

# Universidad de Guadalajara

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS



Ingeniería Informática.

Producto Integrador

**Desarrollo de Front End.**

Yosef Sánchez Gutiérrez  
Marco Alejandro González Mireles  
Mirella Stephania Palomera Gómez  
Roberto Carlos Martínez Aviña

Mtra. Rosalia Iñiguez.

## Justificación del proyecto:

### 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.

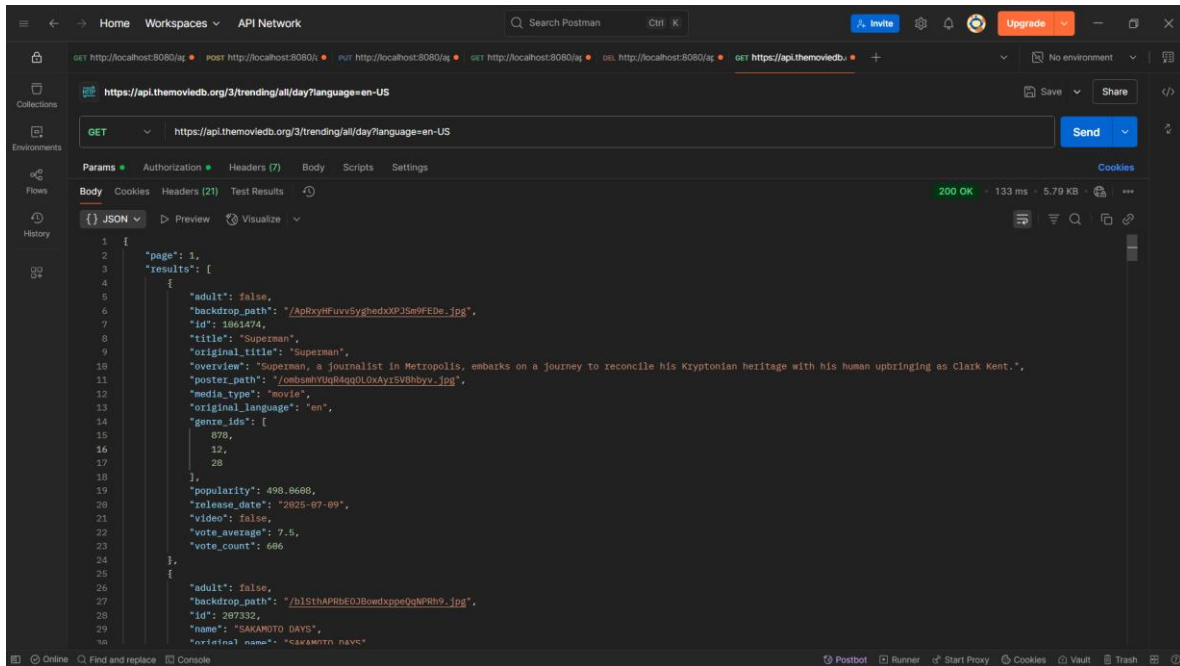
### Cumplimiento de los requisitos de la actividad

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).

## **RECOMENDACIONES 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.

## Introducción:

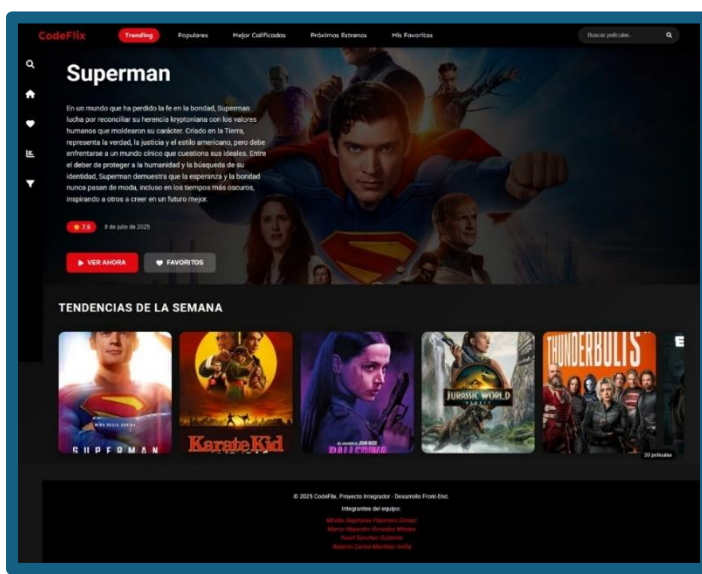
Durante este curso de Programación Web, aprendimos y practicamos los temas más importantes para crear sitios web desde cero. Comenzamos entendiendo bien **HTML**, que es la base para estructurar cualquier página, aprendiendo a usar etiquetas, organizar contenido con títulos, párrafos, listas, imágenes, enlaces y tablas, y a crear formularios que permiten la interacción con los usuarios. Después nos enfocamos en **CSS**, que nos ayudó a darle diseño y estilo a nuestras páginas. Vimos cómo cambiar colores, fuentes, márgenes, crear menús, dividir en secciones y usar flexbox o grid para distribuir elementos de manera ordenada. Además, practicamos el diseño responsivo, que es indispensable hoy en día porque los usuarios entran desde sus celulares, tablets o computadoras.

