

Proyecto Integrador

Condiciones Generales

- El proyecto integrador es grupal y se realiza de a grupos de 2 o 3 personas, sin excepciones. Los alumnos pueden elegir sus compañeros de trabajo libremente. Deben confirmar los grupos de trabajo vía Slack hacia la cátedra para el 24/7. En caso de tener algún conflicto es fundamental que lo avisen previamente.
- La fecha de entrega final del proyecto integrador será el **Viernes 21 de Junio** en exposiciones grupales que se realizarán en el aula.
- Las presentaciones se harán en forma grupal y queda a criterio de los alumnos si agregar recursos soporte (como PPTs) para sus exposiciones. Si se espera durante la presentación una demo de la página web y un breve recorrido por el código destacando aspectos que fueron dificultosos durante el desarrollo. Durante la presentación también se harán preguntas a los alumnos que esperamos se respondan de manera individual.
- La entrega se realizará pasando un link vía slack. El link debe apuntar al repositorio actualizado en Github
- Es importante que utilicen GIT durante el proyecto y no únicamente para la entrega. Estaremos observando el commit log y quienes realizan dichos commits. Esto afecta tanto a la nota del trabajo como la nota de concepto

Consigna General

- Deben realizar una sitio web sobre películas. El sitio contará con 6 páginas diferentes las cuales se describirán a continuación. Sin embargo, en caso de desear realizar menos páginas pero que cubran las funcionalidades descritas, puede realizarse consultando previamente con la cátedra.
- La estética del sitio debe mantener una línea en común a través de todas sus páginas e inclusive el header y el footer debe ser común a todas las páginas.
- Las páginas deben ser responsive y contar con resoluciones específicas para al menos 2 resoluciones (Desktop y mobile)
- El diseño y la estética queda a cargo y elección del alumnado aunque en caso de precisar ayuda no duden en pedir soporte a la cátedra.
- El sitio web interactuará con la API provista por <https://www.themoviedb.org/>. Si bien sobre el final de la consigna se especifican los endpoints a utilizar en el trabajo, cuentan con la documentación entera en <https://developers.themoviedb.org/3/>
 - Para poder utilizar la API deben generar una API KEY, ya estando logueados en la plataforma en <https://www.themoviedb.org/settings/api>. Si necesitan ayuda con este paso, no duden en avisar a la cátedra.
 - Por cuestiones legales es fundamental añadir referencias a TMDb en sus proyectos integradores. Para esto les dejamos un link a los logos oficiales de TMDb y debe haber alguno de estos en el footer de su sitio web. <https://www.themoviedb.org/about/logos-attribution>

Acerca del header y el footer

El header debe tener al menos:

- A. El logo de su sitio
- B. Un formulario con un buscador (Al buscar debe enviar a la **página 4**)
 - i. Nótese que la búsqueda debe ejecutarse tanto al hacer enter como al clicar en el botón de buscar (Queda a decisión de los alumnos si incluir o no este botón)
 - ii. Al realizar la búsqueda debe validarse que el campo de búsqueda no esté vacío y tenga al menos 3 caracteres. En caso de error debe avisarse esto al usuario mediante un cartel que desaparezca luego de 3 segundos.
- C. Un link al listado de géneros
- D. Un saludo al usuario
 - i. En caso de no haberse logueado debe haber un botón de “Login” (A continuación se aclara que pasa al clicar en este botón)
 - ii. En caso de ya haber indicado su nombre, saludarlo por el nombre
- E. Un link a “Mis películas preferidas” (**Ver página 6**). Este hipervínculo sólo debe aparecer si el usuario ya indicó su nombre

Al clicar en el botón de Login, el comportamiento esperado es el siguiente:

1. Debe aparecer un modal/pop up que impida seguir navegando el sitio y le pida al usuario:
 - a. Nombre
 - b. Email
 - c. Género

Al enviar los datos debe validarse **por Javascript** que los campos estén completos y que el campo de email sea un email.

Tras enviar el formulario debe reemplazarse el botón de “Login” por un saludo al usuario y debe habilitarse el botón de “Mis películas preferidas”.

Notese que si el usuario efectúa el login en una página, debe seguir logueado en las siguientes.

