

PLANO DE TESTES - CINEMA APP

PLANO DE TESTES — Projeto de Automação “Cinema API & Front-End”

1. Identificação do Projeto

Projeto: CinemaApp – Automação de Testes

Responsável Técnico: Emanuel Felipe Avelino Silva

2. Apresentação

Este documento descreve o plano de testes completo para a aplicação Cinema, que integra uma API RESTful e um Front-End Web.

O objetivo é validar os fluxos críticos de autenticação, listagem de filmes, sessões e reservas, assegurando a qualidade funcional, integridade de dados e estabilidade da aplicação por meio de testes automatizados com Robot Framework.

A automação foi estruturada com padrões modulares, cobrindo camadas independentes (API e UI), permitindo fácil manutenção e expansão futura.

3. Objetivo dos Testes

Garantir que todos os módulos críticos da aplicação “Cinema” estejam funcionando conforme o esperado e atendam aos requisitos de negócio definidos.

Os testes têm como metas:

- Verificar a correção funcional de todos os endpoints da API.
 - Validar a interação completa do usuário via interface web.
 - Assegurar reutilização e padronização do código de teste.
 - Detectar falhas em cenários negativos e de exceção.
 - Prover relatórios automatizados e reproduzíveis de execução.
-

4. Escopo

4.1 Itens no Escopo

O projeto cobre os seguintes módulos e fluxos:

API:

- Autenticação (/auth)
- Filmes (/movies)
- Sessões (/sessions)
- Reservas (/reservations)
- Cenários negativos e erros genéricos (401, 404, 400)

UI:

- Login via interface web
- Visualização de filmes e detalhes
- Fluxo completo de reserva (login → filme → sessão → assento → pagamento)
- Fluxo de registro de usuário e logout

4.2 Itens Fora do Escopo

- Testes de performance e carga.
- Testes mobile.
- Funcionalidades administrativas (rotas “admin”).
- Integrações externas não disponibilizadas pelo ambiente atual.

5. Análise da Aplicação

A aplicação “Cinema” é composta por dois módulos integrados:

- **Back-End (API RESTful):** responsável pela autenticação, listagem de filmes, sessões, reservas e manipulação de dados.
- **Front-End (React/Browser):** interface que permite interação direta do usuário final, consumindo a API.

Fluxos críticos identificados:

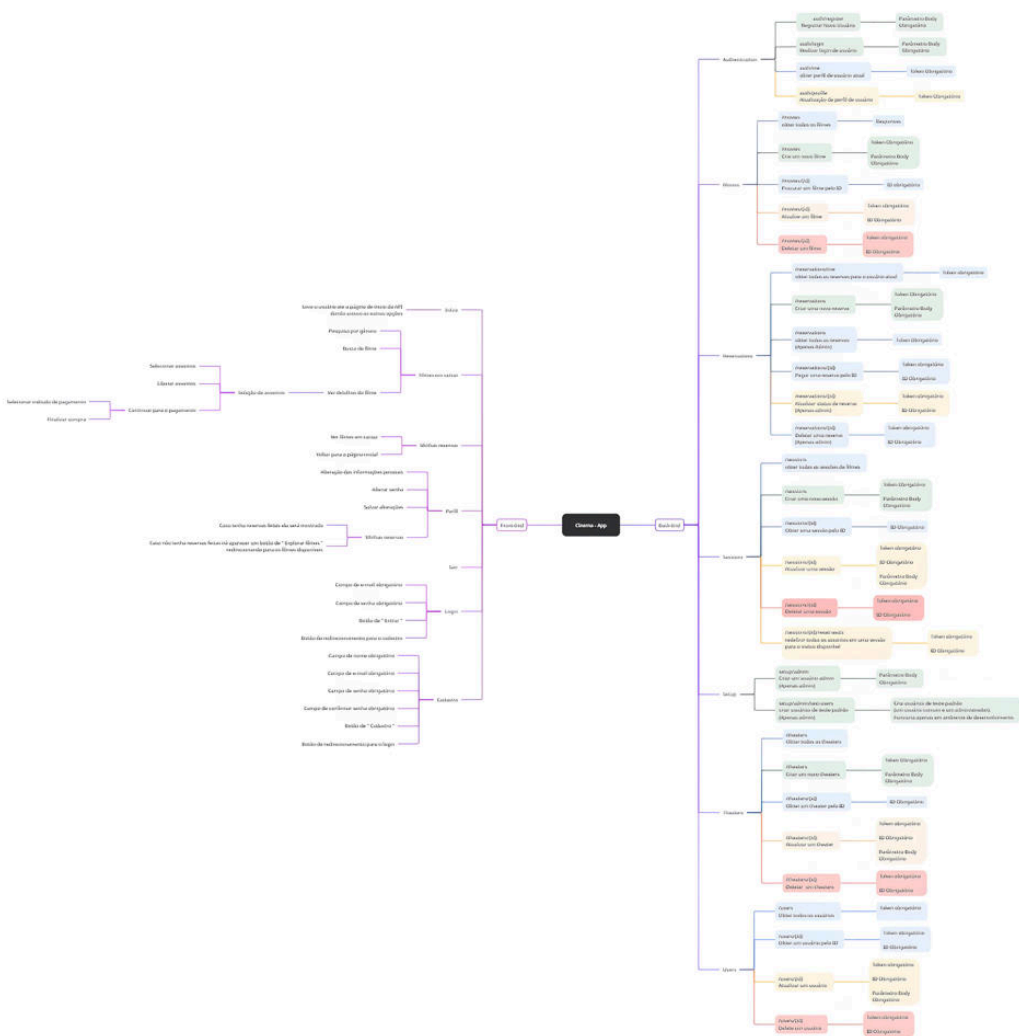
1. Autenticação (Login e Cadastro)
2. Listagem de Filmes e Sessões
3. Criação de Reservas
4. Fluxo E2E de reserva via interface

Esses fluxos foram priorizados por representarem o core funcional da aplicação.

