

Trabajo Práctico: Consumo de JSON Server y react router

Objetivo

Construir una aplicación en React que consuma datos de un JSON Server. Se deben mostrar cursos y al ingresar a un curso, se deben listar los estudiantes asociados utilizando rutas dinámicas y query params para persistencia al recargar tanto de los elementos para pedir a la bd y no perder la vista.

Estructura del JSON (db.json)

```
{
  "cursos": [
    {
      "id": 1,
      "nombre": "Programación Web",
      "estudiantes": [
        { "id": 1, "nombre": "Ana", "edad": 20 },
        { "id": 2, "nombre": "Luis", "edad": 22 },
        { "id": 3, "nombre": "Carlos", "edad": 21 },
        { "id": 4, "nombre": "Valentina", "edad": 23 },
        { "id": 5, "nombre": "Jorge", "edad": 20 },
        { "id": 6, "nombre": "Lucía", "edad": 22 },
        { "id": 7, "nombre": "Pedro", "edad": 24 },
        { "id": 8, "nombre": "Sofía", "edad": 21 },
        { "id": 9, "nombre": "Diego", "edad": 22 },
        { "id": 10, "nombre": "Martina", "edad": 20 }
      ]
    },
    {
      "id": 2,
      "nombre": "Base de Datos",
      "estudiantes": [
        { "id": 11, "nombre": "María", "edad": 21 },
        { "id": 12, "nombre": "Juan", "edad": 23 },
        { "id": 13, "nombre": "Emilia", "edad": 22 },
        { "id": 14, "nombre": "Nicolás", "edad": 20 },
        { "id": 15, "nombre": "Agustina", "edad": 21 },
        { "id": 16, "nombre": "Tomás", "edad": 22 },
        { "id": 17, "nombre": "Camila", "edad": 23 },
        { "id": 18, "nombre": "Lucas", "edad": 24 },
        { "id": 19, "nombre": "Florencia", "edad": 20 },
        { "id": 20, "nombre": "Iván", "edad": 22 }
      ]
    }
  ]
}
```

🧩 Funcionalidades requeridas

1. Vista principal de cursos (/cursos)

- Título de la pantalla.
- Mostrar tarjetas de cursos con nombre, cantidad de alumnos.
- Cada tarjeta tendrá un botón o enlace que redirige `/estudiantes?curso=ID`.

2. Vista de estudiantes (/estudiantes?curso=ID)

- Leer `query param curso` para filtrar y mostrar estudiantes del curso seleccionado.
- Si no hay parámetro, mostrar un mensaje de error o una pantalla vacía.
- Soportar recarga sin perder el curso activo.

3. 📦 Requerimientos técnicos

- React + typescript + React Router DOM v6
- Fetch de datos con `fetch` o `axios`
- Uso de `useEffect`, `useState` y `reactRouter`
- Opcional: Estilos con CSS, Bootstrap o Tailwind

4. 📁 Estructura del proyecto

```
src/  
├─ components/  
│   ├─ screens/                ─ Vistas principales que renderizan el contenido  
│   │   ├─ CursosScreen.tsx  
│   │   └─ EstudiantesScreen.tsx  
│   └─ ui/                    ─ Componentes reutilizables para cada vista  
│       ├─ CursoCard.tsx  
│       └─ EstudianteCard.tsx  
├─ http/                      ─ Lógica para consumir la API (json-server)  
│   └─ api.ts  
├─ routes/                    ─ Configuración de rutas con React Router  
│   └─ AppRouter.tsx  
├─ App.tsx                    ─ Componente raíz  
└─ main.tsx                   ─ Punto de entrada
```

5. 📝 Entregable

- Funcionalidad completa.
- Código bien estructurado y comentado.
- Link del repositorio de github (verifiquen que la raíz de proyecto sea el código no una carpeta)