

TRABAJO PRÁCTICO INTEGRADOR

ETAPA #3 - JS

TUP - Laboratorio de Computación II

Etapas #3 - JavaScript

Continuando con el desarrollo de nuestra aplicación Web de Cine. En esta etapa del TP Integrador, tenemos como objetivo realizar las implementaciones de lógica JavaScript para lograr que la aplicación sea funcional. En resumen, vamos a hacer que la App muestre la cartelera de cine con las últimas películas o las películas más vistas, permita seleccionar las favoritas y muestre más información de estas al usuario final.

Para esta entrega, se presentan requerimientos EXIGIDOS los cuales se deberán implementar sí o sí en su programa para poder aprobar el TP. Al mismo tiempo, tendrán un listado de requerimientos EXTRAS los cuales son opcionales.

Consideraciones importantes:

1. Antes de avanzar con esta etapa, tener la entrega de la Etapa #2 corregida por su profesor y haber realizado los ajustes que se le han indicado vía email.
2. Realizar los cambios en el código HTML que requiera o considere necesarios.
3. Enlazar los archivos .js correspondientes a cada página. Por ejemplo, en el documento favorites.html agregar el common.js y el favorites.js. No te olvides de colocarle el “defer”-
4. Implementar lógica JS a cada página para lograr los requerimientos exigidos en las siguientes secciones. En caso de detectar que una funcionalidad se repite en diferentes páginas colocarla dentro del common.js, sino trabajar en el js correspondiente a cada página.
5. Finalizar todos los requerimientos EXIGIDOS y una vez cerrados avanzar, en caso de que cuente con el tiempo, con los requerimientos EXTRAS.
6. Los registros seleccionados por el usuario deben persistirse mas allá de la sesión, seleccionar el tipo de storage a utilizar.

Recomendación: Subir los cambios al repo a medida que se va avanzando con el desarrollo.

Conceptos JavaScript a poner en práctica:

- Manipulación del DOM
- Manejo de objetos y arreglos
- Local Storage
- Fetch API





Requerimientos Exigidos

Se propone avanzar el desarrollo de JS primero con la pantalla de índice de cartelera y luego con Favoritos. De esta manera se presentan los apartados de esta sección, de todas maneras, está en el desarrollador comenzar la implementación por donde crea más conveniente.

Cartelera de Cine (index.html)

En esta pagina inicial el usuario podrá ver el listado de películas disponibles en cartelera y seleccionar cuál es su favorita.

La grilla (grid) mostrara 4 películas por línea, y se visualizara el poster, el Titulo, el Código, Título original, y Año de la película.

			
Super Mario Bros. La película Código: 502354 Título original: The Super Mario Bros. Movie Idioma original: en Año: 2023-04-05	Rápidos y furiosos X Código: 385687 Título original: Fast X Idioma original: en Año: 2023-05-17	Evil Dead: El despertar Código: 713704 Título original: Evil Dead Rise Idioma original: en Año: 2023-04-13	Sisu Código: 840324 Título original: Sisu Idioma original: es Año: 2023-01-27
Agregar a Favoritos	Agregar a Favoritos	Agregar a Favoritos	Agregar a Favoritos

En este punto, tenemos desarrollado sólo la maqueta del escenario principal y ahora le daremos vida o comportamiento a través de JS. Por este motivo, se deberán hacer ajustes HTML para que la pantalla, cuando ingresa el usuario se encuentre vacía y se cargue a medida que se obtenga la información de la API de películas.

Consideraciones técnicas:

Consultar Cartelera de Cine a través de una API

La cartelera de cine será consultada a través de solicitudes a una API externa utilizando la herramienta [fetch](#).

Utilizar async/await o promesas para las llamadas realizadas con fetch.

La API donde deben consultar esta información es [The Movie Database API](#).

En este apartado podemos consultar la documentación de la misa y hacer pruebas de los distintos métodos:

<https://developer.themoviedb.org/docs>

Vamos a utilizar dos posibilidades, cada uno puede elegir la que mas le guste:

- Películas actualmente en cartelera

<https://developer.themoviedb.org/reference/movie-now-playing-list>

- Películas mas populares

<https://developer.themoviedb.org/reference/movie-popular-list>

El formato de la url es igual en ambos casos, varia el método llamado:

```
https://api.themoviedb.org/3/movie/popular?api_key={API KEY}&language={LENGUAJE}&page={NRO PAGINA}
```

Importante: obtener la clave de la API logueandose en la plataforma (la misma es gratuita), en el siguiente apartado se explica cómo obtenerla.

