# Plano de Testes – API Cinema App

Projeto: Cinema App Backend | Cinema App Frontend

Versao: 1.0.0

Autor: @Thais Nogueira

Data: 23 de jun. de 2025 - 4 de jul. de 2025

- 1. Introdução
- 2. Objetivo
- 3. Escopo dos Testes
  - Excluído do Escopo
- 4. Referências
- 5. Estratégia de Teste
- 6. Mapa Mental
- 7.Técnicas Aplicadas
- 8. Ánalise de teste
- 9. Cenario de teste
  - 9.1 Cenario de teste API
  - Autenticação
  - Filmes
  - Reservas
  - Sessoes
  - Theaters
  - Users
  - 9.2 Cenario de teste WEB
  - Autenticação
  - Filmes
  - Reservas
  - Sessoes
- 10. Matriz de Rastreabilidade
- 11. Matriz de Risco
- 12. Priorização da execução dos cenários de teste
- 13. Cobertura de Testes
- Manual
- Automatizado API
- Automatizado WEB
- 14. Testes Candidatos à Automação
  - 14.1 Automação Back
  - o 14.2 Automação Web
- Observações Finais:

# 1. Introdução 🖉

A aplicação Cinema App composto por uma API e um Front-end web que permite a usuários visualizar filmes em cartaz, selecionar sessões, reservar assentos e gerenciar seus dados e histórico de reservas.

Este plano documenta os testes funcionais, exploratórios realizados com base nas Histórias de Usuário, documentação Swagger e resultados do teste exploratório. O objetivo é validar requisitos, identificar bugs e apoiar a melhoria contínua da qualidade.

# 2. Objetivo 🖉

Garantir a qualidade funcional e a conformidade com as Histórias de Usuário da API e do Front-end do Cinema App através da execução de testes abrangentes e automatizáveis.

#### **Objetivos Secundários:**

- Validar os Critérios de Aceitação de todas as Histórias de Usuário
- Validar os requisitos funcionais.
- Garantir que os fluxos complexos do sistema sejam executados corretamente.
- Automatizar os cenários de teste para garantir testes recorrentes e confiáveis.

# 3. Escopo dos Testes 🕖

Incluido:

#### **API (Backend)**

- Autenticação:
  - o Registro de usuário
  - o Login e emissão de token JWT
  - o Consulta e atualização do perfil do usuário
- Gestão de Filmes:
  - o Listagem pública de filmes
  - CRUD completo de filmes
- Gestão de Sessões:
  - Listagem pública de sessões
  - o CRUD completo de sessões
  - Reset de assentos
- Gestão de Teatros:
  - Listagem pública de teatros
  - CRUD completo de teatros
- Reservas:
  - o Criação de reservas com seleção de assentos
  - o Consulta de histórico de reservas pessoais
  - o Restrições relacionadas à disponibilidade e integridade dos dados
- Gestão de Usuários:
  - o Listagem de usuários
  - o Edição e exclusão de usuários
  - o Restrições relacionadas a usuários com reservas ativas

#### Frontend (Web)

- Fluxo completo de registro e login do usuário
- Visualização de filmes em cartaz
- Visualização de detalhes do filme e sessões disponíveis
- Seleção de assentos e processo de reserva
- Visualização do histórico de reservas do usuário
- Visualização e edição do perfil do usuário
- Navegação entre páginas

- Exibição e validação de mensagens de erro e sucesso
- Restrições visuais e funcionais baseadas em permissões.

## Excluído do Escopo 🖉

- Testes de performance
- Testes de segurança avançados
- Integrações externas não detalhadas
- Testes de usabilidade profunda
- Testes em dispositivos móveis específicos

## 4. Referências @

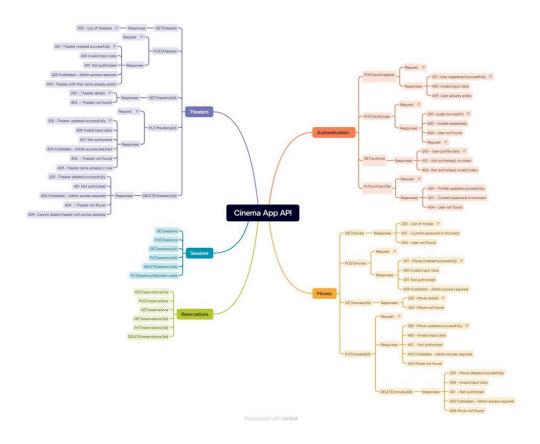
- Histórias de Usuário: Documento fornecido
- Documentação swagger: http://localhost:3000/api/v1/docs/
- Repositório Back-end: 🔾 GitHub juniorschmitz/cinema-challenge-back
- Repositório Front-end: 🖸 GitHub juniorschmitz/cinema-challenge-front
- Teste exploratorio: 🗉 Teste Exploratório Cinema App
- Historia do usuario
- Robot Framework, Postman: Ferramentas de execução de testes

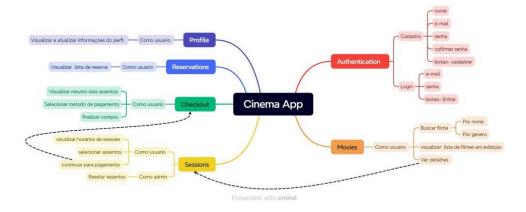
## 5. Estratégia de Teste 🖉

A abordagem se baseia em testes funcionais derivados diretamente das Histórias de Usuário e seus Critérios de Aceitação. Serão combinados testes de API, Front-end e fluxos integrados, visando:

- Validação das regras de negócio
- Garantia da experiência do usuário conforme o esperado
- Detecção de falhas tanto em casos positivos quanto negativos
- Automação total dos cenários, priorizando fluxos críticos

## 6. Mapa Mental @





# 7.Técnicas Aplicadas 🕖

- Particionamento de Equivalência:E-mail válido/inválido, assentos disponíveis/ocupado
- Análise de Valor Limite: senhas mínimas, duração de filme, capacidade de teatro
- Testes Negativos: Ações sem permissão, dados inválidos, endpoints com tokens ausentes
- Testes Baseados em Cenários: : Fluxos ponta a ponta
- Exploratórios: Navegação, erros não documentados