Más adelante, aprendimos **JavaScript**, que es lo que hace a los sitios web interactivos. Con JavaScript pudimos crear funciones, trabajar con variables, arreglos, condicionales y ciclos para automatizar tareas y programar la lógica de la página. También vimos cómo usar el **DOM** para cambiar el contenido de la página desde el código, como mostrar u ocultar información, cambiar textos o imágenes y reaccionar a eventos de botones o formularios.

Algo muy importante fue aprender a consumir **APIs** externas, como la de **TMDB** en este proyecto, usando fetch para traer datos de otros servidores y mostrarlos dinámicamente sin necesidad de recargar la página completa.

En este trabajo final aplicamos todo lo visto para crear un sitio web de películas, donde mostramos información real que obtenemos de la API de **TMDB**. Entre lo que hicimos está:

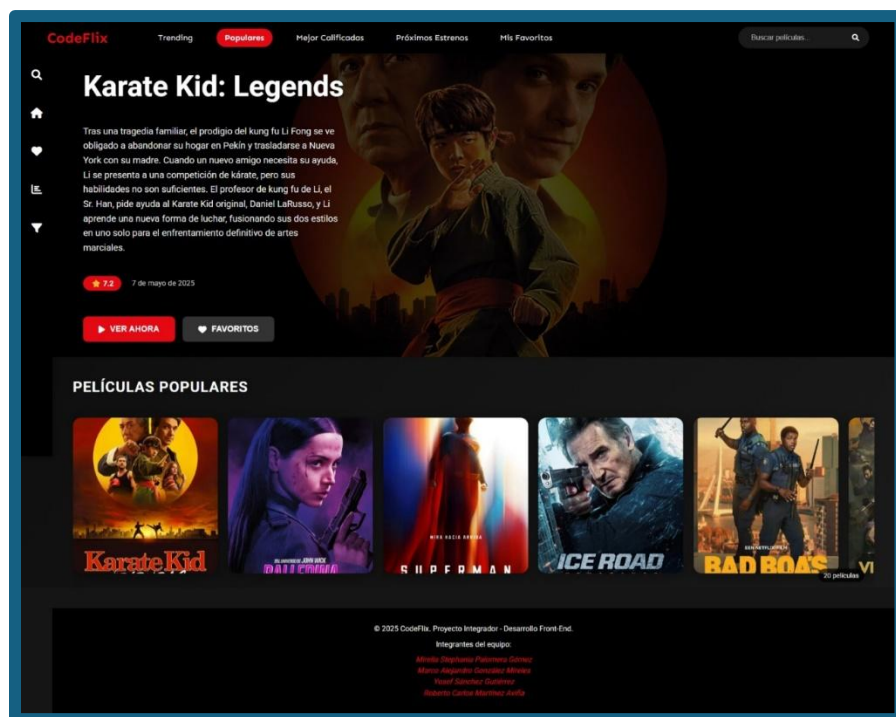
- Una barra de navegación fija con diseño atractivo y un sidebar con iconos.
- Una sección destacada que muestra la película principal con su imagen, título y descripción.
- Filtros para buscar películas por género, año o calificación.
- Listados de películas con tarjetas que tienen su imagen, nombre, calificación y botón de información.
- Un modal que se abre con más detalles de cada película, su sinopsis y gráfica de calificación.
- Un sistema para guardar favoritos usando **LocalStorage** y verlos después.
- Notificaciones y opción de modo claro u oscuro para mejorar la experiencia de uso.



