

Plano de Testes — Sistema Cinema (API + Front-End)

Data: 28/10/2025

Sistema sob teste: API do aplicativo Cinema

Autor: Lázaro Felipe Bezerra dos Santos

1. Apresentação

Este plano de testes tem como objetivo organizar e guiar a execução dos testes do Sistema Cinema, que compreende uma API REST para o gerenciamento de filmes e usuários, e um front-end web para interação com essas funcionalidades.

O foco dos testes está em validar as regras de negócio, a comunicação entre API e interface, e a consistência dos fluxos principais (autenticação, CRUD de filmes e exibição de dados).

2. Objetivo

Garantir que todas as funcionalidades críticas — tanto da API Cinema quanto da aplicação web — estejam funcionando corretamente, conforme os requisitos.

Os testes visam validar:

- Requisições e respostas da API (status codes, body, headers);
 - Integração entre front-end e back-end;
 - Fluxos de autenticação e autorização;
 - Validação de campos obrigatórios e regras de negócio;
 - Experiência e comportamento da interface.
-

3. Escopo

Incluído

- **API (Back-End):**
 - auth: autenticação e geração de token.
 - movies: criação, listagem, atualização e exclusão de filmes.
- **Front-End:**
 - Tela de login.
 - Tela de listagem de filmes.
 - Formulário de cadastro e edição de filmes.
 - Funcionalidades de exclusão e feedback ao usuário.

Excluído

- Integração com sistemas externos de catálogos.
 - Automação de testes visuais.
 - Testes de performance do servidor.
-

4. Análise

Regras de Negócio Principais

- Usuário deve autenticar-se para acessar operações restritas.
- Cada filme deve conter título, gênero, ano e diretor obrigatoriamente.
- Não é permitido criar filmes duplicados.
- Requisições sem token válido devem retornar 401 Unauthorized.
- A interface deve validar campos antes de enviar dados para a API.
- Mensagens de erro e sucesso devem ser exibidas de forma clara no front.

Riscos Identificados

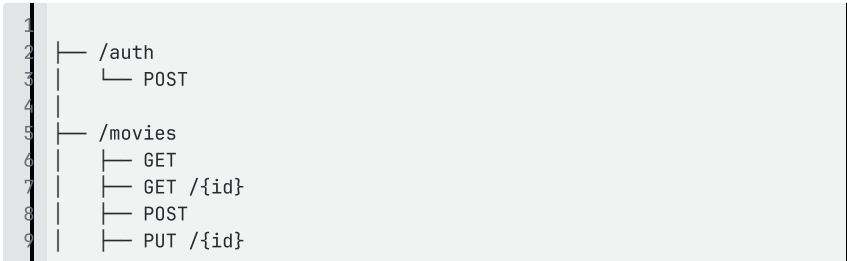
Risco	Impacto	Ação de Mitigação
Token expirado impede operações	Alto	Renovar token antes dos testes principais
Filme duplicado não bloqueado	Médio	Criar casos negativos específicos
Inconsistência entre API e front (dados divergentes)	Alto	Testar integração com dados mock e reais
Falha na exibição de mensagens de erro no front	Médio	Verificar retorno da API e renderização na interface
Tempo de resposta acima de 2s	Médio	Monitorar com asserts de performance

5. Técnicas Aplicadas

- Caixa-preta (funcional e integração)
- Teste de API com Robot Framework (HTTP requests)
- Teste de interface com SeleniumLibrary
- Testes baseados em risco
- Validação de contrato JSON

6. Mapa Mental da Aplicação

Sistema Cinema



```
10 |   └─ DELETE /{id}
11 |
12 | └─ Front-End
13 |   └─ Login
14 |     └─ POST /auth
15 |   └─ Listagem
16 |     └─ GET /movies
17 |   └─ Cadastro
18 |     └─ POST /movies
19 |   └─ Edição
20 |     └─ PUT /movies/{id}
21 |   └─ Exclusão
22 |     └─ DELETE /movies/{id}
23 |
24 |
```

7. Cenários de Teste Planejados

Back-End (API)

1. Login com credenciais válidas retorna token.
2. Login com credenciais inválidas retorna 401.
3. Criar filme com dados válidos (201 Created).
4. Impedir criação sem token (401).
5. Impedir criação com título duplicado (400).
6. Listar filmes (GET /movies) retorna 200.
7. Buscar filme por ID válido.
8. Buscar filme inexistente retorna 404.
9. Atualizar filme válido.
10. Impedir atualização sem token.
11. Excluir filme válido.
12. Impedir exclusão sem token.

Front-End (Interface Web)

13. Login via tela redireciona corretamente.
14. Exibição de mensagem de erro ao login inválido.
15. Exibição de lista de filmes após autenticação.
16. Cadastro de novo filme via formulário.
17. Validação de campos obrigatórios no formulário.
18. Edição de filme existente.
19. Exclusão de filme com confirmação.
20. Exibição de mensagens de sucesso e erro.
21. Verificação de integração com API (status code correto).

8. Priorização da Execução

Prioridade	Casos	Descrição
------------	-------	-----------

Alta (P0)	1, 3, 6, 13, 15	Login, CRUD principal e listagem de filmes
Média (P1)	4, 5, 7, 9, 16, 17, 18	Validação de duplicidade e edição
Baixa (P2)	2, 8, 10, 12, 14, 19, 20, 21	Casos negativos e mensagens visuais

9. Matriz de Risco

Risco	Impacto	Probabilidade	Nível
Token inválido	Alto	Alta	Crítico
Filme duplicado	Médio	Alta	Médio
Falha na integração front-back	Alto	Média	Alto
Tempo de resposta elevado	Médio	Média	Médio
Erros visuais na UI	Médio	Alta	Médio

10. Cobertura de Testes

- **Verbos HTTP:** GET, POST, PUT, DELETE.
- **Endpoints:** auth, movies.
- **Parâmetros:** IDs válidos e inválidos, corpo JSON com e sem campos obrigatórios.
- **Status Codes esperados:** 200, 201, 400, 401, 404.
- **Contrato JSON:** campos obrigatórios (id, title, year, genre, message, authorization).
- **Front-End validado:** Login, listagem, cadastro, edição e exclusão de filmes.