

# JavaScript 2024

---

## Práctica 3: Interactividad y Datos en Tiempo Real con JavaScript

---

Esta práctica tiene como propósito fortalecer la comprensión y habilidad en la manipulación de datos, interacción con APIs externas, y dinamismo en la web.

### Entrega Obligatoria

Junto a los demás integrantes de su grupo, implemente el ejercicio 7 y suba el código al repositorio git asignado por la cátedra.

### Consideraciones sobre la entrega

- Se debe realizar la entrega sobre la rama entrega-02 en el directorio practicas/practica-03.
- Se debe realizar en grupo.
- Se debe subir al repositorio git asignado al grupo en gitlab.
- Cada integrante del grupo debe hacer sus commits con su aporte a la solución.

### Objetivos

- Comprender y aplicar la serialización de objetos.
- Desarrollar habilidades en el uso de APIs externas con AJAX y de WebSockets.
- Practicar la manipulación del DOM para modificar elementos de la página web dinámicamente.

### Ejercicio 1: Explorando la Serialización de Objetos

Verifique qué imprimen estas sentencias.

```

const elements = [ 22, 44, 78, -43, 89 ]
const point1 = { x: 0.0, y: 0.0 }
const point2 = { x:3.0, y: 0.0 }
const point3 = { x:0.0, y: 3.0 }

let triangle = {
  corners : [point1, point2, point3],
}

console.log(JSON.stringify(elements))
console.log(JSON.stringify(point1))
console.log(JSON.stringify(triangle))

```

## Ejercicio 2: Calculando la Huella de Carbono de Viajes

Dada la siguiente asignación:

```

var jsonStr = '['
+ '{"name":"Alice","dob": "2001-03-04T00:00:00.000Z","trips": 15}'
+ '{"name":"Robert","dob": "1997-01-31T00:00:00.000Z","trips": 8}'
+ '{"name":"Charles","dob": "1978-10-15T00:00:00.000Z","trips": 12}'
+ '{"name":"Lucía","dob": "1955-08-07T00:00:00.000Z","trips": 20}'
+ '{"name":"Peter","dob": "1988-03-09T00:00:00.000Z","trips": 5}'
+ '{"name":"Lucas","dob": "1910-12-04T00:00:00.000Z","trips": 3}'
+ ']'

```

y la función `frequentTravelers` que sigue

```

function frequentTravelers(people){
  return people
    .filter(p => p.trips > 10)
    .map(p => p.name)
    .reduce((n1, n2) => n1 + ", " + n2)
}

```

1. Utilice el objeto `JSON` para invocar `frequentTravelers` e imprimir en la consola el resultado que debería ser el que sigue.

Alice, Charles, Lucía

Este resultado indica quiénes han viajado más de 10 veces, sugiriendo una mayor huella de carbono debido a la frecuencia de sus viajes.

2. Crear una función que imprima en la consola el nombre de la próxima persona en cumplir años.

### Ejercicio 3: Imágenes Gatunas en Rotación

Desarrolla habilidades en solicitudes a APIs y manejo del DOM para mostrar y actualizar imágenes de gatos automáticamente.

1. Use la función predefinida [Fetch](#) para hacer una página HTML que muestre una imagen de temática gatuna obtenida de <https://aws.random.cat/meow> o de <https://api.thecatapi.com/v1/images/search>.
2. Haga que la imagen cambie automáticamente cada 2 segundos

*Pista: Puede usar [setTimeout](#).*

### Ejercicio 4: Un Viaje por los Datos de los Países

Implemente una página HTML que liste al menos 5 datos de los países como el nombre, su población, la capital, el continente, etc. usando la API [REST Countries](#) y, por ejemplo, colocando la información como elementos `li` dentro de un `ul`.

*Pista: puede usar [createElement](#) y [appendChild](#).*

### Ejercicio 5: Filtrando Países por Nombre

Agregue a la página del ejercicio 4 un elemento `input` y un botón. Al presionar el botón, el listado de países debe mostrar únicamente aquellos cuyo nombre contiene el texto ingresado en el `input` o un mensaje si ninguno coincide.

*Pista: puede usar [removeChild](#).*

### Ejercicio 6: Clasificación Dinámica de Países

Basándose en los ejercicios 4 y 5, incorpore un elemento `select` que permita elegir un dato del país y ordene el listado por ese dato. Por ejemplo si se selecciona Nombre, el listado se debe ordenar por nombre y si se elige Población, por población.

### Ejercicio 7: Una Galaxia Lejana - Navegando por los Datos de Star Wars

Después de completar la práctica principal, puedes intentar estos desafíos sencillos para seguir practicando. Cada ejercicio debe estar en un archivo separado, en el mismo directorio `practicas/practica-03/`. Una vez

realizados subirlo al repositorio con el mensaje "Desafíos 03 completada por nombreAlumno".

### **Ejercicio adicional 1: Cronómetro Digital**

Implementa un cronómetro digital en una página web que permita iniciar, pausar y reiniciar el tiempo, utilizando setInterval y manipulación del DOM.

### **Ejercicio adicional 2: Generador de Contraseñas Aleatorias**

Crea una herramienta que genere contraseñas aleatorias basadas en criterios seleccionados por el usuario (longitud, inclusión de números, símbolos, etc.).

### **Ejercicio adicional 3: Pokédex Básica**

Desarrolla una aplicación web que funcione como una Pokédex básica, permitiendo a los usuarios buscar información sobre Pokémon por su nombre. La aplicación deberá tener un campo de búsqueda y un botón. Al apretar mostrar información relevante sobre el Pokémon buscado, como su imagen, tipo(s), habilidades y estadísticas básicas. Para eso pueden hacer uso de la [PokeApi](#).

## **Recursos Adicionales Esenciales**

Aquí hay algunos recursos fundamentales para ayudarte a avanzar:

- [Guía de la Fetch API](#)
- [Tutorial REST Countries API](#)
- [The Star Wars API](#)
- [PokeApi](#)