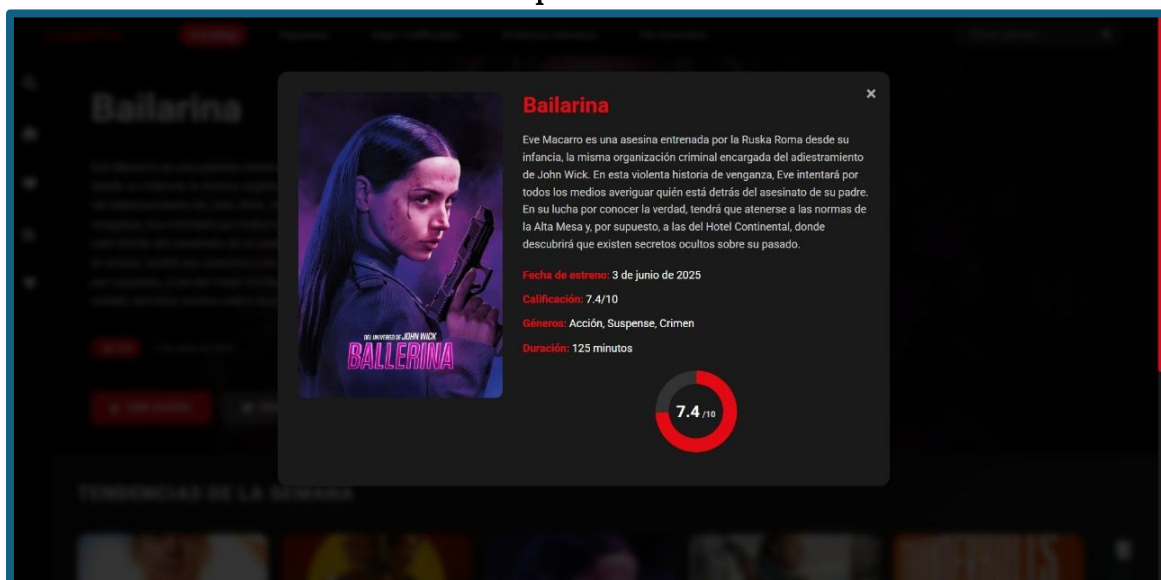
## Desarrollo:

### Vista general

En esta vista se puede apreciar la página principal de mi proyecto, donde se muestran las películas destacadas, las listas con las diferentes categorías y los filtros para buscar por género, año o calificación. Todo el diseño está pensado para que sea claro, atractivo y fácil de usar para cualquier visitante.



### Vista de película detallada



En esta vista se muestra el detalle de una película seleccionada. Aquí el usuario puede ver su póster, el título, la sinopsis completa, la fecha de estreno, la duración, la calificación promedio y los géneros a los que pertenece. Esta sección está diseñada para brindar toda la información necesaria de manera ordenada y visualmente atractiva.

