**Documentación de Cambios, Arreglos y Mejoras (commit14-08-25)**

**1. Corrección del Problema de Autenticación con Google**

**Problema Detectado**

El método **loginGoogle** de la clase **AuthApi** no ejecutaba la redirección de autenticación con Google.

A pesar de que el código parecía estar bien estructurado, al presionar el botón no sucedía nada y no se enviaba ninguna petición a la API.

Tras revisar, se determinó que el problema estaba en el **frontend:**

* El método es asincrónico y devuelve una promesa.
* Al ser pasado directamente a createAsyncThunk en userSlice, perdía su contexto de this.
* Esto provocaba que this.baseURL no estuviera definido, impidiendo construir correctamente la URL de autenticación.

**Solución Implementada**

Se aseguró que el método loginGoogle mantuviera su contexto original (this) al ser invocado por createAsyncThunk.

Para ello, se usó .bind() para vincular explícitamente el método a la instancia de la clase authApi.

**Código Corregido:**

**export const loginUserGoogle = createAsyncThunk(**

**"user/loginUserGoogle",**

**authApi.loginGoogle.bind(authApi)**

**)**

Ahora, this dentro de loginGoogle siempre se refiere a la instancia authApi, permitiendo:

* Acceso correcto a baseURL.
* Ejecución exitosa de la redirección.

**2. Mejoras en la Gestión de Precios y Formato de Datos**

**Justificación de los Cambios**

Se aplicaron ajustes clave para mejorar la robustez y consistencia en el manejo de precios y datos:

**1. Uniformidad en precios:**

* En itemSlice.js, el campo unit\_price ahora se toma desde item.price.
* Esto estandariza la nomenclatura y evita referencias inconsistentes.

**2. Manejo de valores nulos o indefinidos:**

* En ItemModal.jsx, la función formatPrice ahora devuelve 0 si recibe null o undefined.
* Evita errores tipo NaN (Not a Number).

**3. Refactor en orderSlice.js:**

* Se eliminó el formateo previo del precio en createDataOrder.
* Ahora se recibe la orden sin formateo, permitiendo flexibilidad posterior.

**4. Optimización en OrderDetails.jsx:**

* Los ítems se normalizan antes de guardarlos.
* El subtotal se calcula con precios formateados.
* Creación/actualización de órdenes e ítems ahora se realiza en paralelo usando Promise.all → mejora de rendimiento.

**3. Mejoras en la Interfaz de Usuario y Experiencia de Usuario (UX)**

**Justificación de los Cambios**

Se optimizó la experiencia de usuario con varias mejoras:

**ItemModal.jsx**

* Cierre automático del modal con useEffect al retroceder en el historial del navegador.
* selectedProduct pasó de array ([]) a objeto único (null), simplificando la lógica de estado.
* Validación inicial para evitar renderizados erróneos si no hay producto seleccionado.

**SearchItemsProduct.jsx**

* Uso de useRef para cerrar el menú de resultados al hacer clic fuera.
* Filtrado solo activo con un término de búsqueda válido → evita búsquedas innecesarias y mejora el rendimiento.

**Buttons.jsx**

* Texto cambiado de "Agregar Orden" a "Nueva Orden" → mayor claridad para el usuario.

**Consistencia en nomenclatura**

* En OrderDetails.jsx, la prop pasada a SearchItemsProduct cambió de setModal a setIsModalOpen para mantener coherencia y evitar errores.