

## JUSTIFICACIÓN DE LA SOLUCIÓN

### Elección de la API de TMDb

Para el desarrollo del producto integrador, se optó por utilizar la API pública de The Movie Database (TMDb) como fuente principal de datos. Esta decisión responde al objetivo de crear una aplicación web funcional, moderna e interactiva, enfocada en el consumo de información dinámica y en tiempo real desde un servidor real.

TMDb ofrece una infraestructura robusta, gratuita y bien documentada para desarrolladores, con acceso a información actualizada sobre películas, series, actores, géneros, imágenes, trailers, entre otros recursos audiovisuales. Su uso permite construir una experiencia de usuario atractiva sin la necesidad de crear una base de datos manualmente.

### Comparación con un backend local (PHP/MySQL)

Aunque el enfoque tradicional con PHP y MySQL también es válido, el uso de una API externa ofrece ventajas significativas para un proyecto enfocado en Frontend:

Aspecto	TMDb API (Servidor real)	Backend local (PHP/MySQL con XAMPP)
Configuración	No requiere instalación local de servidor	Requiere configurar entorno con XAMPP
Datos	Reales, actualizados automáticamente	Creados manualmente, limitados
Mantenimiento	Sin mantenimiento propio	El alumno gestiona toda la lógica y datos
Tiempo de desarrollo	Enfocado en JS y experiencia de usuario	Se invierte tiempo en backend y BD
Objetivo del curso (Frontend)	Directamente relacionado	Parte del trabajo se desvía a backend

Este enfoque favorece el uso de recursos reales para potenciar las capacidades del lado cliente y centrarse en la manipulación del DOM, programación asíncrona y experiencia de usuario, tal como lo propone el objetivo de la actividad integradora.

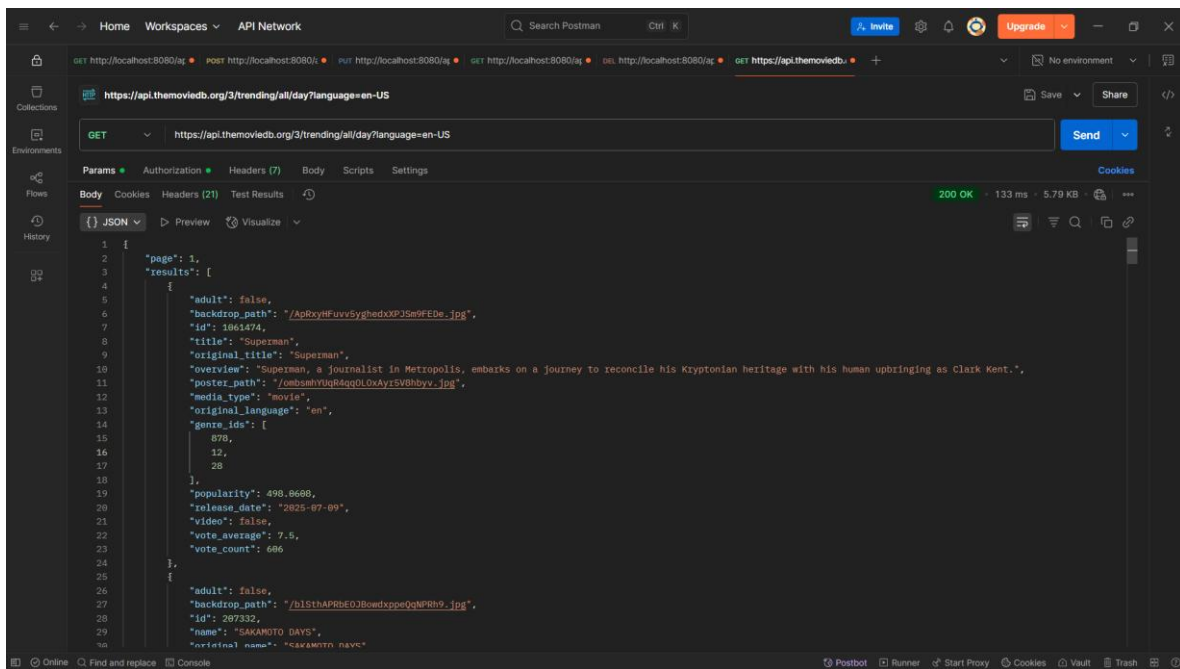
## Cumplimiento de los requisitos de la actividad