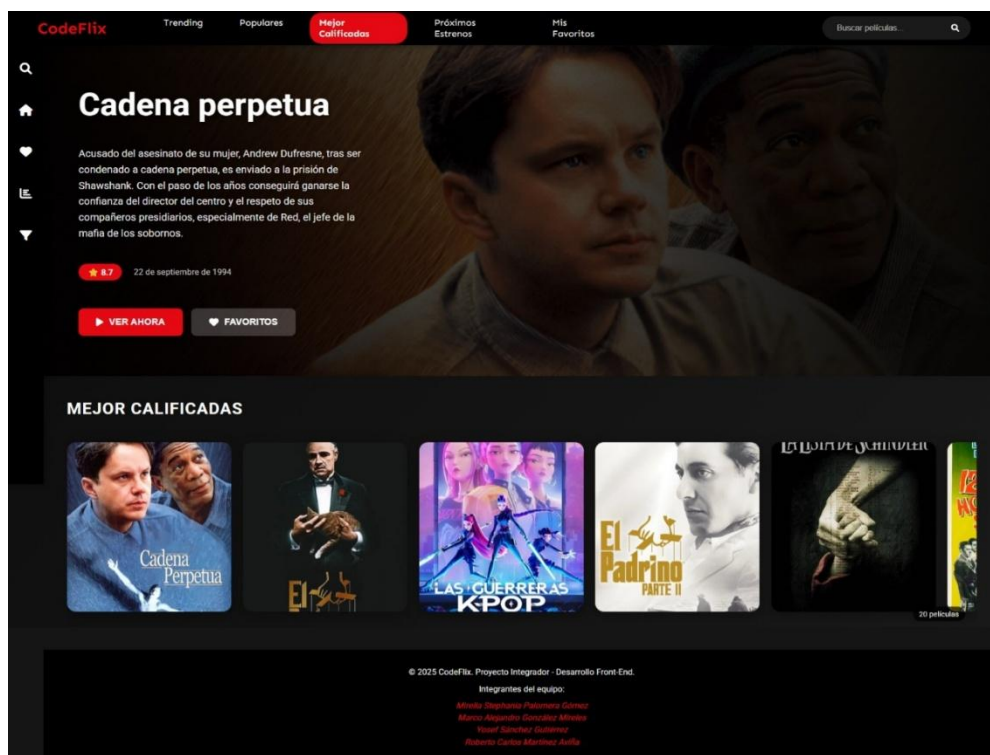
```
// Mostrar el modal de detalles de la película
async function showMovieDetails(movie) {
  console.log(`Mostrando detalles para la película: ${movie.title || movie.name}`)
  showLoading(true)
  try {
    // Obtener detalles completos de la película
    const response = await fetch(`${BASE_URL}/movie/${movie.id}?api_key=${API_KEY}&language=es-ES`)
    if (!response.ok) {
      const errorText = await response.text()
      throw new Error(`Error HTTP: ${response.status} - ${errorText}`)
    }
    const movieDetails = await response.json()
    console.log("Detalles de película recibidos:", movieDetails)

    if (document.getElementById("modalPoster")) {
      document.getElementById("modalPoster").src = movieDetails.poster_path
      ? `${IMAGE_BASE_URL}${movieDetails.poster_path}`
      : "data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' width='300' height='450'><rect width="
    }
    if (document.getElementById("modalTitle")) {
      document.getElementById("modalTitle").textContent =
        movieDetails.title || movieDetails.name || "Título no disponible"
    }
    if (document.getElementById("modalOverview")) {
      document.getElementById("modalOverview").textContent = movieDetails.overview || "Sinopsis no disponible."
    }
  }

  if (document.getElementById("modalDate")) {

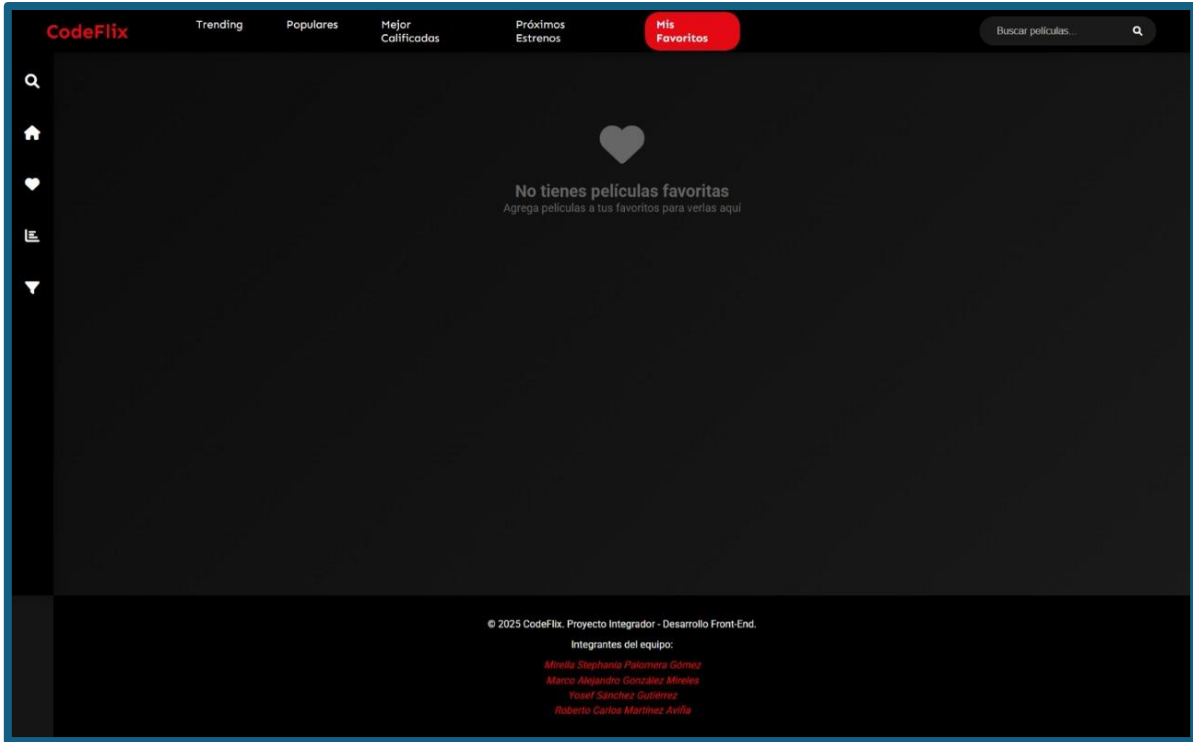
```