## 8. Ánalise de teste @

Os cenários de teste serão identificados com base:

- Das Histórias de Usuário e Critérios de Aceitação
- Da documentação da API e interface do Front-end
- De sessões exploratórias para identificar lacunas de cobertura

#### 8.1 Autenticação

Funcionalidades: Registro, Login, Logout, Perfil

#### Regras de Negócio

- E-mail único e válido
- Senha com tamanho mínimo
- JWT obrigatório para endpoints protegidos

#### Fluxos Principais:

- Registro e login com dados válidos
- Visualização e edição de perfil autenticado

#### Cenários Alternativos e Inválidos

- Registro com e-mail existente
- Senha abaixo do permitido
- Login com dados inválidos
- · Acesso sem token ou com token expirado

#### 8.2 Filmes

Funcionalidades: Listagem, Detalhes, CRUD (Admin)

#### Regras de Negócio:

- Apenas Admin pode criar/editar/deletar
- Título obrigatório e único
- Duração não pode ser negativa

### Fluxos Principais:

- Visualizar lista de filmes
- Acesso à página de detalhes do filme

#### Cenários Alternativos e Inválidos:

- Criar filme com título vazio ou duração negativa
- Tentar CRUD sem permissão

#### 8.3 Sessoes

Funcionalidades: Listagem, CRUD (Admin), Reset de assentos

### Regras de Negócio:

- Vinculadas a Filme e Teatro existentes
- Proibido criar sessões duplicadas ou com data/hora no passado
- Proibido excluir sessão com reservas ativas

### Fluxos Principais:

- Seleção de um filme
- Visualização dos horários, datas e disponibilidade

#### Cenários Alternativos e Inválidos:

- Sessão em horário inválido
- Reset de assentos por usuário comum

### 8.4 Reserva

Funcionalidades: Seleção de assentos, Checkout, Histórico

### Regras de Negócio:

- Apenas usuários autenticados podem reservar
- Proibido reservar assentos já ocupados
- Visualização do histórico pessoal

#### Fluxos Principais:

- Selecionar assentos disponíveis
- Finalizar compra
- Visualizar histórico de reservas

### Cenários Alternativos e Inválidos:

- Seleção de assentos já reservados
- Checkout sem assentos
- Acesso a reservas sem login

#### 8.5 Teatros

Funcionalidades: Listagem, CRUD (Admin)

## Regras de Negócio:

- Nome único
- Capacidade positiva e realista
- Proibido excluir teatro com sessões ativas

### 8.6 Usuários (Admin)

Funcionalidades: Listagem, Edição, Exclusão

### Regras de Negócio:

- Apenas Admin pode gerenciar
- Proibido excluir usuário com reservas ativas

# 9. Cenario de teste 🛭

# 9.1 Cenario de teste API 🖉

# Autenticação 🔗

ID	Ti po	Caso de Teste	Pré- condição	Passos	Payload	Resulta do Esperad o	Priori dade
CT- 001	AP I	Cadastro de usuário válido	API disponível e ambiente configurad o	1. Definir payload com nome, e-mail e senha válidos2. Enviar POST /auth/register 3. Validar status code 2014. Validar mensagem de sucesso	<pre>{ "name": "teste", "email": "teste@teste.com", "password": "teste123" }</pre>	201 - User registere d successf ully	Alta
CT- 002	AP I	Cadastro com e-mail já existente	Usuário já cadastrado	Definir payload com e-mail já existente2. Enviar POST /auth/register 3. Validar status code 4004. Validar mensagem "User already exists"	{ "name": "teste", "email": "teste@teste.com", "password": "teste123" }	400 - User already exists	Media
CT- 003	AP I	Cadastro com e-mail inválido	API disponível e ambiente configurad o	1. Definir payload com e-mail inválido2. Enviar POST /auth/register 3. Validar status code 4004. Validar mensagem "Invalid email format"  1. Definir payload com e-mail inválido email in	{ "name": "teste", "email": "emailinvalido.com", "password": "teste123" }	400 - Invalid email format	Media
CT- 004	AP I	Cadastro com senha abaixo do mínimo	API disponível e ambiente configurad o	1. Definir payload com senha menor que o permitido2. Enviar POST /auth/register 3. Validar status code 4004. Validar mensagem "Password too short"	{ "name": "teste", "email": "teste2@teste.com", "password": "123" }	400 - Passwor d too short	Media
CT- 005	AP I	Cadastro com campos vazios	API disponível e ambiente configurad o	Definir payload com campos     vazios2. Enviar POST     /auth/register 3. Validar     status code 4004. Validar     mensagem "Missing required     fields"	{ "name": "", "email": "", "password": "" }	400 - Missing required fields	Alta

CT- 006	AP I	Cadastro faltando campo e- mail	API disponível e ambiente configurad o	Definir payload sem o campo e-mail2. Enviar POST /auth/register 3. Validar status code 4004. Validar mensagem "Missing required fields"	{ "name": "teste", "password": "teste123" }	400 - Missing required fields	Alta
CT- 007	AP I	Login com dados válidos	Usuário cadastrado	Definir payload com e-mail e senha válidos2. Enviar POST /auth/login 3. Validar status code 2004. Validar mensagem de sucesso	<pre>{ "email": "teste@teste.com", "password": "teste123" }</pre>	200 - Login successf ul	Alta
CT- 008	AP I	Login com e- mail não cadastrado	API disponível e ambiente configurad o	1. Definir payload com e-mail não existente2. Enviar POST /auth/login 3. Validar status code 4004. Validar mensagem "Invalid credentials"	{ "email": "naoexiste@teste.com", "password": "teste123" }	400 - Invalid credenti als	Alta
CT- 009	AP I	Login com senha incorreta	Usuário cadastrado	1. Definir payload com e-mail correto e senha incorreta2. Enviar POST /auth/login 3. Validar status code 4004. Validar mensagem "Invalid credentials"	{ "email": "teste@teste.com", "password": "senhaerrada" }	400 - Invalid credenti als	Alta
CT- 010	AP I	Login com e- mail inválido (formato)	API disponível e ambiente configurad o	1. Definir payload com e-mail inválido 2. Enviar POST /auth/login 3. Validar status code 4004. Validar mensagem "Invalid email format"	{ "email": "teste.teste.com", "password": "teste123" }	400 - Invalid email format	Med
CT- 011	AP I	Login com campos vazios	API disponível e ambiente configurad o	Definir payload com campos vazios2. Enviar POST /auth/login 3. Validar status code 4004. Validar mensagem "Missing required fields"	{ "email": "", "password": "" }	400 - Missing required fields	Alta
CT- 012	AP I	Obter perfil do usuário autenticado	Usuário autenticad o	Enviar GET /auth/me com token válido2. Validar status code 2003. Validar que os dados do usuário sejam retornados	N/A	200 - User profile data	Alta
CT- 013	AP I	Obter perfil sem token	API disponível e ambiente configurad o	1. Enviar GET /auth/me sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authoriz ed	Alta
CT- 014	AP I	Obter perfil	Token expirado	Enviar GET /auth/me com token expirado2. Validar	N/A	401 - Token	Alta

