Práctica: Menú de Restaurante (aplicación de una sola página) con React + fetch

Descripción

Vas a crear una **aplicación de una sola página** que muestre el **menú de un restaurante**. La aplicación debe consumir una **API pública de comida** para obtener los platos y tú podrás **inventarte los precios**. El objetivo es practicar los **fundamentos de usabilidad**, el **manejo de componentes y props**, y el uso de **useEffect** junto con **fetch** para cargar datos.

Objetivos de aprendizaje

- Construir una aplicación de una sola página con React utilizando componentes funcionales.
- Consumir datos externos con **fetch** y gestionarlos con **useEffect**.
- Diseñar componentes reutilizables y pasar datos mediante **props**.
- Aplicar principios básicos de **usabilidad** (claridad, jerarquía visual, legibilidad, feedback de carga/errores).
- Gestionar estados de *loading* y *error* de manera adecuada.

Requisitos funcionales

- Mostrar un listado de platos con: **nombre**, **imagen**, **precio** y **categoría**.
- Cargar los datos mediante **fetch** dentro de un **useEffect** (sin librerías de datos).
- Inventar y asignar **precios** a los platos (la API no proporciona precios).
- Mostrar estados de **carga** (Loading...) y **error** (mensaje claro).
- Estructurar la interfaz con al menos **3 componentes**: <App>, <MenuList>, <MenuItem>.

Requisitos técnicos

- Proyecto con Vite o Create React App.
- Componentes funcionales, **useState** y **useEffect**.
- **fetch** nativo del navegador (sin Axios).
- Estilos sencillos (CSS o inline) con diseño responsive básico (grid o flex), se puede usar recursos externos como Boostrap, Component UI … etc.

API a utilizar (gratuita y sin registro)

Usa este endpoint de **TheMealDB** (no requiere clave):
https://www.themealdb.com/api/json/v1/1/filter.php?c=Seafood

(Podeis usar otros endpoint, pero tienen que ser comida)

La respuesta contiene una lista de platos (meals) con idMeal, strMeal (nombre) y strMealThumb (imagen). Como la API no incluye precios, debes **inventarlos** y mostrarlos en la tarjeta de cada plato.

Instrucciones paso a paso

1. Crea el proyecto React e inicializa el estado: *items*, *isLoading*, *error*.
2. En useEffect, realiza **fetch** al endpoint indicado y parsea la respuesta con res.json().
3. Normaliza los datos para que cada ítem tenga: id, name, category, thumb y price.
4. Pasa los datos como **props** a <MenuList> y luego a <MenuItem>.
5. Muestra **Loading...** mientras se cargan los datos y un **mensaje de error** si algo falla.
6. Diseña una disposición en **grid** para las tarjetas, con jerarquía visual clara (imagen, nombre, categoría, precio).

Fundamentos de usabilidad a aplicar

- Claridad y legibilidad: tipografía estándar y contraste suficiente.
- Jerarquía visual: imagen → nombre → categoría → precio.
- Feedback inmediato: estados de carga y mensajes de error comprensibles.
- Tocables/clicables adecuados (espaciado y tamaños).
- Responsive simple con columnas que se adapten al ancho.

Entregables

- Repositorio o carpeta con el proyecto React (Vite/CRA).
- Archivo README con una captura de pantalla y breve descripción.
- Uso del endpoint indicado de TheMealDB: https://www.themealdb.com/api/json/v1/1/filter.php?c=Seafood.

Criterios de evaluación (rúbrica sugerida)

- Estructura de componentes y uso de props: 30%
- Uso correcto de fetch + useEffect (loading/error): 25%
- Usabilidad y presentación (grid, jerarquía, responsive): 20%
- Calidad del código (nombres, limpieza, organización): 15%
- Extras (filtros/buscador/ordenar/carrito): 10%

Extras opcionales

- Filtro por categoría y/o buscador por nombre.
- Ordenar por precio asc/desc.

Nota

En modo desarrollo con React 18 y *StrictMode*, es normal que ciertos efectos se ejecuten dos veces. No ocurre en build de producción.

¡Éxitos! Prioriza código claro, componentes pequeños y una experiencia agradable para el usuario.