Cuando el usuario selecciona una película para ver más detalles, esta función se encarga de obtener la información completa desde la API de películas. Hace una solicitud para traer datos como el póster, título, sinopsis, fecha de estreno y calificación. Luego, actualiza el contenido del modal con estos datos para mostrarlos en pantalla. Si hay algún error al obtener la información, se muestra una notificación para avisar al usuario. Mientras se carga la información, se muestra un indicador de carga para mejorar la experiencia. Finalmente, el modal se muestra para que el usuario pueda ver todos los detalles de la película seleccionada.





## Vista de “Mis favoritos”



La vista de Favoritos muestra una lista personalizada de películas que el usuario ha marcado para ver más tarde o guardar porque le gustan. Esta sección permite al usuario revisar sus películas favoritas fácilmente sin tener que buscarlas nuevamente. En esta vista, cada película aparece con su imagen, título y opciones para eliminarla o ver más detalles. Así, el usuario tiene un acceso rápido y organizado a sus elecciones guardadas.

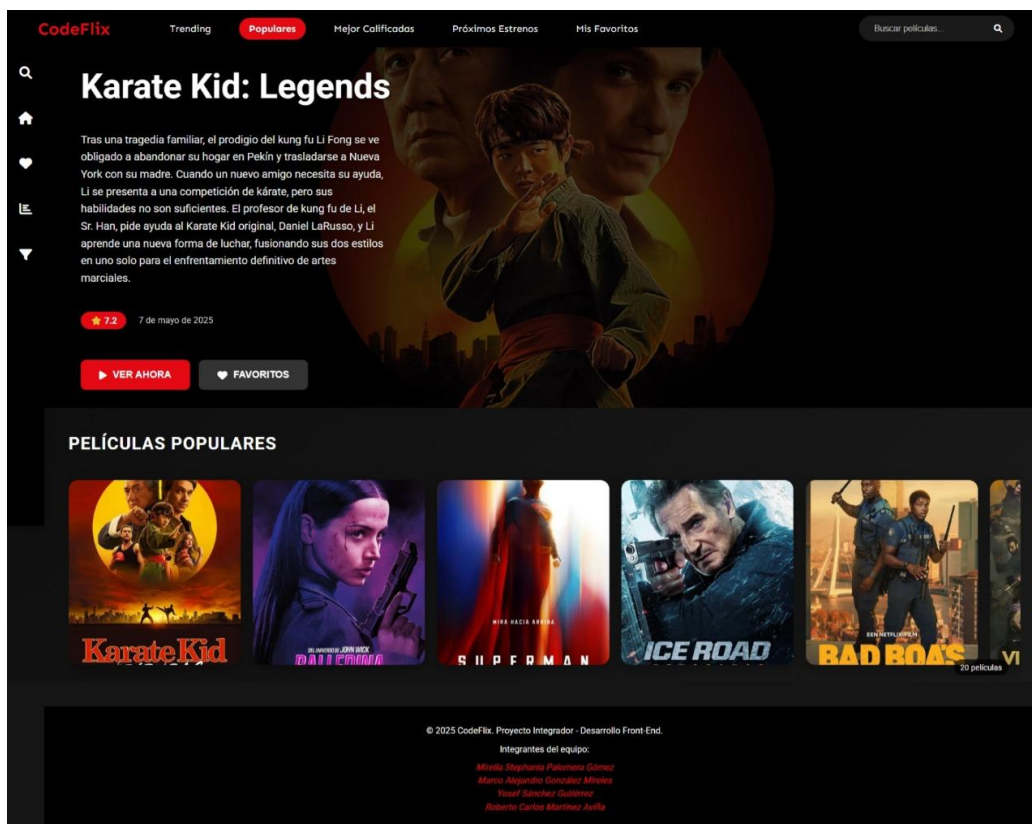
```
// Mostrar la lista de películas favoritas
function showFavorites() {
  currentCategory = "favorites"
  if (favoriteMovies.length === 0) {
    movieListsContainer.innerHTML = `
      <div style="text-align: center; padding: 100px; color: #666;">
        <i class="fas fa-heart" style="font-size: 4rem; margin-bottom: 20px;"></i>
        <h2>No tienes películas favoritas</h2>
        <p>Agrega películas a tus favoritos para verlas aquí</p>
      </div>
    `
    if (featuredContent) featuredContent.style.display = "none"
    return
  }

  if (featuredContent) featuredContent.style.display = "flex"
  displayFeaturedMovie(favoriteMovies[0])
  displayMovieList(favoriteMovies, "MIS PELÍCULAS FAVORITAS")
}
```