		expirado	gerado previament e	status code 4013. Validar mensagem "Token expired"		expired	
CT- 015	AP I	Atualizar perfil com dados válidos	Usuário autenticad o	Definir payload com nome atualizado2. Enviar PUT /auth/profile com token válido3. Validar status code 2004. Validar mensagem de sucesso	<pre>"name": "teste",    "currentPassword": "teste123",    "newPassword": "senha123" }</pre>	200 - Profile updated successf ully	Alta
CT- 016	AP I	Atualizar perfil com token inválido	API disponível e ambiente configurad o	1. Definir payload com nome atualizado2. Enviar PUT /auth/profile com token inválido3. Validar status code 4034. Validar mensagem "Not authorized"	<pre>"name": "teste",    "currentPassword": "teste123",    "newPassword": "senha123" }</pre>	403 - Not authoriz ed	Alta
CT- 017	AP I	Atualizar perfil com campos vazios	Usuário autenticad o	1. Definir payload com campos vazios2. Enviar PUT /auth/profile 3. Validar status code 4004. Validar mensagem "Missing required fields"	<pre>"name": "teste",   "currentPassword": "teste123",   "newPassword": "senha123" }</pre>	400 - Missing required fields	Media
CT- 018	AP I	Atualizar perfil com payload malformado	Usuário autenticad o	1. Enviar PUT /auth/profile com payload malformado (ex: texto simples)2. Validar status code 4003. Validar mensagem "Invalid input data"	"nome"	400 - Invalid input data	Media

## Filmes @

ID	Tip o	Caso de Teste	Pré- condição	Passos	Payload	Resultado Esperado	Prior idad e
CT- 019	API	Listagem de todos os filmes	Filmes cadastrados	<ol> <li>Enviar GET /movies sem autenticação</li> <li>2. 2. Validar status code 200</li> <li>3. 3. Validar que a lista de filmes seja exibida</li> </ol>	N/A	200 - List of movies	Alta

CT- 020	API	Criação de filme com dados válidos	Usuário Admin autenticado	<ol> <li>Definir payload de filme válido</li> <li>2. Enviar POST /movies com token válido</li> <li>Validar status code 201</li> <li>Validar mensagem de sucesso</li> </ol>	<pre>"title": "Como treinar o seu dragao",     "synopsis": "esta é a sinopse do novo filme",     "director": "nome do diretor",     "genres": [         "Animation"  ],     "classification": "livre",     "poster": "poster",     "releaseDate": "2025-06- 26" }</pre>	201 - Movie created successfully	Alta
CT- 021	API	Criação de filme com título em branco	Usuário Admin autenticado	1. Definir payload com título em branco2. Enviar POST /movies 3. Validar status code 4004. Validar mensagem "Invalid input data"	{ "title": "" }	400 - Invalid input data	Medi a
CT- 022	API	Criação de filme com duração negativa	Usuário Admin autenticado	Definir payload com     duração negativa2. Enviar     POST /movies 3. Validar     status code 4004. Validar     mensagem "Invalid input     data"	"duration": -1,	400 - Invalid input data	Medi a
CT- 023	API	Criação de filme sem autenticaçã o	API disponível	1. Definir payload de filme válido2. Enviar POST /movies sem token3. Validar status code 4014. Validar mensagem "Not authorized"	{ "title": "Filme Teste", "duration": 120 }	401 - Not authorized	Alta
CT- 024	API	Criação de filme como usuário comum	Usuário comum autenticado	1. Definir payload de filme válido2. Enviar POST /movies com token de usuário comum3. Validar status code 4034. Validar mensagem "Forbidden - Admin access required"	<pre>"title": "Como treinar o seu dragao",     "synopsis": "esta é a sinopse do novo filme",     "director": "nome do diretor",     "genres": [</pre>	403 - Forbidden - Admin access required	Alta

0.7	ADI	Occupation	Films	1. Frankright (C. 1)	<pre>"duration": 150,  "classification": "livre",  "poster": "poster",  "releaseDate": "2025-06- 26" }</pre>	200 Maria	Alban
CT- 025	API	Consulta de filme por ID válido	Filme existente cadastrado	Enviar GET /movies/{id}     com ID válido2. Validar     status code 2003. Validar     que os dados do filme     sejam retornados	N/A	200 - Movie details	Alta
CT- 026	API	Atualização de filme com dados válidos	Usuário Admin autenticado, filme existente	1. Definir payload de atualização válido2. Enviar PUT /movies/{id} com token válido3. Validar status code 2004. Validar mensagem de sucesso	{ "title": "Filme Atualizado", "duration": 130 }	200 - Movie updated successfully	Alta
CT- 027	API	Atualização de filme com ID inexistente	Usuário Admin autenticado	1. Definir payload de atualização válido2. Enviar PUT /movies/{id} com ID inexistente3. Validar status code 4044. Validar mensagem "Movie not found"	ID: 6861274f949db48d198bd97b	404 - Movie not found	Med a
CT- 028	API	Exclusão de filme com sucesso	Usuário Admin autenticado, filme existente	1. Enviar DELETE /movies/{id} com token válido2. Validar status code 2003. Validar mensagem "Movie deleted successfully"	N/A	200 - Movie deleted successfully	Alta
CT- 029	API	Exclusão de filme sem autenticaçã o	API disponível	1. Enviar DELETE /movies/{id} sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authorized	Alta
CT- 030	API	Exclusão de filme como usuário comum	Usuário comum autenticado	1. Enviar DELETE /movies/{id} com token de usuário comum2. Validar status code 4033. Validar mensagem	N/A	403 - Forbidden - Admin access required	Alta