Esta funcionalidad está asociada al header con lo cual el usuario podrá loguearse en cualquier página.

El footer debe tener al menos:

- a. Nombres de los alumnos que hicieron el proyecto
- b. Logo de referencia a TMDb

Página 1 - Home

La home de su sitio debe contar al menos con una sección principal que cuente al menos con:

1. Listado de las películas más populares (Queda a criterio de los alumnos cuántas películas mostrar)
2. Listado de las películas con mayor puntaje (Queda a criterio de los alumnos cuántas películas mostrar)
3. Listado de las películas que se lanzarán próximamente (Queda a criterio de los alumnos cuántas películas mostrar)

Nótese que al clicar en cualquiera de las películas debe redirigir al detalle de esa película **[Ver página 5]**

Punto extra: Se valora que las películas se muestren con su poster y se valora más aún que figuren en un carousel.

Página 2 - Listado de géneros

El listado de géneros debe mostrar al menos el listado de géneros provistos por TMDB. Al clicar en un género debe llevar a la página 3.

Página 3 - Películas por género

La página de películas por género debe mostrar:

- 1) El nombre del género buscado
 - a) En caso de que no se encuentre el género o no exista, debe indicarse un mensaje de error.
- 2) Un listado con las películas más populares del género (Queda a criterio de los alumnos cuántas películas mostrar)

Notese que al clicar en cualquiera de las películas debe redirigir al detalle de esa película **[Ver página 5]**

Punto extra: Se valora que las películas se muestren con su poster y se valora más aún que figuren en un carousel.

Página 4 - Resultado del buscador

La página de resultados del buscador debe mostrar:

1. El nombre del género buscado

- a. En caso de que no se encuentren resultados debe mostrarse un mensaje que avise esto
2. Un nuevo espacio para rehacer la búsqueda destacado en la parte principal del sitio. Este campo debe aparecer pre-cargado con la información que ya se busco por el usuario. Con el mismo criterio, el buscador del header también ya debería estar pre-cargado. Este espacio de búsqueda también debe estar validado con las mismas reglas que aplican al buscador en el header.
3. Un listado con las películas que sean resultantes de la búsqueda (Queda a criterio de los alumnos cuántas películas mostrar)

Nótese que al clicar en cualquiera de las películas debe redirigir al detalle de esa película [**Ver página 5**]

Punto extra: Se valora que las películas se muestren con su poster y se valora más aún que figuren en un carousel.

Página 5 - Detalle de una película

La página de detalle de la película debe incluir al menos:

1. Título de la película
2. Géneros a los que pertenece
 - a. Cada género debe linkear a la **página 3** que corresponda según el género
3. Lenguaje original
4. Sinopsis
5. Fecha de estreno
6. Poster de la película (en caso de tener)
7. Trailer de la película (en caso de tener)

Además, debe haber un botón que diga “Ver Recomendaciones”. Al clicar en este botón debe cargarse una sección (que a priori no está o aparece oculta) que muestra un listado de películas recomendadas (según la película que uno esté viendo).

Queda a criterio del alumnado cuantas películas recomendadas mostrar.

Nótese que al clicar en cualquiera de las películas debe redirigir al detalle de esa película.

Punto extra: Se valora que las películas recomendadas se muestren con su poster y se valora más aún que figuren en un carousel.

Por último, debe aparecer un ícono o botón en la página que permita que el usuario marque su película como favorita. La lógica al hacer click en este botón debe ser la siguiente:

1. Si el usuario ya indicó su nombre previamente, simplemente debe cambiar el estado del botón para indicar que la película ya es favorita. Se debe almacenar dicha película para “recordar” que ya fue marcada como favorita

2. Si el usuario aún **no** indicó su nombre debe simularse el mismo proceso de “Login” que si el usuario hubiese clickeado en ese botón.

Nótese que tras marcar una película como favorita se esperan los siguientes comportamientos:

- 1) Al volver a clickear en el botón, debe quitarse la película de la lista de favoritas.
- 2) Al recargar la página, el botón debe indicar que la película ya es favorita.

Página 6 - Mis películas preferidas

Esta página debe ser solamente accesible en caso de que el usuario ya haya pasado por el flujo de “Login”. Caso contrario se debe redirigir al usuario a la home.

