Título del Proyecto: Aplicativo de Transporte de Domicilios Locales de Comidas Rápidas

Contexto:

Eres parte de un equipo de desarrollo de software encargado de crear un aplicativo para gestionar pedidos de domicilios de comidas rápidas en locales de tu ciudad. El aplicativo debe permitir a los usuarios ver los menús de los restaurantes, realizar pedidos, actualizar el estado de los pedidos y visualizar todos los pedidos realizados.

Objetivo:

Tu tarea es **analizar, corregir y mejorar** el código proporcionado, asegurándote de que cumpla con los estándares de calidad de la industria del software. Además, deberás **implementar un plan de pruebas** para garantizar que el aplicativo funcione correctamente en todos los escenarios posibles.

Actividades a Realizar

1. Análisis del Código Existente

- Identifica y documenta todos los errores en el código proporcionado.
- Clasifica los errores en las siguientes categorías:
  - Errores de sintaxis.
  - Errores lógicos.
  - Problemas de calidad del código (malas prácticas, falta de optimización, etc.).
  - Falta de manejo de errores y validaciones.

2. Corrección de Errores

- Corrige todos los errores identificados en el código.
- Asegúrate de que el código cumpla con las siguientes buenas prácticas:
  - Uso correcto de const, let y var.

- Validación de entradas (por ejemplo, verificar que los items existan en el menú del restaurante).
- Manejo de errores (por ejemplo, mostrar mensajes claros cuando un restaurante no existe).
- Optimización del código (por ejemplo, evitar búsquedas innecesarias).

3. Mejoras del Código

- Refactoriza el código para mejorar su legibilidad y mantenibilidad.
- Añade comentarios y documentación para explicar el funcionamiento de cada función.
- Implementa nuevas funcionalidades, como:
  - Cancelar un pedido.
  - Calcular el costo total de un pedido.
  - Filtrar pedidos por estado (Pendiente, En camino, Entregado).

4. Implementación de Pruebas

- Diseña y ejecuta un plan de pruebas que incluya:
  - **Pruebas unitarias**: Verifica el correcto funcionamiento de cada función individualmente.
  - **Pruebas de integración**: Verifica que las funciones trabajen correctamente en conjunto.
  - **Pruebas de validación**: Asegúrate de que el aplicativo maneje correctamente entradas inválidas.
  - **Pruebas de rendimiento**: Evalúa el comportamiento del aplicativo con un gran volumen de datos.
  - Utiliza herramientas como **Jest** o **Mocha** o **Selenium** para automatizar las pruebas.

5. Entrega Final

- Entrega un informe que incluya:
  - Lista de errores identificados y cómo se corrigieron.
  - Explicación de las mejoras implementadas.
  - Resultados de las pruebas realizadas.
  - Código fuente corregido y mejorado.

```
// Aplicativo de Transporte de Domicilios Locales de Comidas
Rápidas // Datos de ejemplo
const restaurants = [
 { id: 1, name: "Burger King", menu: ["Whopper", "Cheeseburger", "Fries"]
},  { id: 2, name: "McDonald's", menu: ["Big Mac", "McChicken",
"McFlurry"] },  { id: 3, name: "KFC", menu: ["Bucket Chicken", "Zinger
Burger", "Coleslaw"] } ];
const orders = [];
// Función para mostrar el menú de un restaurante
function showMenu(restaurantId) {
 const restaurant = restaurants.find(r => r.id =
restaurantId);  if (restaurant) {
 console.log(`Menú de ${restaurant.name}:`);
 restaurant.menu.forEach(item => console.log(item));
 } else {
 console.log("Restaurante no encontrado.");
 }
}
// Función para realizar un pedido
function placeOrder(restaurantId, items, address) {
 const restaurant = restaurants.find(r => r.id ==
restaurantId);  if (!restaurant) {
 console.log("Restaurante no encontrado.");
 return; }
 const order = {
id: orders.length + 1,
 restaurant: restaurant.name,
 items: items,
```

```
 address: address,
 status: "Pendiente"
 };
 orders.push(order);
 console.log(`Pedido realizado con ID: ${order.id}`);
}
// Función para actualizar el estado de un pedido
function updateOrderStatus(orderId, status) {
 const order = orders.find(o => o.id === orderId);
 if (order) {
 order.status = status;
 console.log(`Estado del pedido ${orderId} actualizado a: ${status}`);
} else {
 console.log("Pedido no encontrado.");
 }
}
// Función para mostrar todos los pedidos
function showOrders() {
 console.log("Lista de pedidos:");
 orders.forEach(order => {
 console.log(`ID: ${order.id}, Restaurante: ${order.restaurant}, Items:
${order.items.join(", ")}, Dirección: ${order.address}, Estado:
${order.status}`);
 });
}
// Ejemplos de uso
showMenu(1);
placeOrder(1, ["Whopper", "Fries"], "Calle
```

```
123"); placeOrder(2, ["Big Mac"], "Avenida
456"); updateOrderStatus(1, "En camino");
showOrders();
```