TEST PLAN NIVEL: API (DB PELICULAS)



PLAN DE PRUEBAS FUNCIONAL NIVEL API TESTING ℰ

SUMARIO:

LEVEL TEST PLAN / FUNCIONALIDAD: THEMOVIEDB - PELICULAS 🔗

- Este plan se llevará a cabo a nivel de pruebas de INTEGRACIÓN.
- Este plan se llevará a cabo en POSTMAN de manera automatizada.
- Este plan se llevará a regresión.
- Este plan se realizara para determinar QUE pruebas se harán y COMO se realizarán las pruebas

OBJETIVO: ⊘

Validar las funciones básicas para encontrar peliculas, encontrar peliculas populares y filtrado por género usando POSTMAN de forma automatizada.

SUPOSICIONES:

- Suponemos que la API permite la capacidad de ordenar las películas de acuerdo con diferentes parámetros, como por popularidad o fecha de lanzamiento, y que las pruebas verificarán que el ordenamiento personalizado funciona según lo previsto.
- Suponemos que al solicitar películas a través de la API, se obtendrá un número específico de resultados (por ejemplo, 20 películas), y las pruebas validarán que la cantidad de resultados coincida con la configuración esperada.
- Si la API devuelve fechas, suponemos que las fechas se devuelven en un formato específico, como "AAAA-MM-DD," y las pruebas verificarán que las fechas sigan este formato.
- Suponemos que las URL de las imágenes se devuelven en un formato específico, como una URL completa a una imagen, y las pruebas validarán que las URL de las imágenes sean válidas.
- Suponemos que cada película tiene un identificador único (por ejemplo, el campo "id") y que las pruebas verificarán que no hay duplicados en los resultados.
- Suponemos que los géneros proporcionados por la API son válidos y coinciden con los géneros en una base de datos central. Las pruebas confirmarán que los géneros son coherentes.
- Suponemos que la función de búsqueda en la API funciona correctamente, y las pruebas verificarán que se pueden encontrar películas utilizando términos de búsqueda válidos.
- Suponemos que cada película en los resultados contiene información completa y que las pruebas validarán que los campos esenciales NOMBRE, IMAGEN Y DESCRIPCIÓN están presentes en cada respuesta.
- Suponemos que la base de datos de películas en la API puede cambiar con el tiempo debido a actualizaciones o adiciones de nuevas películas, y las pruebas considerarán estos cambios.
- Suponemos que "vote_average" es un valor numérico que representa la calificación promedio de una película en un rango válido, generalmente entre 0 y 10. Las pruebas validarán que los valores estén dentro de este rango.
- Suponemos que "vote_count" es un valor numérico que representa la cantidad de votos o calificaciones que ha recibido una película.

 Las pruebas verificarán que el valor sea igual o mayor que un valor mínimo (por ejemplo, 0 o 1, indicando que al menos una persona ha calificado la película).
- Suponemos que el endpoint de Películas Populares devuelve una lista de películas populares en función de ciertos criterios de popularidad. Las pruebas verificarán que las películas en la lista cumplan con criterios específicos de popularidad y que estén clasificadas de acuerdo con esos criterios.

- Suponemos que el endpoint de **Próximos Estrenos** proporciona una lista de películas que se lanzarán próximamente. Las pruebas verificarán que las películas incluidas sean de hecho próximos estrenos y que las fechas de lanzamiento cumplan con las expectativas.
- Suponemos que las películas próximas a estrenarse pertenecen a géneros apropiados para un lanzamiento en el futuro. Las pruebas validarán que las películas sean coherentes con esta suposición.
- Suponemos que el endpoint de Películas Mejor Valoradas devuelve una lista de las películas mejor valoradas en un momento dado.
 Las pruebas verificarán que las películas incluidas tengan calificaciones que las califiquen como "mejor valoradas" y que estén ordenadas en consecuencia.

▲ SCOPE (FOCO): ∂

- Capacidad de alcanzar solamente las funciones CRUD descritas en los criterios de aceptación de las historias de usuario.
- NO se VALIDARÁN los campos que no fueron descritos en la FUNCIONALIDAD.

▲ RIESGOS: ∂

RIESGO - (LO QUE SE NECESITA SOLVENTAR).	MITIGACIÓN - (CON LO QUE SE PUEDE SOLVENTAR).
No poder guardar el TOKEN de TM DB u olvidarla	 Solicitar el TOKEN al PM o DEV del Backend. Guardarlo en un archivo con nombre .env en la carpeta Backend del repositorio. Guardarlo en una variable en POSTMAN
No conocer como usar la API	Solicitar capacitación y documentación al PM o a los desarrolladores Backend sobre la misma
No tener acceso al Workspace de POSTMAN	Crear un workspace e invitar a los Tester QA para que puedan trabajar en él.
Falta de acceso a internet	Asegurarse de que el equipo de pruebas tenga acceso a Internet estable durante todo el proceso de pruebas.
Cambios no documentados en la API	Establecer una comunicación efectiva con el equipo de desarrollo para recibir notificaciones sobre cambios planificados en la API. Además, se debe monitorear la documentación y las actualizaciones de la API de TMDb para estar al tanto de los cambios.
Desconexion de recursos de desarrollo	Mantener una comunicación constante con el equipo de desarrollo para asegurarse de que los recursos y entornos necesarios para las pruebas estén disponibles y en línea.
Cambio en la estructura de datos de la API	Establecer una estrategia de pruebas sólida que incluya pruebas de regresión para verificar que la API sigue funcionando después de cambios en la estructura de datos.

