

PRÁCTICA 5.02 Enciclopedia básica

Normas de entrega

- En cuanto al **código**:
 - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un **if** un **for** pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
 - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
 - todo el código debe estar situado dentro del evento `window.onload = () => {};` o a través del evento `document.addEventListener("DOMContentLoaded", () => {});`,
 - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador `console.log(información)`,
 - los ejercicios deben realizarse usando **JavaScript ES6** y usar el modo estricto (**use strict**) No se podrá utilizar *jQuery* ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
 - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase*,
 - para la asignación de eventos se utilizará `addEventListener()` indicando sus tres parámetros en su definición,
 - debes dividir tu código en **bibliotecas temáticas** de funciones y/o clases a partir de este ejercicio,
 - se usarán las funcionalidades **import** y **export** para crear bibliotecas de funciones temáticas a partir de esta práctica,
 - todo el código que sea susceptible de retrasar su ejecución debe escribirse de forma asíncrona utilizando asincronismo,
 - todas las funciones asíncronas deben tener control de errores.
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
 - todos los ejercicios en **una carpeta** (creando las **subcarpetas** necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente,
 - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
 - el código contendrá ejemplos de ejecución, si procede, y
 - la carpeta será **comprimida** en formato **ZIP** y será subida a **Aules** de forma puntual.

Ejercicio 1 - Enciclopedia de *Star Wars*

Crea una aplicación que muestre información sobre *Star Wars*. Utiliza el **API REST** que ofrece **The Star Wars API** en <https://swapi.dev> (<https://swapi.dev/api/films/>) para hacer una aplicación con las siguientes características:

- al cargar la aplicación se mostrará un listado (``) con las películas que conforman la saga (**id** y **título**),
- al pulsar sobre el título de una de ellas, se mostrará su sinopsis y una ficha completa con su director, productor y fecha de lanzamiento (en formato europeo),
- aplica algo de tu estilo personal a través de **CSS**,
- ten en cuenta uso de import/export.
- `addEventListener` en funciones (en lugar de `window.load`).
- usa control de errores en los `fetch` (`catch`).
- la aplicación puede tener este aspecto (es una recomendación):

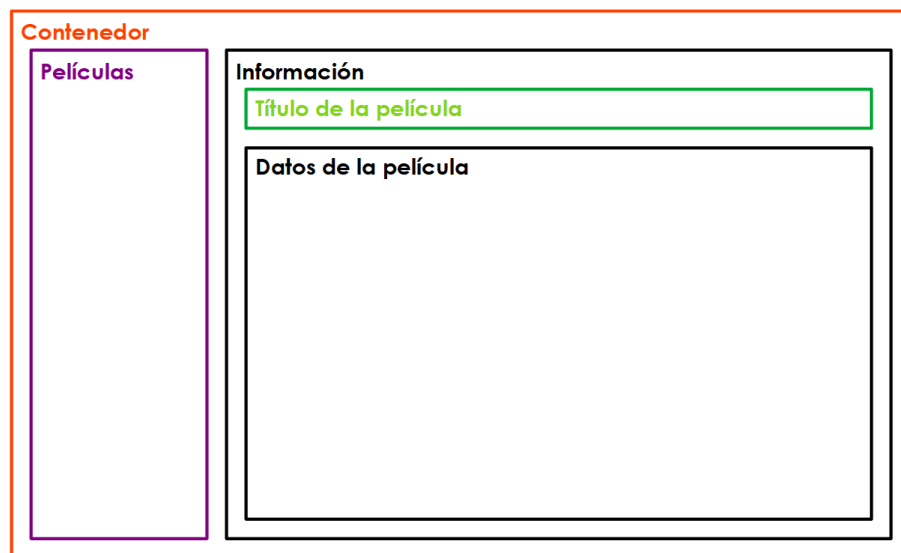


Figura 1: Aspecto general de la enciclopedia de *Star Wars*.

* (Reflexión) En este caso la api nos devuelve toda la información que necesita, imagina que sólo nos da el nombre de la película y una url a los contenidos de cada película. Piensa cómo lo resolverías para no demorar en tiempo de carga.