Consigna de Proyecto: CRUD de Unicornios con React + PrimeReact

👫 Trabajo en Grupos

- Este proyecto se realizará en grupos de hasta 4 integrantes.
- Cada grupo deberá crear un repositorio en GitHub y subir el proyecto completo allí.

Tecnologías a Utilizar

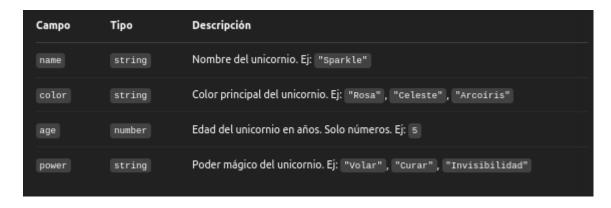
- Vite + React como entorno y framework principal.
- PrimeReact para componentes visuales y presentación de la tabla de unicornios.
- React Router DOM para navegación por rutas.
- API externa: https://crudcrud.com (requiere generar un endpoint para cada grupo).
- Hooks de React: Uso obligatorio de useState y useEffect.

🦄 Estructura del Objeto Unicorn

Cada unicornio deberá tener la siguiente estructura al momento de ser creado:

```
{
  name: "Nombre del unicornio",
  color: "Color principal",
  age: 5,
  power: "Poder especial"
}
```

* Descripción de cada campo:



No incluyas el campo _id, ya que es generado automáticamente por la API al crear el unicornio.

Estructura del Proyecto

Dentro del directorio src, deberán crear una **carpeta llamada unicorns/** con la siguiente estructura:

src/
unicorns/
index.jsx # Punto de entrada, donde se enruta o se exporta el contenedor
UnicornsContainer.jsx # Lógica del CRUD: llamadas a la API, estados, handlers
UnicornsView.jsx # Componente de presentación: renderiza la tabla, botones, formularios

🔄 Funcionalidad CRUD

Deberán implementar un CRUD completo sobre unicornios:

- Create: Formulario para crear un unicornio. Usar componentes de PrimeReact.
- **Read:** Mostrar la lista de unicornios en una tabla de PrimeReact.
- Update: Permitir editar un unicornio directamente desde la tabla o mediante un formulario.
- Delete: Botón para eliminar unicornios.

Mavegación

Utilizando **react-router-dom**, deberán implementar una ruta accesible desde el navegador:

- /unicornios → Vista principal del módulo de unicornios (UnicornsView).
- El uso correcto de rutas será un punto adicional en la calificación.

Requisitos Técnicos

- Uso de useEffect para obtener datos desde la API cuando se carga la vista.
- Manejo de estado mediante useState.
- Los datos y handlers deben pasarse por props desde UnicornsContainer hacia UnicornsView.
- El código debe estar **modularizado**, **limpio y documentado** donde sea necesario.
- Subida del código al repositorio de GitHub correspondiente y compartir el enlace al docente.

Extras (Puntos Adicionales)

- Implementación de navegación con react-router-dom.
- Validaciones de formularios.
- Estilizado adicional o mejoras en la experiencia del usuario.
- Buen manejo de errores de API.