

Uso de Estructuras de Datos en el Proyecto "Gestión de Estudiantes"

Este documento describe cómo se utilizaron las estructuras de datos recomendadas en el proyecto desarrollado en React con JavaScript, enfocado en la gestión de información de estudiantes sin base de datos y con almacenamiento en localStorage.

1. Listas

Se utilizaron listas (arrays en JavaScript) para almacenar el conjunto completo de estudiantes. Cada estudiante está representado como un objeto dentro de este array. React mantiene esta lista en memoria usando el hook useState.

```
1  const [estudiantes, setEstudiantes] = useState([]);
```

2. Diccionarios (Objetos)

Cada estudiante está representado como un objeto JavaScript (equivalente a un diccionario en otros lenguajes), con las propiedades: nombre, carne, materias (array de tuplas) y promedio.

```
1  {  
2    nombre: "Carlos Pérez",  
3    carne: "20230002",  
4    materias: [["Programación I", "PRG101"], ["Matemática I", "MAT101"]],  
5    promedio: 7.8  
6  },
```

3. Tuplas (simuladas con arrays fijos)

JavaScript no tiene tuplas nativas, por lo que se simularon mediante arreglos de longitud fija. Cada materia inscrita por un estudiante se representa como un array con dos valores: [nombreMateria, codigoMateria]. Esto asegura que la información se mantenga inmutable y clara.

```
1  materias: [ ["Programación I", "PRG101"], ["Matemática I", "MAT101"] ],
```

4. Sets

Se utilizó la estructura de datos Set para verificar duplicados de carné. Antes de agregar un nuevo estudiante, se construye un Set con todos los carnés existentes y se verifica si el nuevo carné ya está en uso. Esto asegura la unicidad de cada estudiante en el sistema.

Gracias a la combinación adecuada de estas estructuras, el sistema mantiene una lógica clara, eficiente y fácil de mantener. Además, se cumplen todos los requisitos funcionales propuestos en el ejercicio.

```
1  const agregarEstudiante = (nuevo) => {
2      const carnesSet = new Set(estudiantes.map(e => e.carne));
3
4      if (carnesSet.has(nuevo.carne)) {
5          alert(`Ya existe un estudiante con el carné ${nuevo.carne}`);
6          return;
7      }
8
9      const nuevos = [...estudiantes, nuevo];
10     setEstudiantes(nuevos);
11     guardarEstudiantes(nuevos);
12 };
```