Documentación del Sistema react-tmdb-api

# 1. Resumen del sistema

Nombre: react-tmdb-api  
Descripción: Aplicación web que consume la API oficial de TMDb para mostrar listados de películas, navegación por categorías, visualización de información detallada y portada, todo implementado con React y optimizado para rendimiento y compatibilidad.

# 2. Requerimientos

## Requerimientos Funcionales

|  |  |  |
| --- | --- | --- |
| ID | Descripción | Prioridad |
| RF01 | Mostrar listado de películas | Alta |
| RF04 | Mostrar portada e información de cada película | Alta |
| RF05 | Mostrar calificación/puntuación | Media |
| RF06 | Navegación por categorías (popular, top-rated, upcoming) | Media |
| RF03 | El sistema no tiene página de login | Alta |

## Requerimientos No Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Descripción | Categoría | Prioridad |
| RNF01 | Tiempo de respuesta rápido | Rendimiento | Alta |
| RNF02 | Compatibilidad con navegadores modernos | Compatibilidad | Alta |
| RNF03 | Manejo de errores implementado | Fiabilidad | Alta |
| RNF04 | Uso de React como biblioteca principal | Implementación | Alta |
| RNF05 | Uso de API oficial de TMDb | Implementación | Alta |

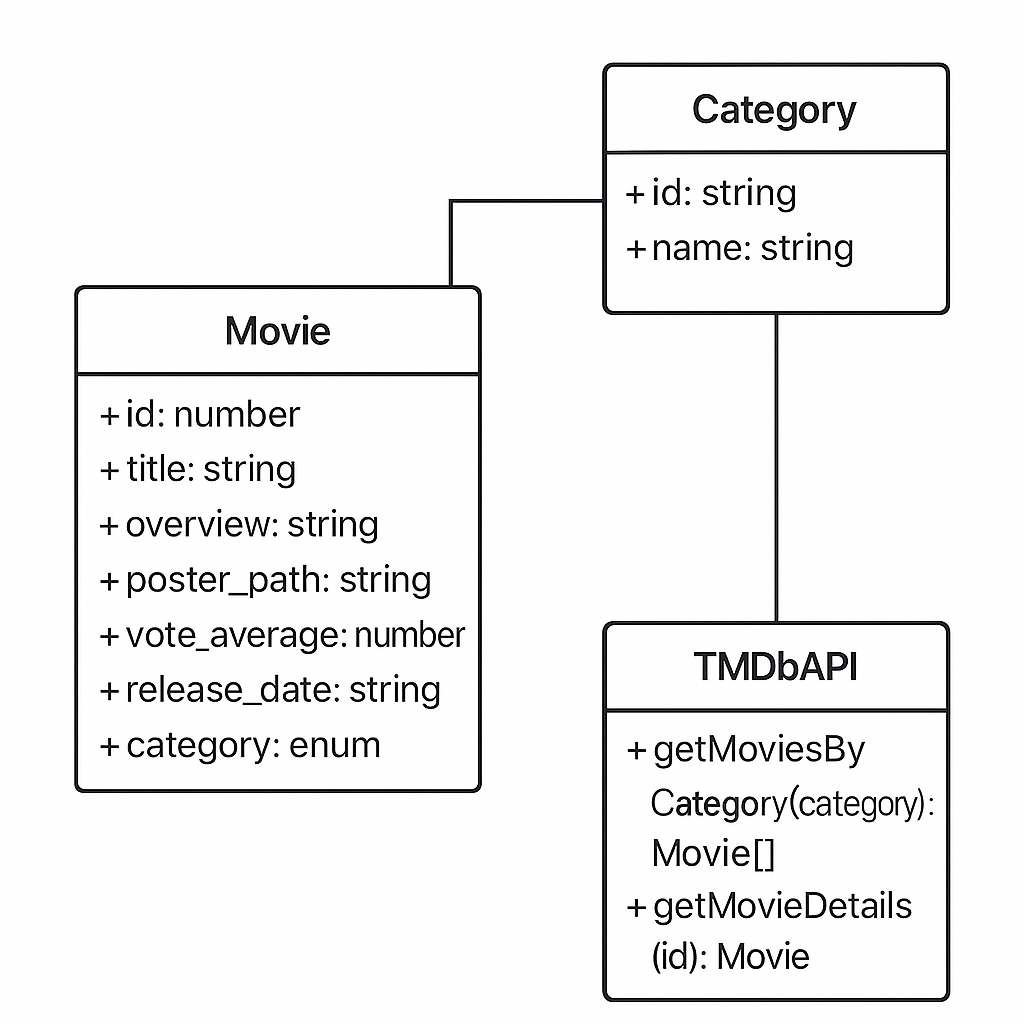
# 3. Entidades principales

🎬 Movie:  
- id: number  
- title: string  
- overview: string  
- poster\_path: string  
- backdrop\_path: string  
- vote\_average: number  
- release\_date: string  
- category: enum

