

Exercicio 58: Galería filtrable de portadas de discos de Scorpions

Obxectivo: Crear unha galería interactiva que permita filtrar portadas de discos de Scorpions por categorías: *albums*, *compilacións* e *DVD/Blu-ray*.

Pasos a seguir:

Para realizar esta páxina web necesitarás as seguintes estruturas e elementos:

1. HTML (index.html):

- Crea a seguinte estrutura no `<main>` para contener o sistema de filtrado, na cal usaremos un **atributo personalizado** que neste caso decidimos chamar ***data-filter***, os atributos personalizados sempre comezan co prefixo *data*, este tipo de atributos son moi útiles para almacenar información adicional en elementos HTML:

```
<div class="filter-container">
    <button class="filter-btn active" data-filter="all">Todos</button>
    <button class="filter-btn" data-filter="albums">Albums</button>
    <button class="filter-btn" data-filter="compilation">Compilacións</button>
    <button class="filter-btn" data-filter="dvd">DVD/Blu-ray</button>
</div>

<div class="gallery-container" id="gallery">
    <!-- As portadas mostraranse aquí e serán construídas dende JavaScript -->
</div>
```

2. CSS (style.css):

- **Crear a folla de estilos:** a continuación presentamos a proposta de deseño para esta páxina. Podes consultar o deseño en **Figma** na seguinte ligazón: [Mockup da páxina](#).
- Na carpeta *assets* atoparás os recursos restantes para completar o deseño como as **fontes tipográficas e as imaxes dos discos, etc.**

3. JavaScript (script.js):

- Crea un ficheiro *script.js* cun **array** que conteña as **portadas e as súas categorías**. **Estrutura de cada obxecto:**
 - **src:** URL da portada
 - **category:** Categoría (albums, compilation ou dvd)
 - **title:** Título do álbum

Exemplo:

```
const albums = [ { src: "./assets/caratulas/albums/Blackout.jpg", category: "albums", title: "Blackout" },
  { src: "./assets/caratulas/DVD/MTV-Unplugged-Live-In-Athens.jpg ", category: "dvd", title: "MTV Unplugged Live in Athens" },
  { src: "./assets/caratulas/compilations/BEST.jpg", category: "compilations", title: "Scorpions Best" },
  //Engadir o resto de portadas aquí seguindo esta estrutura
];
```

- Creamos a función para crear elementos de HTML usando os datos que temos almacenados no array *albums* que acabamos de crear.
 - Crea un `<div>` novo
 - Engádelle unha clase CSS combinada (`album-item` + categoría)
 - Insire dentro:
 - Imaxe coa portada
 - Párrafo co título
 - Devolve o elemento creado

```
const gallery = document.getElementById('gallery');
const filterButtons = document.querySelectorAll('.filter-btn');

// Función para crear elementos de álbum
function createAlbumElement(album) {
  const albumElement = document.createElement('div');
  albumElement.className = `album-item ${album.category}`;
  albumElement.innerHTML = `
  <p>${album.title}</p>`;

  return albumElement; }

```

- Creamos unha función para mostrar álbumes segundo o filtro. A lóxica desta é a seguinte:
 - Baleira a galería actual
 - Percorre todos os álbums
 - Filtra segundo:
 - Mostrar **todos** (`filter === 'all'`)
 - Ou só a categoría seleccionada
 - Crea e engade os elementos filtrados

```
function displayAlbums(filter = 'all') {
  gallery.innerHTML = '';
  albums.forEach(album => {
    if (filter === 'all' || album.category === filter) {
      gallery.appendChild(createAlbumElement(album));
    }
  });
}

```

- Creamos un evento para os botóns de filtro. O comportamento deste é o seguinte para cada botón de filtro:
 - Cando se fai clic:
 - Elimina a clase *active* de todos os botóns
 - Engade *active* ao botón pulsado
 - Actualiza a galería co novo filtro (usando o atributo *data-filter*)

```
filterButtons.forEach(button => { button.addEventListener('click', () => {
  filterButtons.forEach(btn => btn.classList.remove('active'));

```

```
button.classList.add('active'); displayAlbums(button.getAttribute('data-filter'));  
});  
});
```

- Por último invocamos a función *displayAlbums()* para mostrar tódolos álbumes ao cargar a páxina

```
displayAlbums();
```

Entrega: Unha vez completado todo **subiremos a carpeta *Exercicio58*** ao espazo para entrega de exercicios que temos no OneDrive de Microsoft Teams.