	"Forbidden - Admin		
	access required"		

## Reservas $\mathscr O$

ID	Ti po	Caso de Teste	Pré-condição	Passos	Payload	Resultado Esperado	Prior idad e
CT- 031	AP I	Obter todas as reservas do usuário com sucesso	Usuário autenticado com reservas cadastradas	Enviar GET /reservations/me     com token válido2. Validar status     code 2003. Validar que a lista de     reservas do usuário seja exibida     corretamente	N/A	200 - User's reservations	Alta
CT- 032	AP I	Obter todas as reservas do usuário sem autenticação	API disponível	Enviar GET /reservations/me sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authorized	Alta
CT- 033	AP I	Criação de reserva válida	Usuário autenticado, sessão existente, assentos disponíveis	Definir payload de reserva com assentos disponíveis2. Enviar      POST /reservations com token válido3. Validar status code 2014. Validar mensagem de sucesso	{ "session":  "idSessao",  "seats": [{  "row": "A",  "number": 1,  "type":  "full" }],  "paymentMetho d":  "credit_card" }	201 - Reservation created successfully	Alta
CT- 034	AP I	Criação de reserva com assento já ocupado	Usuário autenticado, sessão existente, assento ocupado	Definir payload de reserva com assento já ocupado2. Enviar     POST /reservations 3. Validar status code 4004. Validar mensagem "Seats already taken"	{ "session":  "idSessao",  "seats": [{  "row": "A",  "number": 1,  "type":  "full" }],  "paymentMetho d":  "credit_card" }	400 - Seats already taken	Alta
CT- 035	AP I	Criação de reserva sem autenticação	Sessão existente	1. Definir payload de reserva válida2. Enviar POST /reservations sem token3. Validar status code 4014. Validar mensagem "Not authorized"	{ "session": "idSessao", "seats": [{ "row": "A", "number": 1, "type":	401 - Not authorized	Alta

					"full" }],  "paymentMetho d":  "credit_card" }		
CT- 036	AP I	Criação de reserva para sessão inexistente	Usuário autenticado	Definir payload de reserva com     ID de sessão inexistente2. Enviar     POST /reservations 3. Validar     status code 4044. Validar     mensagem "Session not found"	{ "session":   "idInexistent   e", "seats":   [{ "row":   "A",   "number": 1,   "type":   "full" }],   "paymentMetho   d":   "credit_card" }	404 - Session not found	Alta
CT- 037	AP I	Consulta de reserva por ID válido	Reserva existente e usuário autenticado	Enviar GET /reservations/{id}     com ID válido e token2. Validar     status code 2003. Validar     detalhes da reserva	N/A	200 - Reservation details	Alta
CT- 038	AP I	Atualizar status da reserva com sucesso	Reserva existente, usuário admin autenticado	Definir payload com novos status válidos2. Enviar PUT /reservations/{id} com token de admin3. Validar status code 2004. Validar mensagem de sucesso	{ "status": "confirmed", "paymentStatu s": "completed" }	200 - Reservation updated successfully	Alto
CT- 039	AP I	Atualizar status da reserva sem autenticação	Reserva existente	1. Definir payload com novos status válidos2. Enviar PUT /reservations/{id} sem token3. Validar status code 4014. Validar mensagem "Not authorized"	<pre>{ "status":   "confirmed",   "paymentStatu s":   "completed" }</pre>	401 - Not authorized	Alto
CT- 040	AP I	Atualizar status da reserva como usuário comum	Reserva existente, usuário comum autenticado	1. Definir payload com novos status válidos2. Enviar PUT  /reservations/{id} com token de usuário comum3. Validar status code 4034. Validar mensagem "Forbidden - Admin access required"	{ "status": "confirmed", "paymentStatu s": "completed" }	403 - Forbidden - Admin access required	Alta
CT- 041	AP I	Atualizar status da reserva com status inválido	Reserva existente, usuário admin autenticado	1. Definir payload com status inválido2. Enviar PUT /reservations/{id} com token de admin3. Validar status code 4004. Validar mensagem "Invalid status transition"	<pre>{ "status":  "invalid",  "paymentStatu s":  "completed" }</pre>	400 - Invalid status transition	Med
CT- 042	AP I	Exclusão de reserva	Reserva existente e usuário admin	1. Enviar DELETE /reservations/{id} com token	N/A	200 - Reservation	Alto

		como admin (válido)	autenticado	de admin2. Validar status code 2003. Validar mensagem "Reservation deleted successfully"		deleted successfully	
CT- 043	AP I	Exclusão de reserva como usuário comum	Reserva existente, usuário comum autenticado	1. Enviar DELETE  /reservations/{id} com token  de usuário comum2. Validar  status code 4033. Validar  mensagem "Forbidden - Admin  access required"	N/A	403 - Forbidden - Admin access required	Alta

## Sessoes @

ID	Tipo	Caso de Teste	Pré-condição	Passos	Payload	Resultado Esperado	Prioridade
CT- 044	API	Listagem pública de sessões	Sessões cadastradas	1. Enviar GET /sessions sem autenticação2. Validar status code 2003. Validar que a lista de sessões seja exibida corretamente	N/A	200 - List of sessions	Alta
CT- 045	API	Criação de sessão válida	Usuário admin autenticado, filme e teatro existentes	1. Definir payload de sessão válida2. Enviar POST /sessions com token de admin3. Validar status code 2014. Validar mensagem de sucesso	{ "movie": "idFilme", "theater": "idTeatro", "datetime": "2025- 07-01T20:00:00Z", "fullPrice": 20, "halfPrice": 10 }	201 - Session created successfully	Alta
CT- 046	API	Criação de sessão com dados duplicados	Usuário admin autenticado, sessão já existente	1. Definir payload igual a sessão já cadastrada2. Enviar POST /sessions 3. Validar status code 4004. Validar mensagem "Duplicate session not allowed"	{ "movie":  "idFilme",  "theater":  "idTeatro",  "datetime": "2025- 07-01T20:00:00Z",  "fullPrice": 20,  "halfPrice": 10 }	400 - Duplicate session not allowed	Alta
CT- 047	API	Criação de sessão com	Usuário admin autenticado	Definir payload     com data no     passado2. Enviar	{ "movie": "idFilme", "theater":	400 - Invalid datetime	Media