📂 Category:  
- id: string  
- name: string

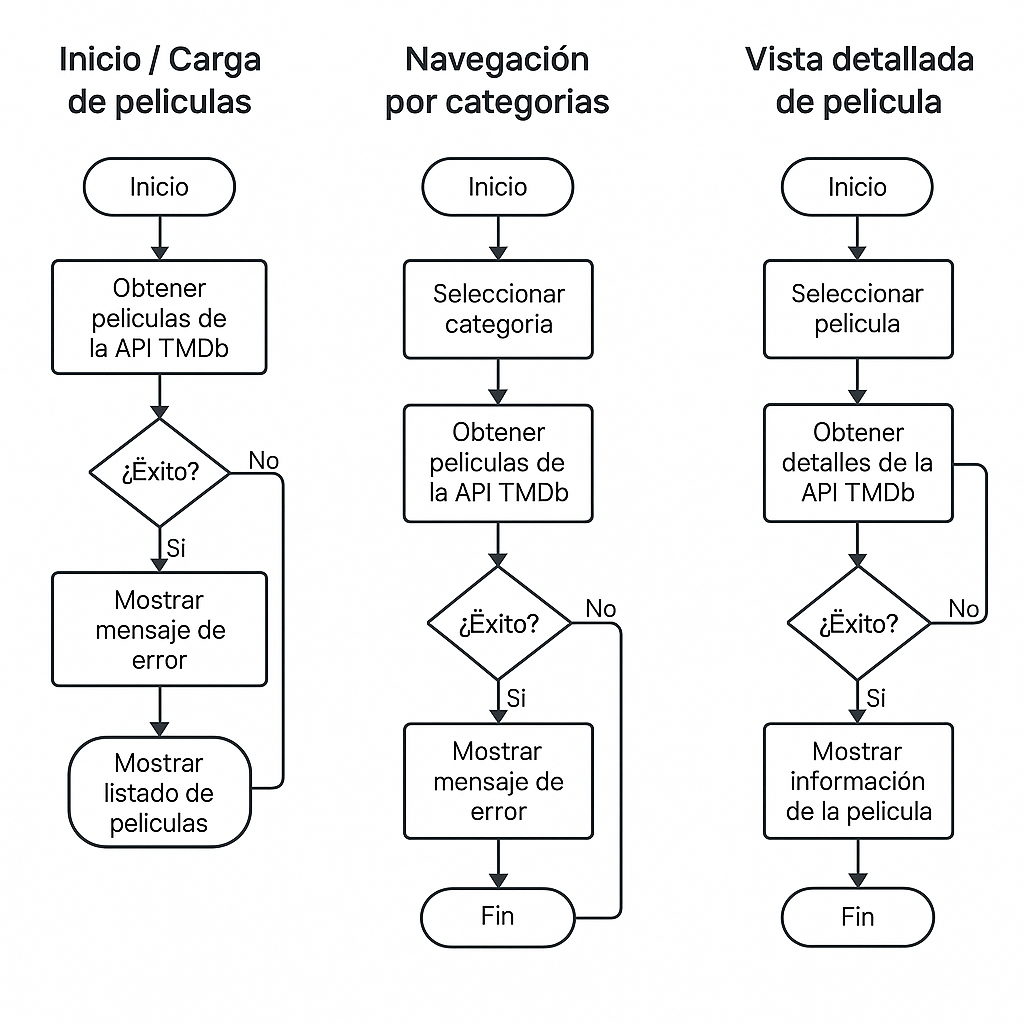
🌐 TMDbAPI:  
- getMoviesByCategory(category): Movie[]  
- getMovieDetails(id): Movie

# 4. Diagrama de clases UML



# 5. Flujos del sistema

🔹 Inicio / Carga de películas  
1. Usuario accede al sitio.  
2. Se realiza una llamada a getMoviesByCategory('popular').  
3. Se muestran resultados o mensaje de error.  
  
🔹 Navegación por categorías  
1. Usuario hace clic en una categoría.  
2. Se llama a getMoviesByCategory(categoría).  
3. Se actualiza la vista.  
  
🔹 Vista detallada de película  
1. Usuario selecciona una película.  
2. Se llama a getMovieDetails(id).  
3. Se muestra la información completa.



# 6. Arquitectura general

Frontend (React):  
- Componentes: MovieList, MovieCard, MovieDetail, CategoryMenu  
- React Router para navegación  
- Llamadas a TMDb desde hooks personalizados  
  
Backend/API:  
- TMDb API oficial  
- Capa de caché opcional en frontend para mejorar rendimiento

# 7. Consideraciones de diseño

- React asegura una UI dinámica y eficiente.  
- Se utilizan hooks para mantener separación de lógica de negocio.  
- El manejo de errores permite mostrar mensajes claros si falla la API.  
- Las imágenes de portada y calificaciones permiten una mejor UX.

# 8. Conclusión

El sistema react-tmdb-api está diseñado para ser ágil, compatible, confiable y estéticamente atractivo, cumpliendo con todos los requisitos planteados. Gracias a su arquitectura modular y uso de estándares modernos, puede escalarse fácilmente y mantenerse en el tiempo.