La respuesta que obtendremos vendrá en formato de objeto JSON. Este es un ejemplo de la misma:

```
{
  "dates": {
    "maximum": "2023-06-02",
    "minimum": "2023-04-15"
  },
  "page": 1,
  "results": Array(20)
}

[Prototype]: Object
1: {adult: false, backdrop_path: "/u4SDPKnBWEEOczPJvL0NwApkQSR.jpg", genre_ids: Array(5), id: 502356, original_language: "en", original_title: "The Super Mario Bros. Movie", overview: "Mientras trabaja bajo tierra para arreglar una tubería de agua, los plomeros de Brooklyn Mario y su hermano Luigi son tra...", popularity: 6344.538, poster_path: "/zNKs1T0VZuJiVuhL5GSCNkGdx.jpg", release_date: "2023-04-05", title: "Super Mario Bros. La película", video: false, vote_average: 7.8, vote_count: 3578}
2: {adult: false, backdrop_path: "/aAgGrfBwna1nO4M2USxwFgK500t.jpg", genre_ids: Array(2), id: 713704, original_language: "en", ...}
3: {adult: false, backdrop_path: "/94TIUEhuwv8PhdIAEvSuwPlj5S.jpg", genre_ids: Array(2), id: 840326, original_language: "fi", ...}
4: {adult: false, backdrop_path: "/hJoMSAltRx5xv1AXvKyDdugjucJ.jpg", genre_ids: Array(4), id: 447277, original_language: "en", ...}
5: {adult: false, backdrop_path: "/5YZbUmjbMa3C1vSW1Wj3D6XGolb.jpg", genre_ids: Array(3), id: 447365, original_language: "en", ...}
6: {adult: false, backdrop_path: "/i1eghEBiC0gN4KnuuUGC2fNiX1f.jpg", genre_ids: Array(2), id: 552688, original_language: "en", ...}
7: {adult: false, backdrop_path: "/hiHGRbyTcbZoLSYYkO4QiCLYe34.jpg", genre_ids: Array(2), id: 758323, original_language: "en", ...}
8: {adult: false, backdrop_path: "/eTvN54pd83TrSE0z6wbsXE3ktCV.jpg", genre_ids: Array(3), id: 882569, original_language: "en", ...}
9: {adult: false, backdrop_path: "/qE1NES0sHVQcbzvGrTx7ccpGziz.jpg", genre_ids: Array(3), id: 842675, original_language: "zh", ...}
10: {adult: false, backdrop_path: "/wIb43R1Pf5iYeoBAYIfxfhpE8NI.jpg", genre_ids: Array(2), id: 1094319, original_language: "en", ...}
11: {adult: false, backdrop_path: "/ceYZCBfwbBwSpGJ6PapMVw5jqLG.jpg", genre_ids: Array(4), id: 916224, original_language: "ja", ...}
12: {adult: false, backdrop_path: "/T5xXoFqyc9jNXZiB4Sw0jwkwjw.jpg", genre_ids: Array(4), id: 605886, original_language: "en", ...}
```

El contenido de esta respuesta puede ser manipulado como un **objeto** en JavaScript. Para poder obtener el detalle de la película/películas consultadas debemos inspeccionar las propiedades de ese

objeto. La única propiedad sobre las cuales nos vamos a centrar va a ser Results, que es donde se van a mostrar todos los resultados para la página consultada.

El objetivo será recorrer los resultados de la consulta y armar el contenido de la cartelera de la películas utilizando las propiedades devueltas para armar la tarjeta de cada una de ellas:

