# Documentación de Cambios, Arreglos y Mejoras

DOCUMENTACIÓN DE CAMBIOS, ARREGLOS Y MEJORAS

---

## 1. Error en la autenticación con Google

### Problema:

Al intentar iniciar sesión con Google, el botón no realizaba ninguna acción y no llegaba petición alguna a la API.

### Causa:

El método `loginGoogle` de la clase `AuthApi` perdía el contexto de `this` al ser pasado directamente como parámetro a `createAsyncThunk`, lo que ocasionaba que `this.baseURL` no estuviera definido.

### Solución:

Se utilizó `.bind(authApi)` para mantener el contexto correcto.

export const loginUserGoogle = createAsyncThunk("user/loginUserGoogle", authApi.loginGoogle.bind(authApi))

---

## 2. Cambios en `itemSlice.js`

- Se corrigió la asignación del precio unitario para que tome el valor desde `item.price` en lugar de `item.unit\_price`.

### Antes:

unit\_price: formatPrice(item.unit\_price)

### Después:

unit\_price: formatPrice(item.price)

---

## 3. Cambios en `orderSlice.js`

- Se modificó la importación de `formatPrice` para utilizar la versión localizada en `shared/utils/formatPriceOrders`.

- Se eliminó el formateo de `total\_amount` en la creación de órdenes para que este valor se maneje de forma más flexible y acorde al flujo posterior.

---

## 4. Cambios en `Buttons.jsx`

- El texto del botón `"Agregar Orden"` fue cambiado por `"Nueva Orden"` para mayor claridad.

---

## 5. Cambios en `ItemModal.jsx`

- Se mejoró la función `formatPrice` para manejar valores `null` o `undefined`.

- Se cambió `selectedProduct` para que sea un objeto único en lugar de un array.

- Se agregó control para cerrar el modal al usar el botón de retroceso del navegador.

- Se modificó el guardado del producto para que use `price` en vez de `unit\_price` y para reiniciar los estados al cerrar.

---

## 6. Cambios en `OrderDetails.jsx`

- Se agregó el formateo correcto de precios usando `formatPriceOrders`.

- Se reestructuró la función `handleOrderSave` para normalizar items antes de guardar y ejecutar las operaciones de creación/actualización en paralelo.

- Se mejoró la visualización del subtotal y cálculo del total neto.

---

## 7. Cambios en `SearchItemsProduct.jsx`

- Se integró `useDispatch` para manejar el estado de `selectedProduct` de forma centralizada.

- Se agregó cierre automático de la lista al hacer clic fuera.

- Se implementó filtrado de productos en tiempo real y búsqueda más robusta.

---

## 8. Cambios en `productSlice.js`

- `selectedProduct` pasó de ser un array a un valor único (`null` por defecto).

- `setSelectedProduct` ahora asigna directamente el producto en lugar de hacer push.

- `voidSelectedProduct` restablece a `null` en lugar de un array vacío.

---

\*\*Objetivo General de los Cambios:\*\*

Optimizar el manejo de estado, mejorar la coherencia en la estructura de datos, evitar errores de contexto (`this`) y ofrecer una mejor experiencia de usuario en el flujo de órdenes y selección de productos.