

EXAMEN TEMAS 9 Y 11: JQUERY Y FETCH

EJERCICIO 1 (4 PUNTOS)

Aplicación: Herramienta de Gestión de Inventario de Videojuegos

Esta es una aplicación que permite a una tienda gestionar su inventario de videojuegos. Cada videojuego tiene las siguientes propiedades:

- Título (titulo)
- Plataforma (plataforma) (PlayStation, Xbox, PC, Switch)
- Género (genero) (Acción, Aventura, Deportes, RPG, etc.)
- Nivel de popularidad (1 a 5) (nivelPopularidad)

1. Página principal (index.html) (0 puntos)

Debe incluir un menú con las opciones:

- Insertar un nuevo videojuego (enlace a insertar.html).
- Listar videojuegos (enlace a listar.html).

La página no debe tener CSS ni JavaScript asociado.

2. Página de inserción (insertar.html + insertar.js) (1 punto)

Crear un array vacío videojuegos.

Llamar a una función `rellenaVideojuegos()` para precargar datos desde una API.

Formulario con los campos: **(0.25 puntos)**

- Título del videojuego (input de texto).
- Plataforma (campo select).
- Género (campo select).
- Nivel de popularidad (campo select del 1 al 5).

Botón Insertar que: **(0.75 puntos)**

- Cree un objeto nuevo con los datos del formulario.
- Inserte el objeto en el array videojuegos.
- Llame a una función `insertaVideojuego()` para enviar el array actualizado a la API.

3. Página de listado (listar.html + listar.js) (1 punto)

- Zona div para mostrar la lista de videojuegos.
- Enlace para volver al inicio.
- Crear un array vacío videojuegos.
- Llamar a una función `rellenaVideojuegos()` que cargue los datos desde la API.
- Mostrar cada videojuego con: Título, plataforma, género y nivel de popularidad.

Funciones requeridas:

1. Función `rellenaVideojuegos()` (1 punto)

- Implementar con `fetch` o `async/await`.
- URL para obtener los datos:

- La función debe devolver un array con los datos.

2. Función insertaVideojuego() (1 punto)

- Implementar con fetch o async/await y el método PUT.
- Enviar el array videojuegos convertido a JSON (JSON.stringify) a la misma URL.
- No devuelve nada.

EJERCICIO 2 (1.5 PUNTOS)

Hay una API que esta graciosa (GET), llamada: **(0.5 puntos)**

https://api.imgflip.com/get_memes

La cual nos devuelve un fichero en formato “.json” con muchos **memes** para poder visualizar.

Controlar la respuesta que devuelve esta página y hacer una zona en nuestra web que muestre una tarjeta (puede ser similar a la del ejercicio 2) con los siguientes datos: **(1 punto)**

- Nombre (h2)
- Imagen (img)

EJERCICIO 3 (2 PUNTOS)

Información sobre libros con la API de Google Books (GET)

Usa la API de Google Books para buscar información sobre libros basados en un término ingresado por el usuario. La petición se hará a esta URL: “<https://www.googleapis.com/books/v1/volumes>”

A esta URL hay que adjuntarle un parámetro “q” con el libro que busquemos.

En cada resultado, muestra los siguientes datos en un contenedor <div>: **(0.75 puntos)**

- Título del libro (etiqueta <h2>, campo **volumeInfo.title**)
- Autor(es) (etiqueta <h3>, campo **volumeInfo.authors**, concatenados si hay varios)
- Fecha de publicación (etiqueta <h4>, campo **volumeInfo.publishedDate**)
- Imagen de la portada (etiqueta , campo **volumeInfo.imageLinks.thumbnail**)
- Enlace a más información (etiqueta <a>, campo **volumeInfo.infoLink**)

Además:

1. Implementa una barra de búsqueda en la página HTML donde el usuario pueda escribir el término de búsqueda. **(0.25 puntos)**
2. Añade un botón para filtrar resultados por año de publicación (p. ej., solo libros publicados después del año 2000). **(1 punto)**

EJERCICIO 4 (2.5 PUNTOS)

Como hace tiempo que no jugamos vamos a darle a una API de videojuegos. **(0.5 puntos)**

La url <https://api.rawg.io/api/games>

La key: 2f4e8a1147e74fc1901bc08e5806c6d6

Mostrar (de la manera menos cutre que os de tiempo), lo siguiente de cada juego (que hay chorromil en la lista que obtendremos): **(2 puntos)**

- Nombre (**name**)
- Fecha de lanzamiento (**released**)
- Fondo del juego (**background_image**) Oh yeah!!!
- Calificación (**rating**)
- Un array de comentarios (**ratings**), de estos comentarios mostrar a su vez
 - Título (**title**)
 - Porcentaje (**percent**)
- Puntuación en metacritic (**metacritic**)
- Una array de plataformas donde ha salido (**platforms**), de este mostrar
 - Nombre (**name**)
 - Imagen de fondo de esa plataforma (**image_background**)

Pues eso, a los mandos