		data/hora no passado		POST /sessions 3. Validar status code 4004. Validar mensagem "Invalid datetime"	"idTeatro",  "datetime": "2022- 01-01T20:00:00Z",  "fullPrice": 20,  "halfPrice": 10 }		
CT- 048	API	Criação de sessão com filme inexistente	Usuário admin autenticado	1. Definir payload com ID de filme inexistente2. Enviar POST /sessions 3. Validar status code 4044. Validar mensagem "Movie not found"	{ "movie":  "idInexistente",  "theater":  "idTeatro",  "datetime": "2025- 07-01T20:00:00Z",  "fullPrice": 20,  "halfPrice": 10 }	404 - Movie not found	Alta
CT- 049	API	Consulta de sessão por ID válido	Sessão existente	1. Enviar GET /sessions/{id} com ID válido2. Validar status code 2003. Validar que os dados da sessão sejam exibidos corretamente	N/A	200 - Session details	Alta
CT- 050	API	Consulta de sessão com ID inexistente	ID inexistente	1. Enviar GET /sessions/{id} com ID inexistente2. Validar status code 4043. Validar mensagem "Session not found"	id: 96861274f949db48d19 8bd97b7b	404 - Session not found	Media
CT- 051	API	Atualização horario sessão com dados válidos	Sessão existente, usuário admin autenticado	1. Definir payload de atualização válida2. Enviar PUT /sessions/{id} com token de admin3. Validar status code 2004. Validar mensagem de sucesso	{ "datetime": "2025- 08-01T20:00:00Z", "fullPrice": 25, "halfPrice": 12 }	200 - Session updated successfully	Alta
CT- 052	API	Exclusão de sessão válida	Sessão existente, sem reservas, admin autenticado	1. Enviar DELETE /sessions/{id} com token de admin2. Validar status code 2003. Validar mensagem	N/A	200 - Session deleted successfully	Alta

				"Session deleted successfully"			
CT- 053	API	Exclusão de sessão com reservas ativas	Sessão com reservas, admin autenticado	1. Enviar DELETE /sessions/{id} com token de admin2. Validar status code 4003. Validar mensagem "Cannot delete session with active reservations"	N/A	400 - Cannot delete session with active reservations	Alta
CT- 054	API	Exclusão de sessão sem autenticação	Sessão existente	1. Enviar DELETE /sessions/{id} sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authorized	Alta
CT- 055	API	Exclusão de sessão como usuário comum	Sessão existente, usuário comum autenticado	1. Enviar DELETE /sessions/{id} com token de usuário comum2. Validar status code 4033. Validar mensagem "Forbidden - Admin access required"	N/A	403 - Forbidden - Admin access required	Alta
CT- 056	API	Resetar assentos com sucesso	Sessão existente, admin autenticado	1. Enviar PUT /sessions/{id}/r eset-seats com token de admin2. Validar status code 2003. Validar mensagem "Seats reset successfully"	N/A	200 - Seats reset successfully	Alta
CT- 057	API	Resetar assentos sem autenticação	Sessão existente	1. Enviar PUT /sessions/{id}/r eset-seats sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authorized	Alta

API	Resetar	Sessão	1. Enviar PUT	N/A	403 -	Alta
	assentos	existente,	/sessions/{id}/r		Forbidden -	
	como usuário	usuário	eset-seats com		Admin access	
	comum	comum	token de usuário		required	
		autenticado	comum2. Validar			
			status code 4033.			
			Validar mensagem			
			"Forbidden -			
			Admin access			
			required"			
		como usuário	como usuário usuário comum	como usuário usuário eset-seats com token de usuário comum token de usuário comum2. Validar status code 4033.  Validar mensagem "Forbidden - Admin access	como usuário usuário eset-seats com comum token de usuário comum2. Validar status code 4033. Validar mensagem "Forbidden - Admin access	como usuário usuário eset-seats com Admin access comum token de usuário required comum2. Validar status code 4033.  Validar mensagem "Forbidden - Admin access

## Theaters $\mathscr Q$

ID	Tip o	Caso de Teste	Pré-condição	Passos	Payload	Resultado Esperado	Prioridade
CT- 059	API	Listagem todas as salas	Teatros cadastrados	1. Enviar GET /theaters sem autenticação2. Validar status code 2003. Validar que a lista de teatros seja exibida corretamente	N/A	200 - List of theaters	Alta
CT- 060	API	Criação de teatro válida	Usuário admin autenticado	1. Definir payload de teatro válido2. Enviar POST /theaters com token de admin3. Validar status code 2014. Validar mensagem de sucesso	{ "name": "Sala 1", "capacity": 150, "type": "IMAX" }	201 - Theater created successfully	Alta
CT- 061	API	Criação de teatro com nome em branco	Usuário admin autenticado	1. Definir payload com nome em branco2. Enviar POST /theaters 3. Validar status code 4004. Validar mensagem "Invalid input data"	{ "name": "", "capacity": 150, "type": "IMAX" }	400 - Invalid input data	Media
CT- 062	API	Criação de teatro	Usuário admin autenticado	Definir payload     com capacidade	{ "name": "Sala 2", "capacity": -50,	400 - Invalid	Media

		com capacidad e negativa		negativa2. Enviar POST /theaters 3. Validar status code 4004. Validar mensagem "Invalid capacity"	"type": "Standard" }		
CT- 063	API	Criação de teatro com dados duplicado s	Usuário admin autenticado	1. Definir payload com dados ja cadastrados 2. Enviar POST /theaters 3. Validar status code 409 4. Validar mensagem "Theater with that name already exists"	{ "name": "Sala 1", "capacity": 150, "type": "IMAX" }	409 -  1. Theater with that name already exists	Alta
CT- 064	API	Consulta de teatro por ID válido	Teatro existente	1. Enviar GET /theaters/{id} com ID válido2. Validar status code 2003. Validar que os dados do teatro sejam exibidos corretamente	N/A	200 - Theater details	Alta
CT- 065	API	Atualizaçã o de teatro com dados válidos	Teatro existente, usuário admin autenticado	1. Definir payload de atualização válida2. Enviar PUT /theaters/{id} com token de admin3. Validar status code 2004. Validar mensagem de sucesso	{ "name": "Sala 1 Atualizada", "capacity": 160, "type": "IMAX" }	200 - Theater updated successfully	Alta
CT- 066	API	Atualizaçã o de teatro sem autenticaç ão	Teatro existente	1. Definir payload de atualização válida2. Enviar PUT /theaters/{id} sem token3. Validar status code 4014. Validar mensagem "Not authorized"	{ "name": "Sala 1 Atualizada", "capacity": 160, "type": "IMAX" }	401 - Not authorized	Alta

