

Servicio Nacional de Aprendizaje SENA

Programa: Tecnólogo en Análisis y Desarrollo de Software

Ficha Número: 2775029

Presentado por: Sebastián Urrego

1. Análisis del Código Existente

Se revisó el código del aplicativo y se identificaron errores en varias categorías:

1.1 Errores de Sintaxis

Uso incorrecto de asignación en la función showMenu:

```
const restaurant = restaurants.find(r => r.id = restaurantId);
```

Corrección: Se reemplazó = por === para realizar una comparación adecuada.

1.2 Errores Lógicos

Validación de pedidos: No se verificaba si los productos solicitados existían en el menú del restaurante antes de confirmar la orden.

Corrección: Se agregó una validación para evitar que se agreguen items inexistentes.

Manejo de pedidos inexistentes: No se manejaba correctamente el caso en `updateOrderStatus` cuando el pedido no existía.

Corrección: Se agregó un mensaje de error si el pedido no se encuentra.

1.3 Problemas de Calidad del Código

Uso incorrecto de comparaciones (`==` en lugar de `===`).

Falta de comentarios explicativos en las funciones.

Salida de mensajes en consola poco clara.

1.4 Falta de Manejo de Errores y Validaciones

Pedidos vacíos: `showOrders` imprimía una lista vacía sin advertir al usuario.

Corrección: Se agregó una condición para mostrar un mensaje cuando no hay pedidos disponibles.

Direcciones vacías en `placeOrder`.

Corrección: Se agregó validación para evitar direcciones vacías.

2. Corrección de Errores

Se aplicaron las siguientes mejoras:

Se aseguró el uso correcto de const, let y var.

Se validaron correctamente las entradas del usuario.

Se optimizó la búsqueda de restaurantes y pedidos.

3. Mejoras Implementadas

3.1 Refactorización de Funciones

placeOrder: Ahora verifica que los productos solicitados existan en el menú antes de agregar el pedido.

showOrders: Evita imprimir una lista vacía si no hay pedidos disponibles.

updateOrderStatus: Maneja correctamente pedidos inexistentes mostrando un mensaje adecuado.

3.2 Nuevas Funcionalidades

Cancelar un pedido: Se agregó una función para eliminar pedidos antes de que sean entregados.

Calcular costo total: Se incluyó una función para mostrar el total del pedido con precios simulados.

Filtrar pedidos por estado: Se permite visualizar pedidos según su estado.

4. Implementación de Pruebas

Se diseñó un plan de pruebas con los siguientes tipos:

4.1 Pruebas Unitarias

Validar que `showMenu` imprima correctamente el menú de un restaurante existente y maneje restaurantes inexistentes.

Verificar que `placeOrder` no permita agregar pedidos con productos inexistentes.

4.2 Pruebas de Integración

Crear varios pedidos y cambiar su estado, asegurando que `updateOrderStatus` los maneje correctamente.

4.3 Pruebas de Validación

Intentar realizar pedidos con direcciones vacías y verificar el comportamiento del sistema.

Probar con ID de restaurante inexistente.

4.4 Pruebas de Rendimiento

Agregar una gran cantidad de pedidos y medir tiempos de respuesta.

5. Conclusiones

La refactorización mejoró la estabilidad, mantenibilidad y calidad del código. Con la implementación del plan de pruebas, se garantiza un comportamiento robusto del aplicativo.

El código final corregido y mejorado está listo para ser implementado y probado en un entorno de producción.