

TRABAJO PRÁCTICO NRO. 05 – JAVASCRIPT

- Trabajo Práctico de INTEGRACIÓN de CONCEPTOS – Trabajo Práctico Final

Especificaciones:

- El Alumno debe trabajar con un solo Proyecto Integrador. El nombre del Proyecto se debe denominar “TrabajoPracticoFinal_MaldonadoDaniel.rar”.
- Dentro debe tener un archivo de texto plano que debe estar en el directorio raíz del proyecto, allí consignarán todos sus datos actuales: **(por ejemplo)**

Apellidos: Maldonado

Nombres: Enrique Daniel

Dni: 22.333.444

Correo electrónico: ingdmaldonado@gmail.com

Telefono Celular: 3834402862 (sin guiones).

- Todos los Ejercicios deben tener su propio HTML, CSS (a gusto).

PUNTO 01: MVC Básico: Consumo de la API RestCountries

Conectarse a los siguientes endpoints de la API RestCountries:

- <https://restcountries.com/v3.1/region/africa>
- <https://restcountries.com/v3.1/region/americas>
- <https://restcountries.com/v3.1/region/asia>
- <https://restcountries.com/v3.1/region/europe>
- <https://restcountries.com/v3.1/region/oceania>

La tarea consiste en:

1. Seleccionar uno de estos endpoints.
2. Conectarse utilizando un proceso asíncrono con fetch.
3. Mostrar el resultado recuperado con console.log.

Extensión del ejercicio

Definir una función en formato arrow function que cumpla con lo siguiente:

- Reciba como parámetro la ruta a un endpoint (cualquiera de los anteriores).
- Se conecte a esa ruta usando fetch y un proceso asíncrono (async/await).
- Convierta la respuesta a formato JSON.
- Devuelva los datos a quien invoque la función.

Observaciones importantes sobre la organización (MVC básico)

- Modelo:
 - Carpeta: Modelo/
 - Archivo: funcionesDelModelo.js
 - Contiene toda la lógica de consultas a APIs y procesamiento de datos.
- Controlador:
 - Carpeta: Controlador/
 - Archivo: funcionesDelControlador.js
 - El Controlador solicita al Modelo los datos (es decir, llama a las funciones del Modelo).
 - Con el resultado que devuelven esas funciones, el Controlador se los pasa a la Vista.
- Vista:
 - Carpeta: Vista/
 - Archivo: funcionesDeLaVista.js
 - Se encarga de recibir los datos procesados desde el Controlador y renderizarlos en el DOM.

📄 Flujo del MVC:

1. El Controlador pide datos al Modelo.



2. El Modelo devuelve los datos.
3. El Controlador pasa esos datos a la Vista.
4. La Vista los renderiza en el DOM.

PUNTO 02: Destructuring de los datos Obtenidos

Con el array de países obtenido en el Punto 1, practique distintas formas de destructuring para acceder a los atributos de cada país y mostrar por consola la siguiente información:

Nombre → name.common

Capital → primer elemento de capital (si existe)

Población → population

Bandera → flags.png (imagen)

Inciso A: Recorra el array de países con `forEach`. Dentro de la función callback, utilice destructuring para extraer los atributos solicitados. Muestre los datos por consola en una sola línea por país.

Inciso B: Recorra el array de países con `forEach`. Dentro de la función callback, realice destructuring de los atributos usando alias para renombrar las variables. Muestre los datos por consola en una sola línea por país.

Inciso C: Recorra el array de países con `forEach`. En lugar de hacer destructuring dentro del cuerpo, impleméntelo directamente en los parámetros de la función callback. Muestre los datos por consola en una sola línea por país.

Inciso D: Recorra el array de países con `forEach`. En la función callback, practique destructuring de arrays utilizando el atributo `capital`. Obtenga la primera capital mediante destructuring y asigne un valor por defecto en caso de que no exista. Muestre los datos por consola en una sola línea por país.



PUNTO 03: Spread Operator

Con los datos obtenidos de la API de países, practique el uso del Spread Operator (...) para trabajar con arrays y objetos.

Inciso E: Combinar regiones, obtenga el listado de países de dos regiones distintas (por ejemplo, América y Europa) y combínelos en un solo array utilizando el Spread Operator. Muestre por consola la cantidad total de países combinados.

Inciso F: Crear un objeto simplificado, seleccione un país del array y, utilizando destructuring para extraer atributos y el Spread Operator para construir un nuevo objeto, genere una versión simplificada que contenga únicamente:

Nombre → name.common

Capital → primer elemento de capital (si existe)

Población → population

Bandera → flags.png (imagen)

Muestre este objeto simplificado por consola.

PUNTO 04: Renderizado:

A partir del resultado de la consulta a la API (array de países) obtenida en el Punto 1, renderice esos datos en el DOM. El Controlador recibe el array de países y se lo pasa a la Vista La Vista se encarga de dibujar/renderizar en el DOM.

Renderizar en una tabla HTML (o, alternativamente, en contenedores con flex), mostrando como mínimo estos atributos por país:

Nombre → name.common

Capital → primer elemento de capital (si existe)

Población → population

Bandera → flags.png (imagen)