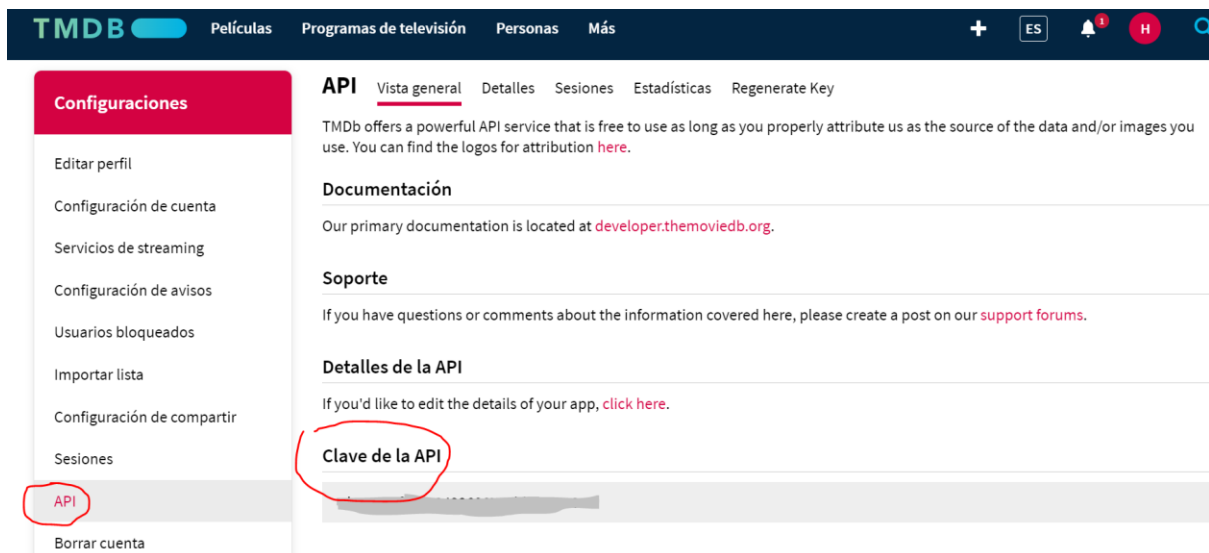
```
  
<h3 class="titulo">${title}</h3>  
<p><b>Código:</b> ${id}<br>  
<b>Título original:</b> ${original_title}<br>  
<b>Idioma original:</b> ${original_language}<br>  
<b>Año:</b> ${release_date}<br>
```



Como obtener la API KEY

Registrarse en el portal con datos mínimos, (no es requerida información fidedigna como el teléfono).

Una vez registrados ir a Configuraciones desde el menú desplegable a través del icono del usuario. Y luego al apartado API. Allí encontraran la Clave de API que necesitan para poder usar el servicio.



Agregar películas a favoritos por Código

Si el usuario conoce el código de una película podrá agregarla en forma directa.

Agregar Película Favorita por Código:

Debemos validar que:

- El valor ingresado es numérico.
- la película ya no haya sido ingresada, en ese caso se debe mostrar un mensaje de error.

La Película ingresada ya se encuentra almacenada

- Luego de ingresar una película válida se debe mostrar un mensaje de éxito al usuario

Película agregada con éxito

- La película exista o hayamos tenido un problema en la consulta, caso contrario mostrar un mensaje de error

Error: La Película seleccionada no se encuentra en la API o se produjo un error al consultar

Una vez que se hayan validado los puntos anteriores se debe almacenar el valor del código de la película en el storage FAVORITOS

IMPORTANTE: Validar la existencia del Código y no repetir el mismo en el array

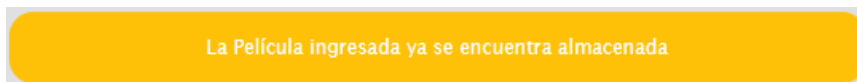
Agregar películas a favoritos

Desde el botón Agregar a Favoritos de cada tarjeta de película el usuario puede seleccionar la misma para que se le almacene en sus favoritos

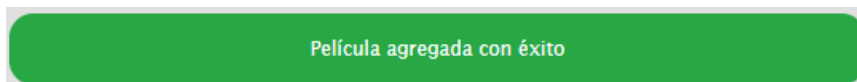


Debemos validar que:

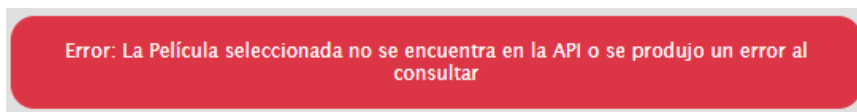
- La película ya no haya sido ingresada, en ese caso se debe mostrar un mensaje de error.



- Luego de ingresar una película válida se debe mostrar un mensaje de éxito al usuario



- Tuvimos un inconveniente en la consulta, caso contrario mostrar un mensaje de error



Una vez que se hayan validado los puntos anteriores se debe almacenar el valor del código de la película en el storage FAVORITOS

IMPORTANTE: Validar la existencia del Código y no repetir el mismo en el array

Mis Favoritas (favorites.html)

Esta página servirá para mostrar las películas que fueron seleccionadas como favoritas en la cartelera.



