

### Etapa 3. (Aplica Parcial 2)

Se requiere agregar las siguientes funcionalidades al Sistema de Gestion de Cursos:

Hay puntos que son “opcionales” de hacer, pero que sería una buena oportunidad para desarrollarlos.

#### 1. Edición y eliminación de cursos y estudiantes:

- **Edición de cursos y estudiantes:** Permitir que el usuario edite los datos de un curso o estudiante ya existente (por ejemplo, cambiar el nombre del curso o actualizar la nota de un estudiante).
- **Eliminación de cursos y estudiantes:** Incluir un botón de "Eliminar" junto a cada curso y estudiante para que el usuario pueda eliminarlos si ya no son necesarios.

#### Implementación:

- **Edición:** Al hacer clic en un botón de "Editar", rellenar los campos del formulario con la información actual del curso o estudiante, y luego permitir que el usuario la actualice.
- **Eliminación:** Agregar un botón de eliminación que borre el curso o estudiante seleccionado del array y actualice la vista.

#### 2. Validaciones avanzadas en los formularios:

- Añadir **validaciones más complejas** en los formularios para asegurar que los datos ingresados sean correctos (por ejemplo, que la edad sea un número positivo o que las notas estén entre 0 y 10).
- Mostrar **mensajes de error** o advertencias si los datos son incorrectos antes de permitir agregar o editar cursos/estudiantes.

#### Implementación:

- Usar JavaScript para validar los datos en el momento del ingreso, y CSS para mostrar los errores junto a los campos que están incorrectos.

#### 3. Filtros y búsquedas dinámicas:

- Agregar un **campo de búsqueda** para buscar estudiantes por nombre o filtrar los cursos según el profesor a cargo.
- Permitir **ordenar a los estudiantes** por sus notas o por edad, para que sea más fácil visualizar la información relevante.

#### Implementación:

- Crear una función de búsqueda que filtre el array de estudiantes o cursos y muestre solo los resultados que coinciden con la búsqueda.

- Crear botones para ordenar (por ejemplo, de mayor a menor nota) y reorganizar la lista visual.

#### 4. Almacenamiento persistente con LocalStorage:

- Usar `LocalStorage` para que los datos (cursos y estudiantes) se guarden y se mantengan entre sesiones. Esto permitirá que si el usuario cierra la página, al abrirla de nuevo, los datos ingresados sigan estando disponibles.

##### Implementación:

- Guardar el estado actual de los cursos y estudiantes en `LocalStorage` cada vez que se agregue o edite un curso o estudiante, y cargar esos datos cuando la página se recargue.

#### 5. Diseño más avanzado con Bootstrap : (Opcional)

- Usa una **librería de CSS** como **Bootstrap** para mejorar la apariencia visual de la página, con botones estilizados, formularios con mejor formato, y una disposición más sofisticada del contenido.
- **Mejora la responsividad** de la página para que funcione bien en dispositivos móviles, asegurándote de que los formularios y listas se adapten correctamente.

##### Implementación:

- Puedes integrar componentes de Bootstrap como modales, alerts para mostrar errores o confirmaciones, y tarjetas para presentar los cursos y estudiantes de forma más estilizada.

#### 6. Estadísticas adicionales:

- Agregar una sección que muestre **estadísticas generales** del sistema, como el número total de estudiantes, el número de cursos, el promedio general de todos los estudiantes, y el curso con la mejor calificación promedio.

##### Implementación:

- Crear una función que recorra todos los cursos y estudiantes y calcule estadísticas como la cantidad de estudiantes o el promedio general, y luego mostrar esos datos de forma visual en la página.

#### 7. Interacción visual mejorada: (opcional)

- Añadir **animaciones** y transiciones suaves cuando se agregue o elimine un curso o estudiante, para hacer la experiencia más fluida.
- Usar **íconos visuales** con librerías como FontAwesome para mejorar la presentación (por ejemplo, un ícono de "lápiz" para editar o un ícono de "basura" para eliminar).

**Implementación:**

- Usar CSS para crear animaciones simples cuando los elementos se agreguen o eliminen, como transiciones al mostrar/ocultar estudiantes o cursos.

**8. Exportación de datos:**

- Permitir que el usuario pueda **exportar los datos** de cursos y estudiantes a un archivo JSON o CSV, para que pueda guardar la información de los cursos en su computadora.

**Implementación:**

- Usar JavaScript para convertir el array de cursos y estudiantes en un archivo JSON o CSV y luego ofrecer un botón de descarga.