6. Estratégia e Técnicas de Teste

Tipo de Teste	Técnica Aplicada	Objetivo Principal
Funcional (API)	Caixa-preta	Validar resposta e comportamento dos endpoints
Funcional (UI)	Caixa-preta com interação DOM	Validar comportamento visual e funcional da interface
Negativo	Validação de erro e exceção	Confirmar tratamento correto de falhas
Reusabilidade	Modularização com Keywords	Manter consistência e reduzir redundância
Validação de Dados	Comparação de JSON e HTTP Codes	Verificar integridade das respostas

7. Mapa Mental da Aplicação



8. Cenários de Teste Planejados

8.1 Autenticação (API)

ID	Cenário	Tipo	Resultado Esperado
API-A1	Login com credenciais válidas	Positivo	Retorna status 200 e token JWT válido
API-A2	Login com credenciais inválidas	Negativo	Retorna 401

API-A3	Acesso sem token	Negativo	Retorna 401
---------------	------------------	----------	-------------

8.2 Filmes (API)

ID	Cenário	Tipo	Resultado Esperado
API-M1	Listar todos os filmes	Positivo	Retorna lista com status 200
API-M2	Buscar filme por ID válido	Positivo	Retorna detalhes do filme
API-M3	Buscar filme inexistente	Negativo	Retorna 404

8.3 Sessões (API)

ID	Cenário	Tipo	Resultado Esperado
API-S1	Listar todas as sessões	Positivo	Retorna lista (200)
API-S2	Buscar sessão por ID	Positivo	Retorna dados da sessão
API-S3	Listar sessões por filme	Positivo	Retorna apenas sessões do filme
API-S4	Buscar sessão inexistente	Negativo	Retorna 404

8.4 Reservas (API)

ID	Cenário	Tipo	Resultado Esperado
----	---------	------	--------------------

API-R1	Criar reserva válida	Positivo	Retorna 201 e dados da reserva
API-R2	Criar reserva com assento ocupado	Negativo	Retorna 400
API-R3	Criar reserva sem token	Negativo	Retorna 401
API-R4	Listar reservas do usuário	Positivo	Retorna 200 com lista de reservas

8.5 UI – Interface Web

ID	Cenário	Tipo	Resultado Esperado
UI-L1	Login com credenciais válidas	Positivo	Redireciona à página inicial
UI-M1	Exibir lista de filmes	Positivo	Filmes renderizados corretamente
UI-M2	Clicar em filme e abrir detalhes	Positivo	Exibe informações completas
UI-B1	Fluxo completo de reserva	Positivo	Reserva finalizada com sucesso
UI-R1	Fluxo completo de cadastro, logout e novo login	Positivo	Cadastro e autenticação concluídos

8.6 Testes Negativos Gerais

ID	Cenário	Tipo	Resultado Esperado
NEG-1	Buscar recurso inexistente	Negativo	Retorna 404
NEG-2	Enviar payload inválido	Negativo	Retorna 400
NEG-3	Token inválido	Negativo	Retorna 401

9. Priorização da Execução

Prioridade	Módulo	Justificativa
P1 (Alta)	Autenticação / Sessões / Reservas	Fluxos críticos do negócio
P2 (Média)	Filmes / UI Visual	Complementam o fluxo principal
P3 (Baixa)	Negativos gerais	Ampliam robustez e validação de erros

10. Matriz de Risco

Risco	Probabilidade	Impacto	Mitigação
Instabilidade no servidor	Alta	Alta	Garantir ambiente ativo antes da execução
Mudança nos endpoints	Média	Alta	Centralizar URLs em arquivo de variáveis

Falhas no front-end	Média	Média	Adicionar waits dinâmicos
Autenticação inválida	Baixa	Alta	Criar keyword de login reutilizável
Ausência de dados no banco	Alta	Média	Pré-popular dados antes dos testes

11. Cobertura de Testes

Módulo	Cenários Cobertos
Autenticação	3
Filmes	3
Sessões	4
Reservas	4
Negativos	3
UI	5
Total	22

12. Ferramentas e Estrutura Técnica

Categoria	Ferramenta
Framework de Automação	Robot Framework
Bibliotecas	RequestsLibrary, Browser, Collections
Linguagem Base	Robot Syntax
Controle de Versão	Git / GitHub

13. Critérios de Aceitação

- 100% dos testes executando com sucesso.
 - Relatórios gerados corretamente em /results_priority1/.
 - Estrutura modular e reutilizável.
 - Scripts executáveis em qualquer ambiente local com servidor ativo.
-

14. Boas Práticas e Inovações

- Separação lógica entre testes API e UI.
 - Modularização via Resources e Keywords.
 - Uso de variáveis dinâmicas e waits adaptativos.
 - Automação com validação de resposta HTTP e JSON.
 - Documentação e relatórios claros e versionados no GitHub.
-

15. Conclusão

A automação do projeto *Cinema* assegura uma cobertura robusta e confiável dos fluxos principais da aplicação.

Com a estrutura atual, os testes garantem alta rastreabilidade, fácil manutenção e reprodutibilidade, demonstrando boas práticas de QA e automação profissional.

O sistema encontra-se validado funcionalmente, com base nos requisitos propostos e execução completa sem falhas.