

Preparcial (20%) – CRUD con NextJs

Objetivo general

El objetivo de este taller es integrar conocimientos fundamentales y avanzados de React para diseñar una aplicación web que permita gestionar información mediante un CRUD (Crear, Leer, Actualizar y Eliminar), haciendo uso de:

- Manejo de estado con useState
- Manejo de eventos y formularios controlados
- Navegación entre páginas
- Obtener datos usando useEffect

Preparación del Preparcial

- Realice un fork del siguiente repositorio:
<https://github.com/isis3710-uniandes/bookstore-back>
- Realice el build de la imagen docker usando el comando:
`docker build ./ -t bookstore`
- Ejecute la imagen usando el comando:
`docker run -d -p 127.0.0.1:8080:8080 bookstore`
- Verifique que el API quedo funcioando ingresando a la siguiente url:
<http://127.0.0.1:8080/api/authors>

Enunciado del reto

El parcial podria incluir validaciones locales con ReactHook Form y validaciones globales con Zod.

Diseñar y desarrollar una aplicación web en NextJS que implemente un CRUD de usuarios con las siguientes funcionalidades:

Tendrán que generar interfaces gráficas definidas por nosotros haciendo uso de React con el framework Next.js y estilos de Tailwind CSS.

- Listar usuarios existentes en la plataforma (/api/authors)
- Crear nuevos usuarios mediante un formulario controlado
- Editar la información de un usuario existente
- Eliminar usuarios de la lista
- Navegar entre diferentes páginas con React Router (por ejemplo: Author, Crear Author) El routing que deben implementar es con el AppRouter de Next.js y no con react-router.

Requisitos mínimos

El código debe estar fuertemente tipado, es decir, las extensiones de los archivos deben ser .ts (TypeScript) y .tsx (Componentes de React con TypeScript), no JSX como se menciona en el enunciado.

- Uso de JSX y useState para manejar datos locales
- Implementación de eventos y formularios controlados
- Navegación entre vistas utilizando React Router
- Creación de un custom hook para la lógica del CRUD

El routing que deben implementar es con el AppRouter de Next.js y no con react-router.

El parcial podría incluir gestión de estado local con useState y gestión de estado global con Zustand.

Puntos a desarrollar

Paso 1 – Lista de autores

1. Crear un componente que muestre una lista de autores
2. Use un hook useEffect para obtener los datos del api (/api/authors)
3. Renderizar la lista en la página Autores.

Paso 3 – Formulario para crear usuarios

1. Diseñar un formulario controlado con los campos: birthDate, description, image, name.
2. Manejar el estado de cada campo con useState.
3. Al enviar el formulario, agregar el nuevo usuario a la lista.
4. Ubicar este formulario en la página Crear Usuario.

Paso 4 – Configuración de rutas

1. Definir al menos dos rutas:
 - /authors → Lista de autores
 - /crear → Formulario de creación de autor

Paso 5 – Conectar formulario y lista

1. Verificar que, al crear un autor desde el formulario, este se muestre en la lista.
2. Asegurar que los datos persistan en el estado mientras navegas entre rutas.

Paso 6 – Editar autores

1. Agregar un botón de Editar en cada autor de la lista.
2. Al seleccionarlo, mostrar un formulario con los datos del autor cargado.
3. Al guardar los cambios, actualizar el autor en la lista.

Paso 7 – Eliminar autor

1. Agregar un botón de Eliminar en cada autor de la lista.
2. Al hacer clic, remover ese autor del estado y actualizar la vista.

Entrega

La entrega de este preparcial se realizará por medio de bloqueneon, se debe enviar un link a un repositorio público que contenga la solución. El uso de herramientas de ia generativa no esta permitido y los profesores se reservan el derecho de solicitar una sustentación de la solución entregada.