Plano de Testes – Cinema app

Informações Gerais

Apresentação

Este documento foi elaborado com o intuito de organizar e documentar o planejamento da rodada de testes referente ao **Cinema app**
O plano serve como base para a execução controlada dos testes manuais e/ou automatizados, garantindo rastreabilidade entre requisitos e testes.

> **Responsável pelo documento:** @Maria Eduarda Martins Rodrigues
> **Data de criação:** 23 de jun. de 2025
> **Última atualização:** 24 de jun. de 2025
> **Sprint:** Sprint 06

Objetivo

Validar o comportamento da aplicação **Cinema App**, assegurando o correto funcionamento dos fluxos principais, como autenticação, cadastro de usuários, gerenciamento de filmes, sessões, reservas e perfil de usuário, conforme os critérios de aceitação definidos nas **User Stories** e nos detalhes da **API (Swagger UI)**.
Esta rodada de testes tem como foco garantir a estabilidade, a aderência às regras de negócio descritas, e a identificação de possíveis inconsistências funcionais nos endpoints disponíveis.

Escopo

Funcionalidades incluídas:

- **Cadastro e autenticação de usuários:** Registro de novos usuários, login via token JWT, logout e atualização de perfil.
- **Listagem e detalhamento de filmes:** Visualização de lista de filmes, detalhamento por ID e filtros por título e gênero.
- **Gerenciamento de sessões:** Listagem de sessões, visualização de detalhes de uma sessão e controle de assentos.
- **Gerenciamento de reservas:** Criação de reservas, visualização de histórico individual e listagem/administração de reservas (admin).
- **Navegação geral:** Acesso a informações de perfil, reservas e filmes através de um menu de navegação funcional.
- **Gerenciamento de salas (teatros):** Visualização, criação, atualização e exclusão de teatros (somente admin).
- **Gerenciamento de filmes:** Visualização, criação, atualização e exclusão de filmes (somente admin).

Fora de Escopo:

- **Testes de performance e carga:** Não serão realizados testes de estresse, escalabilidade ou latência.
- **Segurança avançada:** Testes de vulnerabilidades (SQL Injection, XSS, etc) estão fora desta rodada.
- **Interface gráfica (UI):** Esta fase se concentra nos testes da API. Validação de layout e UX será tratada em uma etapa separada.

Ambiente de Testes

Item	Detalhes
Projeto	Cinema App
Versão	1.0
Tipo de Teste	Teste Baseado em Requisitos, Particionamento de Equivalência, Análise de Valor Limite, Tabela de Decisão, Transição de Estado, Teste Baseado em Fluxo de Uso, Teste Negativo, Teste Exploratório e Teste Automatizado
Período de Execução	25 de jun. de 2025 - 3 de jul. de 2025
Responsável(is)	@Maria Eduarda Martins Rodrigues
Ambiente	Testes
Acesso / URL	API local em ambiente de desenvolvimento conforme Swagger

Análise Inicial

Tipo de Análise	Detalhes
Requisitos	User Stories detalhadas e documentação Swagger. As funcionalidades estão descritas com critérios de aceitação claros, exceto para algumas validações de regras de negócios em endpoints específicos, que serão verificadas durante os testes exploratórios.

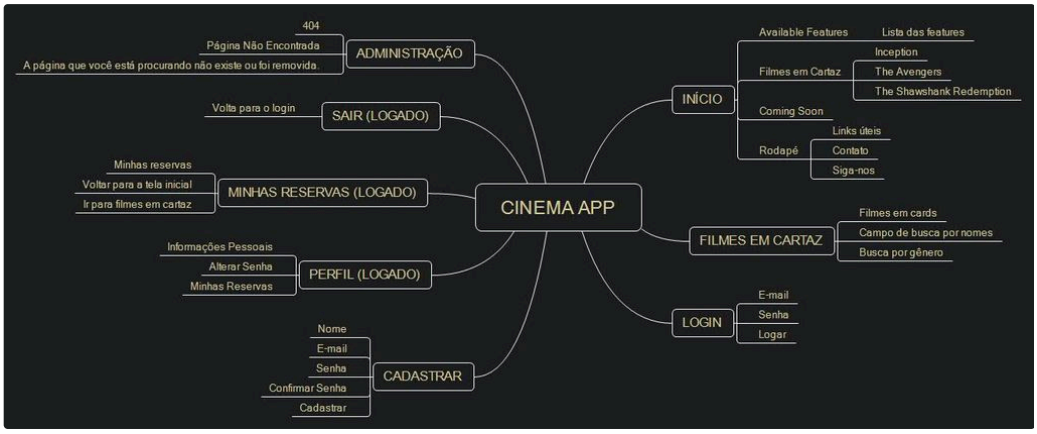
Riscos Identificados	Instabilidade do ambiente de desenvolvimento, falhas intermitentes nos endpoints, falta de mensagens claras de erro, tratamento inconsistente de permissões entre usuários comuns e administradores, e ausência de documentação de limites (ex: tamanho máximo de campos).
Histórico de Falhas	Não se apllica (primeira execução).
Apontamentos	A API do Cinema App foi analisada a partir das User Stories e da documentação Swagger. Os endpoints seguem uma arquitetura RESTful clara, o que facilita a definição de cenários de teste com rastreabilidade direta. Futuramente recomenda-se ampliar a cobertura com testes de segurança e performance.