Rápidos y furiosos X

Código: 385687

Título original: Fast X

Idioma original: en

Resumen: A través de varias misiones y contra lo imposible, Dom Toretto y su familia han sido más astutos y más rápidos que todos los enemigos se le han cruzado en su camino. Ahora se enfrentan a su enemigo más letal: una amenaza aterradora que surge de las sombras del pasado que está alimentado de una venganza sangrienta, y está decidido a destruir a su familia y destruir todo, y a cualquier persona, a los que Dom ama.

Quitar de Favoritos

La grilla (grid) de resultados a mostrar en la pantalla tendrá una visualización de dos películas por línea.



Rápidos y furiosos X

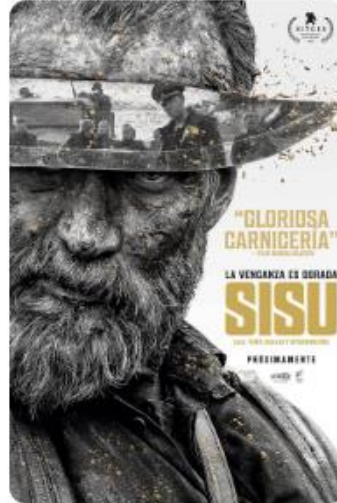
Código: 385687

Título original: Fast X

Idioma original: en

Resumen: A través de varias misiones y contra lo imposible, Dom Toretto y su familia han sido más astutos y más rápidos que todos los enemigos se le han cruzado en su camino. Ahora se enfrentan a su enemigo más letal: una amenaza aterradora que surge de las sombras del pasado que está alimentado de una venganza sangrienta, y está decidido a destruir a su familia y destruir todo, y a cualquier persona, a los que Dom ama.

Quitar de Favoritos



Sisu

Código: 840226

Título original: Sisu

Idioma original: fi

Resumen: Durante los últimos y desesperados días de la Segunda Guerra Mundial, un solitario buscador de oro (Jorma Tommila) se cruza con los nazis en una retirada al norte de Finlandia. Cuando los nazis le roban el oro, descubren rápidamente que no se han metido con un minero cualquiera. Aunque no exista una traducción directa de la palabra finlandesa "sisu", este legendario ex-comando encarnará lo que significa sisu: una forma de coraje y determinación inimaginables frente a probabilidades abrumadoras. Y no importa lo que los nazis le echen encima, el escuadrón de la muerte de un solo hombre hará todo lo posible por recuperar su oro, aunque eso signifique matar a todos los nazis que se crucen en su camino.

Quitar de Favoritos

Código HTML inicial

Al iniciar, la pantalla debe mostrar todas las películas que tiene almacenadas en el storage.

Si no hay películas almacenadas en el storage, el usuario debe ver un mensaje que indica que no tiene películas seleccionadas.

No tiene películas seleccionadas en sus favoritos

Si hubo un error con la carga de información o la consulta a la API se debe mostrar un mensaje de error al usuario.

Mostrar películas Favoritas

Recuperar las películas que han sido agregadas al listado desde el localStorage y mostrarlas en la grilla al usuario.

Generar una función async que recupere el contenido de nuestro campo en el storage, validar que el objeto no esté vacío y recorrerlo para consultar el detalle de la película en el servicio API. Esto se puede hacer transformando el resultado en un array o en un JSON, de cualquiera de las dos formas podemos recorrerlos de igual manera:


```
let favorites = localStorage.getItem("FAVORITOS");
    if (favorites) {
        favorites = JSON.parse(favorites);
        for (let i = 0; i < favorites.length; i++) {
```

Para consultar el detalle de la película utilizaremos el método MOVIE al cual debemos proporcionarle el ID de la película.

```
https://api.themoviedb.org/3/movie/${favorites[i]}?api_key=xXxXxX&language=es-
MX&append_to_response=${favorites[i]}
```

Por cada película se mostrara la tarjeta con el Poster, el Título, Código, Título Original, Idioma Original y Resumen.

```
img class="poster" src="https://image.tmdb.org/t/p/w500/${poster_path}">
<h3 class="titulo">${title}</h3>
<p><b>Código:</b> ${id}<br>
<p><b>Título original:</b> ${original_title}<br>
<b>Idioma original:</b> ${original_language}<br>
<b>Resumen:</b> ${overview}</p>
```

Quitar una película del listado

Cada película tendrá un botón que permita al usuario quitar de la lista de favoritos esa película; reorganizando de esta manera el array de valores guardados en el Storage.

