



TP 1 JS Arrays

PROGRAMACIÓN III - 2025 2do cuatrimestre
TECNICATURA UNIVERSITARIA EN DESARROLLO WEB

Objetivos

Aplicar los conceptos de arrays en JavaScript, utilizando sus métodos comunes y avanzados, y ejecutar el código en el entorno Node.js para simular la gestión de un conjunto de datos.

Descripción

Los alumnos deberán crear un script de JavaScript que simule la gestión de una pequeña "base de datos" de elementos (por ejemplo, productos de una tienda, turnos de una casa de cumpleaños, o estudiantes de un curso) utilizando arrays. Deberán realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y otras transformaciones utilizando los métodos de arrays vistos en clase. El script deberá ser ejecutable con Node.js.

Condiciones de Entrega

- El Trabajo Práctico deberá ser:
 - Realizado en forma grupal.
 - Los grupos deberán contar con un mínimo de 4 y un máximo de 6 alumnos.
 - Cargado en la sección del Campus Virtual correspondiente, en un archivo comprimido en formato ZIP, RAR, TAR.GZ u otro formato similar. También se acepta subir archivo .TXT con URL a repositorio GitHub.
 - Deberá indicarse número de grupo, apellido y nombre de los integrantes.
 - Entregado antes de la fecha límite informada en el campus.
- Las soluciones deben ser de autoría propia. Aquellas que se detecten como idénticas entre diferentes grupos serán desaprobadas para todos los involucrados.
- Uso de Inteligencia Artificial (IA): Las herramientas de IA (como generadores de código o texto) pueden ser utilizadas únicamente como asistentes para el aprendizaje, depuración o para generar código repetitivo. No deben ser usadas para generar la solución completa.

Consigna

1. Configuración Inicial:

- 1) Instalar Node.js, ya que es el entorno de ejecución que utilizaremos.
- 2) Crear un nuevo archivo JavaScript (ej. gestionDatos.js).
- 3) Dentro de este archivo, declarar un array llamado **productos** (o libros, turnos, etc.) que contenga al menos 5 objetos JavaScript. Cada objeto debe representar un elemento con propiedades como id, nombre, precio y stock.

2. Operaciones Básicas y Acceso:

- 1 Imprimir la longitud total del array **productos**.
- 2 Acceder e imprimir por consola el nombre del segundo y cuarto elemento del array utilizando su índice.

3. Recorrido del Array:

1. Recorrer el array **productos** utilizando un bucle for...of e imprimir el nombre y el precio de cada elemento.
2. Recorre el array **productos** utilizando el método forEach() e imprimir la misma información que en el punto anterior, pero agregando una frase descriptiva (ej. "Producto: [nombre], Precio: [precio]").

4. Manipulación de Arrays:

1. Agregar dos elementos al final del array **productos** utilizando push().
2. Eliminar el último elemento del array **productos** utilizando pop().
3. Agregar un nuevo elemento al inicio del array **productos** utilizando unshift().
4. Elimina el primer elemento del array **productos** utilizando shift().
5. Crear un nuevo array llamado **productosConStock** que contenga solo los elementos del array **productos** donde el stock sea mayor que 0 utilizando filter().
6. Crear un nuevo array llamado **nombresProductos** que contenga solo los nombres de todos los **productos** en el inventario utilizando map().
7. Encontrar y guardar en una variable el primer producto en **productos** que tenga un id específico (ej. id:3) utilizando find(). Si no lo encuentra, indicar que no existe.
8. Crear un nuevo array llamado **productosOrdenados** que contenga los productos ordenados por precio en orden decreciente. (*investigar método sort()*).

Imprimir en consola el array original o creado para verificar las operaciones realizadas.