La función `showFavorites` muestra al usuario todas las películas que ha guardado como favoritas. Primero, establece que la categoría actual es "favoritos" para identificar la sección correcta. Si el usuario no tiene películas favoritas, se muestra un mensaje indicando que no hay favoritos y se oculta el contenido destacado. En caso contrario, se muestra el contenido destacado con la primera película favorita y, debajo, una lista con todas las películas favoritas. Esto permite al usuario acceder fácilmente a sus películas guardadas para verlas cuando desee.

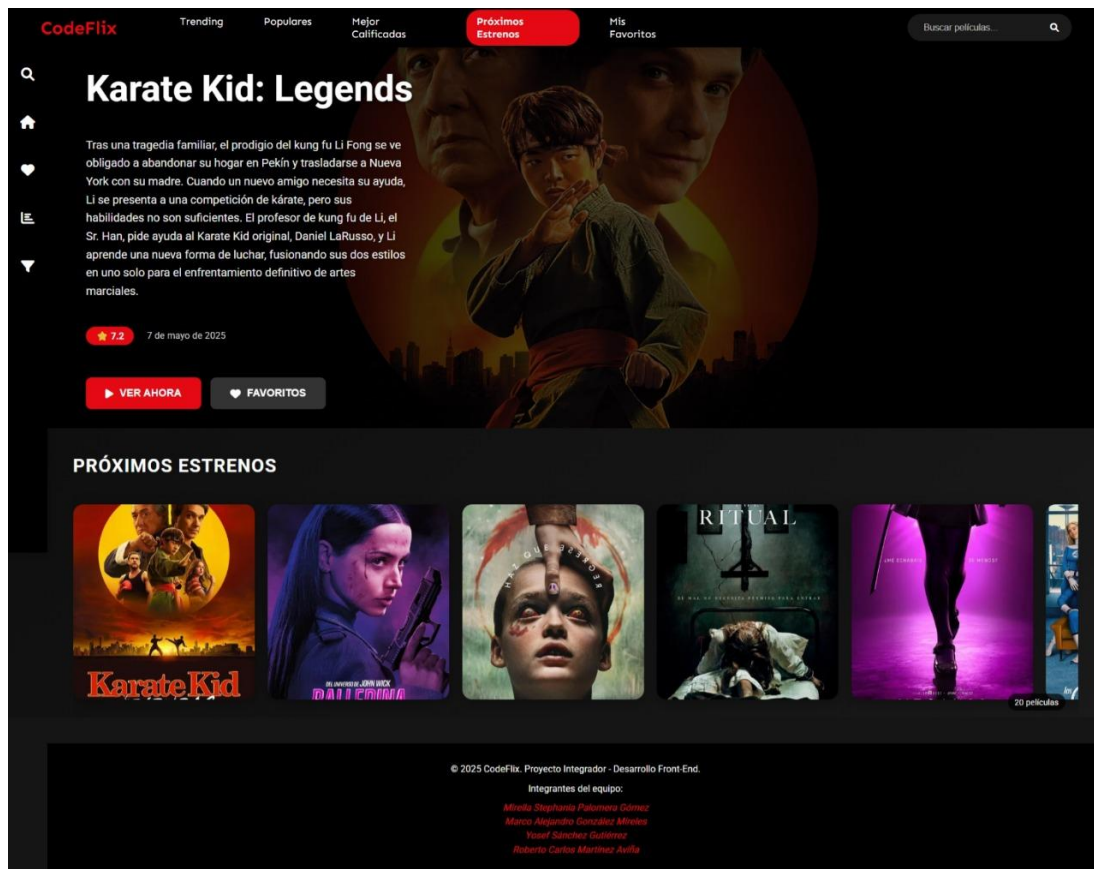
## Vista de “Populares”