A continuación, se detalla cómo esta solución cumple con cada uno de los puntos establecidos en la consigna:

### Comunicación con servidor (API real):

Se realiza mediante peticiones `fetch()` al servidor de TMDb, utilizando una clave de autenticación (`api_key`) para obtener resultados según la consulta del usuario.

*Captura en Postman o consola del navegador mostrando una solicitud exitosa a TMDb (endpoint + JSON de respuesta).*



### Uso de programación asíncrona:

Las peticiones HTTP se gestionan con `async/await`, permitiendo cargar contenido sin recargar la página y manteniendo la fluidez de la interfaz.

*Captura sugerida: Fragmento del código JS donde se use `async function` y `await fetch()`.*

### Consumo y manipulación de datos JSON:

La respuesta de la API llega en formato JSON, que es procesado para extraer y mostrar información relevante como título, sinopsis, imagen y calificación.

*Captura sugerida: Objeto JSON de respuesta (en Postman o consola del navegador) y captura del resultado mostrado en pantalla tras el procesamiento.*

### Interacción dinámica con el usuario:

La aplicación responde a eventos de entrada del usuario (búsqueda, clics), actualizando dinámicamente el DOM con resultados personalizados. Además, se usa una biblioteca auxiliar (SweetAlert2) para mejorar la comunicación visual.

*Captura mostrando búsqueda de una película, resultados, y una alerta personalizada (por ejemplo, en caso de error o búsqueda vacía).*

### Referencias

Página oficial de The Movie Database: <https://www.themoviedb.org/>

Documentación técnica para desarrolladores: <https://developer.themoviedb.org/docs>

## INSTRUCCIONES PARA PROBAR EL PROYECTO

A continuación, se indican los pasos necesarios para ejecutar y probar la aplicación web desarrollada como producto integrador. El objetivo es facilitar la evaluación y asegurar que la aplicación funcione correctamente desde cualquier equipo.

### 1. Requisitos previos

Para ejecutar la aplicación, únicamente se requiere:

- Un navegador web moderno (se recomienda Google Chrome o Microsoft Edge).
- Conexión a Internet (para poder consumir la API de TMDb).

No se necesita servidor local (como XAMPP), ya que la aplicación consume datos desde un servidor externo real.

### 2. Archivos del proyecto

El proyecto está compuesto por los siguientes archivos:

- **index.html** → Página principal de la aplicación.
- **styles.css** → Hoja de estilos personalizada.
- **script.js** → Archivo JavaScript con toda la lógica de la aplicación.

### **3. Pasos para ejecutar la aplicación**

1. Descargue todos los archivos del proyecto en una carpeta local.
2. Abra el archivo index.html con doble clic o haciendo clic derecho → Abrir con → Navegador.
3. Espere unos segundos a que cargue la interfaz.
4. En la barra de búsqueda, escribe el nombre de una película (ej. Interestelar) y presione "Buscar".
5. Verifica que se muestran los resultados dinámicamente en pantalla.

### **4. Interacción esperada**

El usuario puede buscar películas escribiendo su nombre y recibiendo resultados dinámicos desde la API.

Si no se encuentra nada, se muestra un mensaje amigable mediante SweetAlert2.

El contenido se genera dinámicamente en el DOM, mostrando título, imagen, descripción y calificación.

### **5. Observaciones adicionales**

La aplicación usa una clave de acceso (api\_key) a la API de TMDB. Esta clave es de uso personal y está insertada en el archivo app.js para pruebas académicas y la revisión del código.

La API permite hasta 40 peticiones por cada 10 segundos, por lo que es poco probable que se bloquee durante la evaluación.

En caso de que no aparezcan resultados, asegúrese de tener conexión a Internet o verifique en la consola del navegador si hubo algún error de red.