Ejercicio 58: Galería filtrable de portadas de discos de Scorpions

Objetivo:

Crear una galería interactiva que permita filtrar portadas de discos de Scorpions por categorías: álbumes, compilaciones y DVD/Blu-ray.

Pasos a seguir:

Para realizar esta página web necesitarás las siguientes estructuras y elementos:

1. HTML (index.html):

- Crea la siguiente estructura en el elemento `<main>` para contener el sistema de filtrado, en la cual usaremos un **atributo personalizado** que en este caso decidimos llamar ***data-filter***, los atributos personalizados siempre comienzan con el prefijo *data*, este tipo de atributos son muy útiles para almacenar información adicional en elementos HTML:

```
<div class="filter-container">
  <button class="filter-btn active" data-filter="all">Todos</button>
  <button class="filter-btn" data-filter="albums">Álbumes</button>
  <button class="filter-btn" data-filter="compilation">Compilaciones</button>
  <button class="filter-btn" data-filter="dvd">DVD/Blu-ray</button>
</div>

<div class="gallery-container" id="gallery">
  <!-- Las portadas se mostrarán aquí y se construirán desde JavaScript -->
</div>
```

2. CSS (style.css):

- **Crea la hoja de estilos.** A continuación, presentamos una propuesta de diseño para esta página. Puedes consultar el diseño en Figma en el siguiente enlace: [Mockup de la página.](#)
- En la carpeta **assets** encontrarás los recursos restantes para completar el diseño, como las **fuentes tipográficas y las imágenes de los discos**, etc.

3. JavaScript (script.js):

1. **Crea un archivo script.js con un array que contenga las portadas y sus categorías.**

Estructura de cada objeto:

- **src:** URL de la portada
- **category:** Categoría (albums, compilation o dvd)
- **title:** Título del álbum

Ejemplo:

```
const albums = [
  { src: "./assets/caratulas/albums/Blackout.jpg", category: "albums", title: "Blackout" },
  { src: "./assets/caratulas/DVD/MTV-Unplugged-Live-In-Athens.jpg", category: "dvd",
    title: "MTV Unplugged Live in Athens" },
  { src: "./assets/caratulas/compilations/BEST.jpg", category: "compilations", title:
    "Scorpions Best" },
  // Agrega el resto de portadas aquí siguiendo esta estructura
];
```

2. **Crea una función para generar elementos HTML usando los datos almacenados en el array albums.**

Pasos:

- Crea un nuevo <div>.
- Añádele una clase CSS combinada (album-item + categoría).
- Inserta dentro:
 - Una imagen con la portada.
 - Un párrafo con el título.
- Devuelve el elemento creado.

```
const gallery = document.getElementById('gallery');
const filterButtons = document.querySelectorAll('.filter-btn');
```

```
// Función para crear elementos de álbum
function createAlbumElement(album) {
  const albumElement = document.createElement('div');
  albumElement.className = `album-item ${album.category}`;
  albumElement.innerHTML = `
    
    <p>${album.title}</p>
  `;
  return albumElement;
}
```

3. **Crea una función para mostrar álbumes según el filtro seleccionado.**

Lógica:

- Vacía la galería actual.
- Recorre todos los álbumes.
- Filtra según:
 - Mostrar todos (filter === 'all').
 - O solo la categoría seleccionada.
- Crea y añade los elementos filtrados.

```
function displayAlbums(filter = 'all') {
  gallery.innerHTML = '';
  albums.forEach(album => {
    if (filter === 'all' || album.category === filter) {
      gallery.appendChild(createAlbumElement(album));
    }
  });
}
```

4. **Crea un evento para los botones de filtro.**

Comportamiento:

- Cuando se hace clic en un botón:
 - Elimina la clase active de todos los botones.
 - Añade la clase active al botón pulsado.
 - Actualiza la galería con el nuevo filtro (usando el atributo data-filter).

```
filterButtons.forEach(button => {  
  button.addEventListener('click', () => {  
    filterButtons.forEach(btn => btn.classList.remove('active'));  
    button.classList.add('active');  
    displayAlbums(button.getAttribute('data-filter'));  
  });  
});
```

5. **Por último, invoca la función `displayAlbums()` para mostrar todos los álbumes al cargar la página.**

```
displayAlbums();
```

Entrega:

Una vez completado todo, sube la carpeta **Ejercicio58** al espacio para entrega de ejercicios que tenemos en OneDrive de Microsoft Teams.