La sección de *Películas Populares* presenta los títulos más vistos o consultados recientemente por los usuarios en la plataforma de TMDb. Esta lista se carga dinámicamente utilizando la categoría popular de la API (`/movie/popular`), y se presenta visualmente en formato horizontal, con tarjetas que muestran el póster, el nombre del filme y su calificación. Cada película en esta sección incluye un botón de interacción rápida para marcarla como favorita, y un clic sobre la tarjeta despliega el modal con detalles ampliados. Esta sección permite a los usuarios estar al tanto de lo más popular en el momento, favoreciendo decisiones rápidas de visualización.



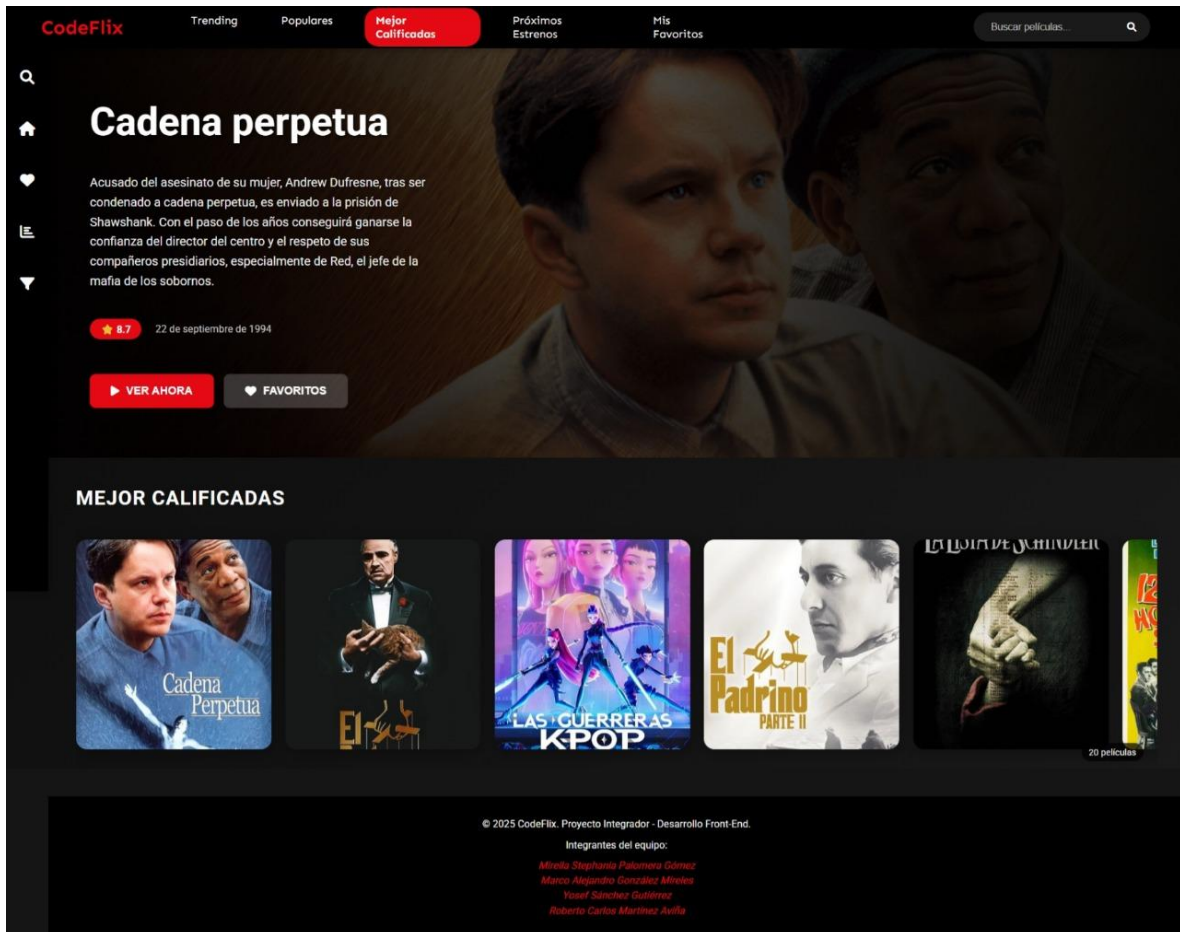
## Vista de “Proximos estrenos”

