempo Real: Un mapa en tiempo real donde se pueden visualizar las posiciones de los vehículos y el estado de cada entrega (en proceso, entregado, etc.).

2. Portal para Conductores

- Iniciar Sesión: Los conductores deben iniciar sesión para ver sus tareas del día.

- Lista de Entregas Asignadas: Visualización de la lista de entregas asignadas y la ruta optimizada.

- Actualización del Estado del Pedido: Los conductores pueden cambiar el estado de cada pedido (recogido, en camino, entregado).

- Sistema de Mensajería: Comunicación directa con el administrador en caso de problemas o retrasos.

3. Portal para Clientes

- Rastreo de Pedidos: Los clientes pueden introducir su número de pedido y ver la ubicación de su paquete y el tiempo estimado de llegada en tiempo real.

- Notificaciones: Recibir notificaciones sobre el estado del pedido (pedido en camino, pedido próximo a llegar, pedido entregado).

- Historial de Pedidos: Consultar pedidos pasados y su estado de entrega.

4. API RESTful

- Crear una API en Node.js y Express para conectar la base de datos (PostgreSQL) con las aplicaciones de administración, conductores y clientes.

- Endpoint para gestionar los pedidos, usuarios, rutas y notificaciones.

5. Base de Datos (PostgreSQL)

- Modelo de datos: Deberá almacenar la información de pedidos, usuarios (administradores, conductores y clientes), rutas optimizadas y estado de cada pedido.

- Relaciones:

- Conductores a Pedidos (1 a muchos)

- Pedidos a Clientes (1 a 1)

- Rutas a Pedidos (1 a 1)

Funcionalidades Clave

1. Optimización de Rutas: Crear algoritmos o integrar APIs externas para calcular las rutas óptimas, mejorando así la eficiencia y reduciendo el tiempo de entrega.

2. Tracking en Tiempo Real: La posición de los vehículos de entrega se actualiza periódicamente y se muestra en el portal del cliente y en el panel de administración en tiempo real.

3. Gestión de Estado: Los pedidos deben poder marcarse en diferentes estados (pendiente, en proceso, entregado), permitiendo a los conductores y administradores actualizar el estado según el progreso de la entrega.

4. Notificaciones Automáticas: Utilizar notificaciones push o SMS para alertar a los clientes en diferentes etapas del envío.

5. Historial y Reportes: Crear reportes que permitan a la administración ver métricas de eficiencia, como tiempos promedio de entrega, pedidos entregados por conductor, y áreas geográficas con más entregas.

Consideraciones Técnicas

1. Frontend (React): Desarrollar una interfaz de usuario intuitiva y rápida para cada portal (administrador, conductor y cliente), utilizando componentes modernos y una arquitectura bien organizada.

2. Backend (Node.js + Express): Configurar la API RESTful para que sea segura y eficiente, optimizando las consultas a la base de datos y gestionando la lógica de negocio necesaria para la asignación de rutas y pedidos

3. Base de Datos (PostgreSQL): Diseño de base de datos con relaciones adecuadas, índices en las tablas necesarias y mecanismos de seguridad para el acceso a datos sensibles (como la ubicación en tiempo real de los conductores).

4. Seguridad: Implementar autenticación segura (JWT o similar) para el acceso de usuarios (administrador, conductores y clientes) y asegurar las transacciones en la base de datos.

5. Escalabilidad: Asegurarse de que el sistema pueda manejar múltiples usuarios simultáneamente, especialmente durante picos de entrega.

6. Integración con APIs Externas: Para las funcionalidades de mapeo y optimización de rutas, considerar la integración con servicios como Google Maps o Mapbox.