DESCRIPCION	RUTA ENDPOINT
LISTA DE 20 PELICULAS.	http://localhost:4000/getmovies
LISTA DE GÉNEROS.	http://localhost:4000/filters/genres
LISTA DE PELICULAS QUE COINCIDAN CON EL FILTRO SOLICITADO (Acción en este caso).	http://localhost:4000/filters/genre/28
LISTA DE PELICULAS POPULARES	http://localhost:4000/list/popmovies
LISTA DE PROXIMOS ESTRENOS	http://localhost:4000/list/upcomings
LISTA DE PELICULAS MAS VALORADAS	http://localhost:4000/list/top
LISTA CON ENDPOINTS DE GENEROS DE PELICULAS	1. Acción (28): http://localhost:4000/filters/genre/28 2. Aventura (12): http://localhost:4000/filters/genre/12 3. Animación (16): http://localhost:4000/filters/genre/16 4. Comedia (35): http://localhost:4000/filters/genre/35 5. Crimen (80): http://localhost:4000/filters/genre/80 6. Documental (99): http://localhost:4000/filters/genre/99 7. Drama (18): http://localhost:4000/filters/genre/18 8. Familia (10751): http://localhost:4000/filters/genre/10751 9. Fantasía (14): http://localhost:4000/filters/genre/14 10. Historia (36): http://localhost:4000/filters/genre/36 11. Terror (27): http://localhost:4000/filters/genre/27 12. Música (10402): http://localhost:4000/filters/genre/10402 13. Misterio (9648): http://localhost:4000/filters/genre/9648 14. Romance (10749): http://localhost:4000/filters/genre/10749 15. Ciencia ficción (878): http://localhost:4000/filters/genre/10770 17. Suspense (53): http://localhost:4000/filters/genre/10770 17. Suspense (53): http://localhost:4000/filters/genre/53 18. Bélica (10752): http://localhost:4000/filters/genre/10752 19. Western (37): http://localhost:4000/filters/genre/10752
GENEROS DE SERIES DE TV	http://localhost:4000/filters/genre/10770 17. Suspense (53): http://localhost:4000/filters/genre/53 18. Bélica (10752): http://localhost:4000/filters/genre/10752

PAGINAS CON SERIES DE TV	1. http://localhost:4000/tv?page=1
	2. http://localhost:4000/tv?page=2
	3. http://localhost:4000/tv?page=3
	4. http://localhost:4000/tv?page=4
	5. http://localhost:4000/tv?page=5
	6. http://localhost:4000/tv?page=6
	7. http://localhost:4000/tv?page=7
	8. http://localhost:4000/tv?page=8
	9. http://localhost:4000/tv?page=9

RECURSOS / ITEMS: Ø

- POSTMAN: Como herramienta principal para probar las API
- Documentación del proyecto: Como fuente de guia para saber como usar las API.
- A DIFERENCIA de UI testing, (que serian mas casos de pruebas y mas pasos de prueba), el API TESTING podra generar menos casos de prueba y menos pasos de prueba.
- Test plan (SOLO EN JIRA CONFLUENCE) para planificacion de las pruebas

★RECURSOS DE SISTEMA: ②

HERRAMIENTAS DEL PROYECTO	DESCRIPCIÓN DEL USO
SLACK	Para comunicación fluida, corrida, actualización de temas, noticias, y más.
Jira con XRAY	Herramienta principal, gestor de incidencias para trabajar y seguir las tareas. CLICK PARA ACCEDER AL ESPACIO DE TRABAJO JIRA
CONFLUENCE	Herramienta de Documentación definitiva, Docs de Jira, Proyecto, Plans, etc. CLICK PARA ACCEDER A LA DOCUMENTACION EN JIRA CONFLUENCE
DISCORD	Para realizar las Planning, resolver problemas rápido, etc. CLICK PARA ACCESO A CANAL DE DISCORD
POSTMAN	Herramienta principal para probar las API CLICK PARA ACCESO AL ESPACIO DE TRABAJO EN POSTMAN (QA)

ullet TEST DELIVERY - LOS ENTREGABLES DE QA ${\mathscr O}$

(Tareas por realizar en cada etapa del Testing)

Entregables que deben ser realizado ANTES DE la fase de TEST EXECUTION: ${\mathscr O}$

- Documento de plan de pruebas
- Documento de casos de prueba

Entregables que deben ser realizado DURANTE la fase de TEST EXECUTION: 🔗

- Test Scripts
- Documento: Matriz de trazabilidad
- Error logs y logs de ejecucion.

Entregables que deben ser realizado DESPUÉS DE la fase de TEST EXECUTION: 🔗

- Resultados de test/reportes
- Reporte de defectos
- Instalacion/ Guias de procedimiento para los test