CT- 067	API	Exclusão de teatro válido	Teatro existente, sem sessões vinculadas, admin autenticado	1. Enviar DELETE /theaters/{id} com token de admin2. Validar status code 2003. Validar mensagem "Theater deleted successfully"	N/A	200 - Theater deleted successfully	Alta
CT- 068	API	Exclusão de teatro com sessões vinculadas	Teatro com sessões, admin autenticado	1. Enviar DELETE /theaters/{id} com token de admin2. Validar status code 4003. Validar mensagem "Cannot delete theater with active sessions"	N/A	400 - Cannot delete theater with active sessions	Alta
CT- 069	API	Exclusão de teatro sem autenticaç ão	Teatro existente	1. Enviar DELETE /theaters/{id} sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authorized	Alta

## Users @

ID	Tip o	Caso de Teste	Pré-condição	Passos	Payload	Resultado Esperado	Prioridade
CT- 070	API	Listagem de usuários como admin com sucesso	Usuário admin autenticado	Enviar GET /users     com token de admin2.     Validar status code     2003. Validar que a     lista de usuários seja     exibida corretamente	N/A	200 - List of users	Alta
CT- 071	API	Obter usuário por ID com sucesso	Usuário admin autenticado, usuário existente	1. Enviar GET  /users/{id} com ID  válido e token de  admin2. Validar status  code 2003. Validar que  os dados do usuário  sejam exibidos  corretamente	N/A	200 - User details	Alta
CT- 072	API	Edição de usuário	Usuário existente,	Definir payload de     atualização válido2.	{ "name": "Nome Atualizado",	200 - User updated	Alta

		com dados válidos	admin autenticado	Enviar PUT /users/{id} com token de admin3. Validar status code 2004. Validar mensagem de sucesso	"role": "user" }	successfully	
CT- 073	API	Edição de usuário como usuário comum	Usuário existente, usuário comum autenticado	1. Definir payload de atualização válido2. Enviar PUT /users/{id} com token de usuário comum3. Validar status code 4034. Validar mensagem "Forbidden - Admin access required"	{ "name": "Novo Nome" }	403 - Forbidden - Admin access required	Alta
CT- 074	API	Exclusão de usuário com sucesso como admin	Usuário existente, admin autenticado, sem reservas ativas	1. Enviar DELETE /users/{id} com token de admin2. Validar status code 2003. Validar mensagem "User deleted successfully"	N/A	200 - User deleted successfully	Alta
CT- 075	API	Exclusão de usuário com reservas ativas	Usuário admin autenticado, usuário com reservas	1. Enviar DELETE /users/{id} com token de admin2. Validar status code 4003. Validar mensagem "Cannot delete user with active reservations"	N/A	400 - Cannot delete user with active reservations	Alta
CT- 076	API	Exclusão de usuário sem autenticaç ão	Usuário existente	1. Enviar DELETE /users/{id} sem token2. Validar status code 4013. Validar mensagem "Not authorized"	N/A	401 - Not authorized	Alta

## 9.2 Cenario de teste WEB $\mathscr{O}$

# Autenticação ${\mathscr O}$

ID Web	Caso de Teste Web	Passos Adaptados para o Navegador	Resultado Esperado
CTW-001	Cadastro de usuário válido via interface	Acessar tela de cadastro, preencher campos válidos,	Mensagem "Cadastro realizado com sucesso"

		clicar em "Cadastrar"	exibida
CTW-002	Cadastro com e-mail já existente via interface	Preencher tela de cadastro com e-mail existente, clicar em "Cadastrar"	Mensagem "Usuário já existe" exibida
CTW-003	Cadastro com e-mail inválido via interface	Informar e-mail inválido, clicar em "Cadastrar"	Mensagem "Formato de e- mail inválido" exibida
CTW-004	Cadastro com senha abaixo do mínimo via interface	Informar senha inválida, clicar em "Cadastrar"	Mensagem "Senha muito curta" exibida
CTW-005	Cadastro com campos vazios via interface	Deixar campos em branco, clicar em "Cadastrar"	Mensagem "Campos obrigatórios" exibida
CTW-006	Cadastro faltando campo e- mail via interface	Deixar o campo e-mail em branco, clicar em "Cadastrar"	Mensagem "Campos obrigatórios" exibida
CTW-007	Login com dados válidos via interface	Acessar tela de login, preencher e-mail e senha válidos, clicar em "Entrar"	Login realizado e redirecionamento
CTW-008	Login com e-mail não cadastrado via interface	Informar e-mail não existente, clicar em "Entrar"	Mensagem "Credenciais inválidas" exibida
CTW-009	Login com senha incorreta via interface	Informar senha incorreta, clicar em "Entrar"	Mensagem "Credenciais inválidas" exibida
CTW-010	Login com e-mail inválido (formato) via interface	Informar e-mail inválido, clicar em "Entrar"	Mensagem "Formato de e- mail inválido" exibida
CTW-011	Login com campos vazios via interface	Deixar campos vazios, clicar em "Entrar"	Mensagem "Campos obrigatórios" exibida
CTW-012	Visualizar dados do perfil do usuário via interface	Estar logado, acessar menu "Perfil"	Dados do perfil exibidos corretamente
CTW-013	Restrição: Tentar acessar perfil sem login	Acessar rota de perfil sem estar logado	Redirecionamento para login ou mensagem de acesso negado
CTW-014	Restrição: Tentar acessar perfil após token expirado	Simular logout ou expiração, acessar rota de perfil	Redirecionamento para login ou mensagem "Sessão expirada"
CTW-015	Atualizar dados do perfil via interface	Estar logado, acessar tela de perfil, editar dados, clicar em "Salvar"	Mensagem "Dados atualizados com sucesso" exibida
CTW-016	Restrição: Tentar atualizar perfil sem estar logado	Acessar tela de perfil sem login, tentar alterar dados	Redirecionamento ou mensagem de acesso negado
CTW-017	Atualização de perfil com campos vazios via interface	Estar logado, apagar campos obrigatórios, clicar em "Salvar"	Mensagem "Campos obrigatórios" exibida

