

Servicio Nacional de Aprendizaje SENA

Programa: Tecnólogo en Análisis y Desarrollo de Software

Ficha Número: 2775029

Presentado por: Sebastián Urrego

Se realizó una refactorización del código del aplicativo de transporte de domicilios para corregir errores, mejorar su legibilidad y optimizar su funcionamiento. A continuación, se detallan los cambios aplicados.

Errores Identificados y Corregidos

1. Errores de Sintaxis

Uso incorrecto de asignación en la función showMenu:

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

Se corrigió utilizando === en lugar de = para la comparación.

2. Errores Lógicos

Validación de pedidos:

Se añadió una verificación para asegurar que los items del pedido existan en el menú del restaurante antes de confirmar la orden.

Se muestra un mensaje de error en caso de items no disponibles.

3. Problemas de Calidad del Código

Uso incorrecto de comparaciones:

Se reemplazó `==` por `===` para mejorar la robustez del código y evitar conversiones implícitas.

Legibilidad y mantenibilidad:

Se agregaron comentarios explicativos en las funciones.

Se mejoró la estructura de salida de los mensajes en la consola.

4. Falta de Manejo de Errores y Validaciones

Verificación de existencia de pedidos:

Se agregó una condición en `showOrders` para mostrar un mensaje cuando no hay pedidos disponibles.

Mejoras Implementadas

1. Refactorización de Funciones

`placeOrder:`

Ahora valida si los productos solicitados existen en el menú antes de agregar el pedido.

`showOrders:`

Evita imprimir una lista vacía si no hay pedidos.

updateOrderStatus:

Maneja correctamente pedidos inexistentes mostrando un mensaje adecuado.

2. Optimizaciones

Estructura del código:

Se reorganizaron las funciones para mejorar su claridad y separación de responsabilidades.

Se redujo la duplicación de código mediante validaciones más eficientes.

Plan de Pruebas

Para garantizar el correcto funcionamiento del aplicativo, se diseñó un plan de pruebas incluyendo:

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.

2. Pruebas de Integración

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

3. Pruebas de Validación

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

Probar con ID de restaurante inexistente.

4. Pruebas de Rendimiento

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

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.