Quitar de Favoritos

IMPORTANTE: Quitada la película del storage se debe actualizar la pantalla sin refrescar la pantalla.

Contemplar escenarios de error retornados por las API

En las solicitudes a las API puede que haya un error. Esta funcionalidad servirá para manejar esos errores de forma apropiada e informar al usuario de estos para que realice algún cambio sobre la información enviada.

En caso de obtener un error, notificar con un mensaje de error (los mismos que se usaron hasta ahora en las entregas anteriores) para avisar al usuario del correcto ingreso de la ciudad.

Se recomienda el uso de try catch para proteger diferentes secciones de la aplicación que sean susceptibles a errores e informar los mismos en caso de que ocurran.

Requerimientos Extras

Estos requerimientos a continuación son opcionales y servirán para sumar nota a la calificación final y por ende facilitará alcanzar la condición de aprobación directa.

Contemplar escenario de carga

Esta funcionalidad permitirá darle al usuario un mensaje para informarle cuando una operación está siendo realizada. Puede hacer uso de iconos o spinners en el HTML.

El usuario debería ver este aviso de cargando en la pantalla solo de modo informativo cuando realiza las siguientes operaciones:

- Cargar Cartelera de Cine
- Consultar Favoritos

Si considera que hay operaciones que se realizan de forma muy rápida, que no se alcanza a visualizar el icono de carga, puede utilizar `setTimeout` para dar una mejor experiencia al usuario.

Agregar Video a la película

Es posible agregar video, para ello se debe usar el método `video` y pasarle los parámetros correspondientes:

https://api.themoviedb.org/3/movie/movie_id/videos?

Agregar video en la misma tarjeta de la película en favoritos, con un tamaño que no afecte al resto de la información luego de la sinopsis

Integración con API de envío de emails

Integrar la aplicación actual al servicio de mensajería por email usando la librería [EmailJS](#). Esta librería JS permite el envío de emails únicamente utilizando su API JavaScript que realiza en la operación internamente.

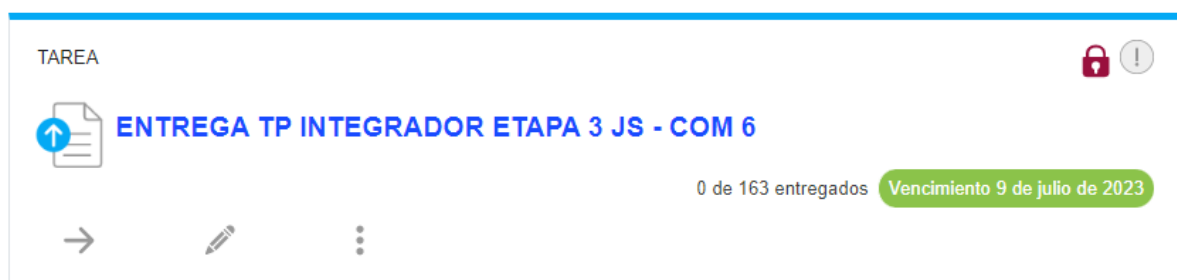
Esta implementación deberá realizarse sobre la página **help.html**. Pueden consultar [aquí](#) la documentación de la librería para obtener instrucciones precisas de cómo integrarla correctamente: <https://www.emailjs.com/docs/tutorial/overview/>.

Subir aplicación a Github Pages

Seguir instrucciones dadas en la página oficial de **Github Pages** (<https://pages.github.com/>) para la correcta subida y despliegue de la aplicación en la plataforma.

Condiciones de entrega

La fecha límite para la entrega es el **09/07/23**, la cual se deberá realizar a través CVG en la tarea habilitada para tal fin en la unidad correspondiente.



En el cuerpo del mensaje, se debe indicar:

- URL del repositorio del código fuente.
- URL de Git Pages donde esta publicado el sitio.
- Consideraciones: cualquier consideración relacionada al desarrollo y/o resolución del problema. Si te pareció muy fácil o complicado. Que mejorarías para las futuras etapas.