## Filmes @

ID Web	Caso de Teste Web	Passos Adaptados para o Navegador	Resultado Esperado
CTW-018	Listagem de filmes via interface (público)	Acessar página inicial ou menu "Filmes"	Lista de filmes exibida corretamente
CTW-019	Visualizar detalhes de filme via interface	Acessar página de detalhes de um filme	Dados completos do filme exibidos corretamente

## Reservas $\mathscr Q$

ID Web	Caso de Teste Web	Passos Adaptados para o Navegador	Resultado Esperado
CTW-020	Listagem de reservas do usuário via interface	Estar logado, acessar menu "Minhas Reservas"	Lista de reservas do usuário exibida corretamente
CTW-021	Realizar reserva de sessão com assento disponível via interface	Estar logado, acessar sessão, selecionar assento disponível, escolher pagamento, confirmar	Mensagem "Reserva realizada com sucesso" exibida
CTW-022	Bloqueio de reserva em assento já ocupado via interface	Estar logado, tentar reservar assento já ocupado	Mensagem "Assento já reservado" exibida
CTW-023	Bloqueio de reserva em sessão inexistente via interface	Estar logado, tentar acessar sessão inexistente, tentar reservar	Mensagem "Sessão não encontrada" exibida
CTW-024	Visualizar detalhes de reserva via interface	Estar logado, acessar "Minhas Reservas", visualizar detalhes de reserva	Dados completos da reserva exibidos corretamente

## Sessoes @

ID Web	Caso de Teste Web	Passos Adaptados para o Navegador	Resultado Esperado
CTW-025	Listagem pública de sessões via interface	Acessar página inicial ou menu "Sessões"	Lista de sessões exibida corretamente
CTW-026	Visualizar detalhes de sessão via interface	Acessar página de detalhes de uma sessão	Dados completos da sessão exibidos corretamente

CTW-027	Bloqueio ao tentar visualizar sessão inexistente via interface	Tentar acessar detalhes de sessão inexistente	Mensagem "Sessão não encontrada" exibida ou redirecionamento
CTW-028	Resetar assentos da sessão via interface (admin)	Estar logado como admin, acessar sessão, clicar em "Resetar Assentos", confirmar	Mensagem "Assentos resetados com sucesso" exibida
CTW-029	Restrição: Resetar assentos como usuário comum	Estar logado como usuário comum, tentar resetar assentos	Mensagem "Acesso restrito" ou bloqueio visível

# 10. Matriz de Rastreabilidade 🔗

Requisito / HU	Casos de Teste Cobertos	Justificativa
HU-001 - Cadastro de Usuário	CT-001, CT-002, CT-003, CT-004, CT-005, CT-006	Validação completa do fluxo de cadastro, incluindo positivos e negativos de campos obrigatórios e regras de negócio
HU-002 - Login de Usuário	CT-007, CT-008, CT-009, CT-010, CT-011	Garante autenticação válida e cenários negativos como e-mail inválido ou senha incorreta
HU-003 - Perfil do Usuário	CT-012, CT-013, CT-014, CT-015, CT-016, CT-017, CT-018	Validação do acesso ao perfil e da atualização de dados, incluindo manipulação de token e dados inválidos
HU-004 - Listagem e Gerenciamento de Filmes	CT-019, CT-020, CT-021, CT-022, CT-023, CT-024, CT-025, CT-026, CT-027, CT-028, CT-029, CT-030	Validação completa dos fluxos de listagem, criação, consulta, atualização e exclusão de filmes, incluindo permissões
HU-005 - Reservas de Sessões	CT-031, CT-032, CT-033, CT-034, CT-035, CT-036, CT-037, CT-038, CT-039, CT-040, CT-041, CT-042, CT-043	Cobertura dos fluxos de reservas, desde a criação à exclusão, com validação de assentos ocupados e permissões
HU-006 - Gestão de Sessões	CT-044, CT-045, CT-046, CT-047, CT-048, CT-049, CT-050, CT-051, CT-052, CT-053, CT-054, CT-055, CT-056, CT-057, CT-058	Garante a integridade das sessões, incluindo horários válidos, duplicidade, dependências e permissões
HU-007 - Gestão de Teatros	CT-059, CT-060, CT-061, CT-062, CT-063, CT-064, CT-065, CT-066, CT-067, CT-068, CT-069	Validação dos fluxos de criação, consulta, atualização e exclusão de teatros, incluindo dependências de sessões
HU-008 - Gestão de Usuários (Admin)	CT-070, CT-071, CT-072, CT-073	Cobertura dos fluxos de listagem, consulta e edição de usuários restritos a administradores

# 11. Matriz de Risco 🔗

ID	Nível de Risco	Justificativa
CT-001 a CT-006	Alto	Cadastro mal validado compromete o acesso e segurança inicial do sistema
CT-007 a CT-011	Alto	Falhas de autenticação permitem ou bloqueiam indevidamente o acesso
CT-012 a CT-014	Alto	Acesso ao perfil sem validação adequada expõe dados pessoais
CT-015 a CT-018	Médio	Atualização de perfil incorreta afeta dados do usuário, mas com impacto menor que login
CT-019 a CT-025	Alto	Listagem e gerenciamento de filmes afetam diretamente o conteúdo principal do app
CT-026 a CT-030	Alto	Exclusão ou alteração indevida de filmes prejudica toda a experiência do usuário final
CT-031 a CT-043	Alto	Reservas impactam diretamente o core do sistema (venda de ingressos) e precisam ser robustas
CT-044 a CT-050	Alto	Gestão incorreta de sessões inviabiliza reservas e vendas de ingressos
CT-051 a CT-052	Alto	Consulta de sessão incorreta compromete informações exibidas ao público
CT-053 a CT-056	Alto	Atualização de sessões incorreta prejudica cronograma de exibição
CT-057 a CT-061	Alto	Exclusão indevida ou sem permissão de sessões impacta diretamente o negócio
CT-062 a CT-064	Médio	Reset de assentos afeta apenas parte do fluxo de reservas, sem impactar dados críticos
CT-059 a CT-065	Médio	Problemas em teatros afetam criação e estrutura de sessões, mas impacto é indireto
CT-066 a CT-069	Alto	Exclusão incorreta de teatros com sessões ativas pode corromper dados e causar inconsistências
CT-070 a CT-071	Alto	Listagem e consulta de usuários é sensível, pode expor dados se não for seguro

