

PROYECTO FINAL: Desarrollo de un CRUD Avanzado usando Arreglos y Funciones de Arreglos en JavaScript

1. Descripción General

Desarrollar una **aplicación web funcional** que implemente un CRUD (Create, Read, Update, Delete) utilizando **HTML, CSS y JavaScript con archivos separados**. El objetivo es aprender a manipular colecciones de datos mediante **arreglos de objetos**, aplicando las principales funciones modernas de JavaScript enseñadas en el curso.

La práctica simula un sistema de **gestión de estudiantes**, donde cada estudiante tiene atributos como nombre, edad, género, número de cursos inscritos y valor de pago. Además del CRUD, el sistema incorpora **búsquedas, filtros y estadísticas**, utilizando funciones avanzadas de los arreglos de JS.

2. Objetivos de Aprendizaje

Al realizar el proyecto, el estudiante será capaz de:

CRUD (Competencias Principales)

- Crear registros y agregarlos a un arreglo.
- Mostrar elementos del arreglo en una tabla HTML.
- Editar datos existentes dentro del arreglo.
- Eliminar elementos aplicando funciones propias de los arrays.

Manejo moderno de Arreglos

Aplicar correctamente las siguientes funciones del módulo:

- map()
- filter()
- reduce()
- some()
- every()
- find()
- findIndex()
- lastIndexOf()
- slice()
- splice()
- fill()
- at()
- Array.isArray()

DOM y Eventos

- Capturar datos desde formularios
- Pintar tablas dinámicamente
- Manejar eventos de clic y entrada de datos
- Manipular secciones de resultados desde JavaScript

3. Descripción Específica de la Aplicación

Tu aplicación web debe permitir:

A. Gestión completa de estudiantes (CRUD)

1. Crear

El usuario puede ingresar:

- nombre
- edad
- género
- número de cursos
- pago

Y al presionar **Guardar**, el estudiante se agrega al arreglo principal.

2. Leer / Listar

Los estudiantes deben visualizarse en una **tabla HTML dinámica**, actualizada en tiempo real.

3. Actualizar

Cada fila debe incluir un botón **Editar**.

Al presionarlo:

- Los datos se cargan en el formulario
- El usuario los modifica
- Se guardan en el arreglo

4. Eliminar

Cada estudiante debe tener un botón **Eliminar**, que lo remueva del arreglo usando `splice()`.

B. Filtros y búsquedas

La aplicación debe incluir:

Filtro por género

- Mostrar solo estudiantes femeninas (usar `filter`)
- Mostrar solo masculinos
- Mostrar todos nuevamente

Búsqueda por nombre

Buscar coincidencias parcial o totalmente (uso de `includes` + `filter`).

C. Estadísticas Avanzadas con Métodos de Arreglos

La aplicación debe generar estadísticas como:

Lista de nombres usando map()

Total acumulado de pagos usando reduce()

Validaciones con some() y every()

Obtener elementos con find() y at()

Extraer clones con slice()

Modificar otros arreglos con fill() y lastIndexOf()

Validar tipos con Array.isArray()

Los resultados deben mostrarse en un panel de texto debajo de la tabla.

4. Requisitos Técnicos

Archivos separados

- index.html
- styles.css
- app.js

Estructura mínima del HTML:

- Formulario para agregar/editar
- Botones de filtro y búsqueda
- Tabla con datos dinámicos
- Sección para estadísticas

Uso obligatorio de:

- Arreglos de objetos
- Métodos del módulo
- Funciones flecha
- Eventos de DOM
- Pintado dinámico de tabla

5. Criterios de Evaluación

Criterio	Puntos
CRUD completo funcionando	30 pts
Uso correcto de map, filter, reduce	15 pts
Uso de funciones complementarias (find, every, some, etc.)	15 pts
Buscador por nombre	10 pts
Filtro por género	10 pts

Criterio	Puntos
Estadísticas generadas correctamente	10 pts
Enlace adecuado de HTML con JS	10 pts
Total	100 pts

6. Productos Entregables

El estudiante deberá entregar:

1. Carpeta con:
 - index.html
 - styles.css
 - app.js
2. Captura o video corto mostrando:
 - Crear estudiantes
 - Editar
 - Eliminar
 - Buscar
 - Filtrar
 - Estadísticas funcionando

7. Indicaciones Finales

- El código debe ser escrito con buena indentación.
- Todas las funciones deben estar comentadas.
- No se permite usar frameworks (Vue, React, etc.).
- Toda la manipulación debe ser en **JavaScript puro**.
- La tabla debe actualizarse sin recargar la página (DOM dinámico).