La categoría *Próximos Estrenos* permite a los usuarios anticiparse a los lanzamientos más esperados. Las películas que aparecen en esta sección son obtenidas a través del endpoint `/movie/upcoming` de TMDb, el cual filtra películas que aún no se estrenan oficialmente pero ya tienen fecha prevista de lanzamiento. Esta sección mejora la experiencia del usuario al permitirle planificar qué ver próximamente. Visualmente se integra igual que otras secciones: con una fila de tarjetas, flechas para navegar lateralmente y la opción de agregar a favoritos antes del estreno. También es posible ver la fecha futura de estreno directamente desde el modal.



## Vista de “Mejor calificadas”

La sección *Mejor Calificadas* contiene películas que han recibido puntuaciones elevadas por parte de los usuarios de TMDB. Se alimenta mediante el endpoint `/movie/top Rated`, y sirve como una guía de calidad, orientando al usuario hacia contenido recomendado por la comunidad. Las películas aquí suelen incluir títulos clásicos y producciones con gran aceptación crítica. Esta sección representa un filtro automático de alta calidad y puede combinarse con filtros adicionales por año o género para refinar los resultados.



## Conclusiones:

El desarrollo de este proyecto integrador representó una oportunidad significativa para aplicar de manera práctica los conocimientos adquiridos a lo largo del curso de Programación Web. A través de la creación de una aplicación dinámica, moderna y completamente funcional, se pudo comprobar la utilidad y complementariedad de las tecnologías fundamentales del desarrollo frontend: HTML, CSS y JavaScript.

Uno de los principales logros del proyecto fue entender cómo se estructura un sitio desde cero, no solo en términos visuales, sino también en su lógica y funcionamiento. Se logró construir una interfaz amigable, atractiva y adaptable a diferentes dispositivos, gracias a la implementación de diseño responsivo y buenas prácticas de maquetación. Asimismo, el uso de CSS permitió experimentar con estilos avanzados, distribuciones flexibles y sistemas de navegación intuitivos.

En el ámbito de la programación con JavaScript, se desarrollaron habilidades esenciales como el manejo de funciones, condicionales, eventos y manipulación del DOM. Sin embargo, uno de los aspectos más valiosos fue el aprendizaje sobre consumo de APIs externas. Integrar la API pública de The Movie Database (TMDB) fue clave para entender cómo trabajar con datos reales en tiempo real, aprender a realizar peticiones HTTP con fetch, gestionar respuestas en formato JSON y renderizar dinámicamente contenido en la página.

El proyecto también incorporó funciones adicionales como el guardado de favoritos mediante LocalStorage, la opción de cambiar entre modo claro y oscuro, y el uso de bibliotecas como SweetAlert2 para mejorar la experiencia del usuario, lo cual agregó una capa de personalización y profesionalismo al producto final.

En conclusión, este trabajo no solo evidenció la comprensión técnica de los temas del curso, sino que también promovió el pensamiento lógico, la resolución de problemas y la creatividad. Deja como resultado una aplicación útil y bien estructurada, además de sentar las bases para futuros desarrollos más complejos en el área del desarrollo web.

## Referencias:

The Movie Database (TMDB):

The Movie Database. (s. f.). The Movie Database (TMDB). <https://www.themoviedb.org/>

Recuperado el 16 de julio de 2025, de <https://www.themoviedb.org/>

Documentación técnica para desarrolladores:

The Movie Database. (s. f.). Developer Documentation. <https://developer.themoviedb.org/docs>

Recuperado el 16 de julio de 2025, de <https://developer.themoviedb.org/docs>

