

Errores detectados y correcciones

1. Errores de sintaxis

- En la función `showMenu(restaurantId)`, el operador de asignación `=` se usó en lugar de `===` en:

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

Corrección:

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

2. Errores lógicos

- En `placeOrder()`, no se valida si los elementos del pedido están en el menú del restaurante.
- `updateOrderStatus()` no valida si el estado es válido (solo debería aceptar "Pendiente", "En camino", "Entregado").

3. Problemas de calidad del código

- Uso inconsistente de `const` y `let`. Se recomienda usar `const` siempre que la variable no cambie.
- Faltan comentarios para mejorar la legibilidad.
- No hay manejo de errores con `try...catch`.
- No hay función para cancelar pedidos.
- El código no maneja entradas inválidas adecuadamente.

Mejoras implementadas

1. Corrección de errores sintácticos y lógicos:

- Se corrigió el error en `showMenu()`.
- Se agregó validación para que los pedidos solo contengan ítems disponibles en el restaurante.
- Se agregó validación de estado en `updateOrderStatus()`.

2. Optimización y buenas prácticas:

- Uso de `const` y `let` correctamente.
- Se eliminaron búsquedas innecesarias.
- Se agregó documentación y comentarios.

3. Nuevas funcionalidades:

- Función `cancelOrder()` para cancelar pedidos si están en estado "Pendiente".
- Función `filterOrdersByStatus()` para filtrar pedidos según su estado.
- Función `calculateOrderCost()` para calcular el precio total de un pedido.

Pruebas recomendadas

Pruebas unitarias:

- Probar `showMenu()` con IDs existentes y no existentes.
- Validar `placeOrder()` con ítems válidos e inválidos.
- Comprobar `updateOrderStatus()` con estados correctos e incorrectos.
- Asegurar que `cancelOrder()` no elimine pedidos ya procesados.

Pruebas de integración:

- Realizar un pedido, actualizar su estado y verificar que los cambios sean correctos.

Pruebas de validación:

- Intentar hacer pedidos con ítems no disponibles.
- Intentar actualizar un estado con un valor inválido.