Esta página debe indicar un listado de las películas que el usuario haya marcado como favoritas. En caso de no tener ninguna, indicarlo mediante un mensaje.

Punto extra: Se valora que las películas recomendadas se muestren con su poster y se valora más aún que figuren en un carrousel.

Punto extra - Paginados

Se valora que las páginas de “Películas por género” y de “Buscador” cuenten con un sistema de navegación de páginas para ver más resultado o una lógica de “Ver más” que siempre cargue más y más películas hasta no tener más resultados.

Puntos extra - Otros

Otras funcionalidades a elección de los alumnos también serán tenidas en cuenta en la nota. En este caso, es importante avisar a la cátedra previamente sobre la implementación de estas funcionalidades.

Anexo - Especificaciones API TMDb

Obtener detalles de una película

Documentación: <https://developers.themoviedb.org/3/movies/get-movie-details>

Ejemplo: https://api.themoviedb.org/3/movie/550?api_key=<<api_key>>

Recomendaciones para una película

Documentación: <https://developers.themoviedb.org/3/movies/get-movie-recommendations>

Ejemplo:

https://api.themoviedb.org/3/movie/550/recommendations?api_key=<<api_key>>&page=1

Imágenes y posters de una película

Docuementación: <https://developers.themoviedb.org/3/movies/get-movie-images>

Ejemplo: https://api.themoviedb.org/3/movie/550/images?api_key=<<api_key>>

Nótese que en el resultado, cada imagen tiene un `file_path`. Para luego conseguir el hipervínculo a la imagen final, la URL sería https://image.tmdb.org/t/p/original/FILE_PATH

Por ejemplo si el `file_path` fuese **awYPmDQc8Wga3P2RRutlpPD6PAD.jpg** la URL de la foto sería <https://image.tmdb.org/t/p/original/awYPmDQc8Wga3P2RRutlpPD6PAD.jpg>

Videos de una película

Documentación: <https://developers.themoviedb.org/3/movies/get-movie-videos>

Ejemplo: https://api.themoviedb.org/3/movie/550/videos?api_key=<<api_key>>

Nótese que en el resultado, cada video tiene un **key**. Para luego conseguir el hipervínculo al video final sería: <https://www.youtube.com/watch?v=key>

Por ejemplo si el **key** fuese **BdJKm16Co6M** la URL del video sería <https://www.youtube.com/watch?v=BdJKm16Co6M>

Luego será trabajo de ustedes trabajar con Youtube para ver cómo embeber dicho video.

Listado de géneros

Documentación: <https://developers.themoviedb.org/3/genres/get-movie-list>

Ejemplo: https://api.themoviedb.org/3/genre/movie/list?api_key=<<api_key>>

Listado de películas por género

Si bien TMDb no tiene un endpoint específico para esto, puede lograrse mediante el método `discover`. Les dejamos un link a la documentación y un ejemplo puntual de cómo lograr esto:

Documentación: <https://developers.themoviedb.org/3/discover/movie-discover>

Ejemplo:

`https://api.themoviedb.org/3/discover/movie?api_key=*<api_key>*&sort_by=popularity.desc&include_adult=true&include_video=true&page=1&with_genres=18`

Búsqueda de películas

Documentación: <https://developers.themoviedb.org/3/search/search-movies>

Ejemplo:

`https://api.themoviedb.org/3/search/movie?api_key=*<api_key>*&query=*<texto_a_buscar>*&page=1&include_adult=true`

Próximas películas

Documentación: <https://developers.themoviedb.org/3/movies/get-upcoming>

Ejemplo: `https://api.themoviedb.org/3/movie/upcoming?api_key=*<api_key>*&page=1`

Películas mejor puntuadas

Documentación: <https://developers.themoviedb.org/3/movies/get-top-rated-movies>

Ejemplo: `https://api.themoviedb.org/3/movie/top Rated?api_key=*<api_key>*&page=1`

Películas Populares

Documentación: <https://developers.themoviedb.org/3/movies/get-popular-movies>

Ejemplo: `https://api.themoviedb.org/3/movie/popular?api_key=*<api_key>*&page=1`