# AC7.1 – aplicación de comida rápida con Redux Toolkit

**Mónica Garrigós Sánchez**

## Sección 1: Redux Toolkit configurado

En esta sección se muestra que Redux Toolkit fue configurado correctamente con *configureStore*, y que el store fue conectado globalmente usando el componente *Provider*.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## Sección 2: Slices de Redux implementados

Aquí se muestran los slices *menuSlice* y *orderSlice*. *MenuSlice* gestiona los elementos del menú (actualización de cantidades). *OrderSlice* gestiona los pedidos usando *createAsyncThunk*, incluyendo *addOrder* y *getOrders*.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## Sección 3: Firebase como backend (persistencia)

Las operaciones CRUD se conectan a Firebase Firestore. Se usa *saveOrder*() para guardar pedidos y *getOrders*() para recuperarlos y mostrarlos.

Texto

El contenido generado por IA puede ser incorrecto.

## Sección 4: Guardado de pedidos con feedback

Al enviar un pedido, se muestra retroalimentación visual usando el estado *orderStatus*.  
El pedido se guarda con *dispatch(addOrder(...))* desde Redux.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

## Sección 5: Error y Logging

Se implementa un sistema de logging para registrar distintos niveles de eventos (debug, info, warning, error). Además, se usó un componente *ErrorBoundary* para capturar errores de ejecución y evitar que la aplicación se rompa.

En la siguiente captura de pantalla podemos observar que se lanza un error al intentar hacer un pedido con una cantidad negativa, este error se queda registrado en consola con un mensaje personalizado.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Sección 6: Visualización de pedidos

Los pedidos recientes se muestran en la sección de disponibilidad, accediendo a la lista *orders* del estado Redux. Esta lista se carga desde Firebase con *getORders* y se renderiza en el componente *OrderLosts.tsx*.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Sección 7: Sitio desplegado

La aplicación fue desplegada en GitHub Pages a través de la rama *gh*-*pages* usando Vite y la configuración correspondiente en *vite.config.ts.*