CT-072 a CT-073	Alto	Edição incorreta de usuários pode
		comprometer permissões e fluxo de
		autenticação

# 12. Priorização da execução dos cenários de teste

Prioridade	Casos de Teste	Justificativa
● Alta	CT-001, CT-004, CT-006, CT-007, CT-011, CT-012, CT-013, CT-014, CT-015, CT-019, CT-020, CT-023, CT-025, CT-026, CT-027, CT-031, CT-032, CT-033, CT-034, CT-035, CT-037, CT-038, CT-044, CT-045, CT-046, CT-047, CT-048, CT-049, CT-051, CT-052, CT-053, CT-054, CT-055, CT-056, CT-059, CT-060, CT-063, CT-064, CT-065, CT-065, CT-066, CT-070, CT-071, CT-072, CT-073	Casos que impactam diretamente as funcionalidades essenciais do sistema (autenticação, cadastro, produtos, sessões, reservas e permissões administrativas). Erros aqui comprometem o uso ou a segurança do sistema.
<mark>● M</mark> édiα	CT-002, CT-003, CT-005, CT-008, CT-009, CT-010, CT-016, CT-017, CT-018, CT-021, CT-022, CT-024, CT-028, CT-029, CT-030, CT-036, CT-039, CT-040, CT-041, CT-042, CT-050, CT-057, CT-058, CT-061, CT-062, CT-067, CT-068, CT-069	Casos relacionados a validações de regras de negócio, consistência de dado e fluxos secundários. Falhas não bloqueiam o uso do sistema, mas geram inconsistências ou prejuízo parcial.

## 13. Cobertura de Testes 🔗

## Manual @

Cobertura (%) = (Caso de testes mapeados : / Casos de teste planejado:)  $\times$  100

Total de Casos de planejado:76

Casos de Teste mapeado: 76

Cobertura de Testes: 100%

## Automatizado API 🔗

Cobertura (%) = (Caso de testes mapeados : / Casos de teste planejado:)  $\times$  100

Total de Casos de planejado: 76

Casos de Teste mapeado: 30

✓ Cobertura de Testes: 43,36%

## Automatizado WEB 🖉

Cobertura (%) = (Caso de testes mapeados : / Casos de teste planejado:) × 100

Total de Casos de planejado: 29

Casos de Teste mapeado: 11

Cobertura de Testes: 37,93%

# 14. Testes Candidatos à Automação 🔗

## 14.1 Automação Back 🔗

- V Fluxos críticos de negócio
- ✓ Fluxo feliz ponta a ponta via API
- Casos positivos e estáveis
- ✓ Validações de segurança e permissões principais

Etapa	Casos de Teste Automatizados	Justificativa da Automação
Fluxo Feliz Ponta a Ponta (API)	CT-001, CT-007, CT-020, CT-060, CT-045, CT-033, CT-037, CT-042, CT-052, CT-028, CT-067	Simular o fluxo completo do sistema, garantindo integração entre cadastro, login, filmes, teatro, sessões, reservas e exclusões
Autenticação e Cadastro (Fluxo Feliz)	CT-001, CT-007, Criar adm, Login Adm	Acesso inicial ao sistema deve ser automatizado para validação rápida
Filmes (CRUD Positivo)	CT-019, CT-020, CT-025, CT-026, CT-028	Garantir funcionamento do CRUD de filmes com dados válidos e fluxo esperado
Teatros (CRUD Positivo)	CT-059, CT-060, CT-064, CT-065, CT-067	Validação de cadastro e edição básica de teatros
Sessões (Fluxo Positivo)	CT-044, CT-045, CT-049, CT-051, CT-052, CT-056	Fluxo essencial de sessões: listagem, criação, consulta, atualização e exclusão
Reservas (Fluxo Positivo)	CT-031, CT-033, CT-037, CT-038, CT-042	Reserva, consulta e exclusão de reservas com dados válidos
Usuários (Administração - Positivo)	CT-070, CT-071, CT-072,CT-074	Gestão básica de usuários feita por administradores

## 14.2 Automação Web 🖉

✓ A automação Web inicialmente cobre apenas fluxos positivos e essenciais

✓ Casos de mensagens de erro, campos inválidos e outros negativos permanecem mapeados, porém serão executados manualmente no Frontend

_		
Etapa	Casos de Teste Automatizados	Justificativa da Automação

Autenticação e Cadastro	CTW-001, CTW-007, CTW-012, CTW-015	Fluxo essencial de acesso ao sistema e gerenciamento básico de perfil. Automatizar garante validação rápida e recorrente do acesso e dos dados do usuário.
Filmes (Listagem e Detalhes)	CTW-018, CTW-019	Validação de funcionalidades públicas essenciais para a experiência do usuário final, garantindo que a listagem e detalhes de filmes sejam exibidos corretamente.
Reservas (Fluxo Positivo)	CTW-020, CTW-021, CTW-024	Fluxo completo de sessões via interface: listagem, detalhes e reset de assentos para administradores. Automatização garante integridade no processo.
Sessões (Fluxo Positivo)	CTW-025, CTW-026	Reserva, consulta e visualização de reservas realizadas pelo usuário

# Observações Finais: @

- Os testes manuais serão executados via QAlity para evidência e acompanhamento
- A automação utilizará o Robot Framework, com keywords reutilizáveis e estrutura modular
- Casos exclusivamente negativos, altamente variáveis ou exploratórios permanecerão como testes manuais, otimizando o esforço de automação
- O fluxo completo ponta a ponta da API já está mapeado como prioridade para automação