Técnicas de Teste Aplicadas

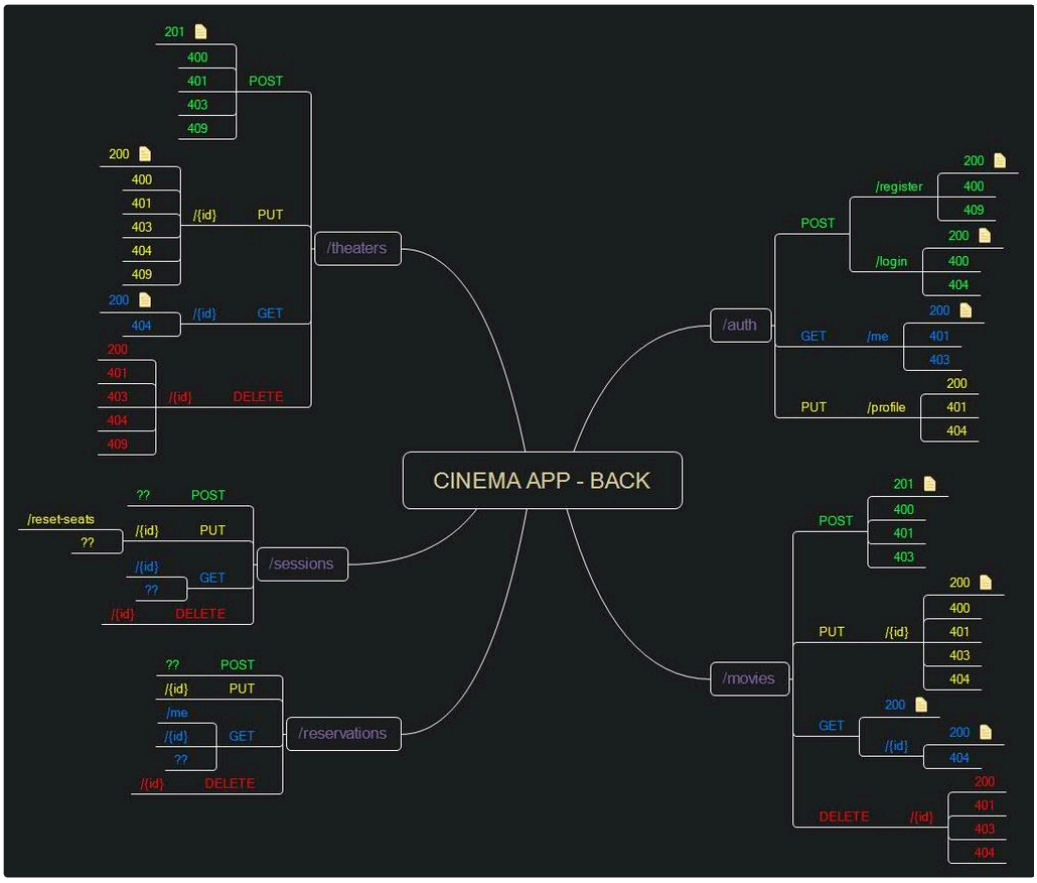
- **Teste Baseado em Requisitos (Requirements-Based Testing):**
  Os cenários serão definidos com base direta nas **User Stories** e **Swagger UI**, garantindo alinhamento com os requisitos de negócio.
- **Particionamento de Equivalência (Equivalence Partitioning):**
  Divisão dos dados de entrada em classes de equivalência válidas e inválidas (ex.: e-mails válidos e inválidos, IDs existentes e inexistentes).
- **Análise do Valor Limite (Boundary Value Analysis):**
  Teste de valores nos limites superior e inferior dos campos numéricos e de texto (ex.: senha com número mínimo e máximo de caracteres, limite de paginação na listagem de filmes).
- **Tabela de Decisão (Decision Table Testing):**
  Verificação de respostas esperadas com base em combinações lógicas de inputs (ex.: combinações de permissões de usuários ao acessar endpoints restritos).
- **Transição de Estado (State Transition Testing):**
  Validação de mudanças de estado do sistema (ex.: status de reservas, estado de sessão antes e depois de atualizar assentos).
- **Teste Baseado em Fluxo de Uso (Use Case Testing):**
  Execução de cenários end-to-end, como: *Registro → Login → Visualização de filmes → Criação de reserva → Verificação de reservas*.
- **Teste Negativo (Negative Testing):**
  Testes com dados inválidos, tokens expirados, usuários não autorizados, tentativas de deletar recursos inexistentes, etc.
- **Teste Exploratório (Exploratory Testing):**
  Execução livre para identificação de falhas inesperadas e comportamentos não documentados.
- **Teste Automatizado (Automated Testing):**
  Automação dos testes mais críticos e repetitivos (ex.: validações de autenticação, CRUD de filmes, validação de reservas).

Mapa Mental da Aplicação

Front:



Back:



Matriz de Rastreabilidade: Requisitos x Casos de Teste

ID	Requisito (Acceptance Criteria)	Casos de Teste (ID)

US-AUTH-001-R01	Usuário deve possuir os campos nome, email e password.	CT001
US-AUTH-001-R02	Sistema deve validar o formato do e-mail e senha durante o registro.	CT003, CT004
US-AUTH-001-R03	Sistema deve impedir o cadastro de e-mails duplicados.	CT002
US-AUTH-001-R04	Após registro bem-sucedido, usuário deve ser redirecionado para a página de login e já autenticado.	CT001
US-AUTH-002-R01	Usuário deve conseguir realizar login com e-mail e senha válidos.	CT005
US-AUTH-002-R02	Após login bem-sucedido, sistema deve gerar um token JWT, manter a sessão e redirecionar para a página inicial.	CT005
US-AUTH-002-R03	Sistema deve retornar mensagens de erro claras para credenciais inválidas.	CT006, CT007
US-AUTH-003-R01	Usuário autenticado deve conseguir realizar logout através do menu de navegação.	CT012
US-AUTH-003-R02	Após logout, rotas protegidas devem ficar inacessíveis.	CT012, CT013
US-AUTH-003-R03	Token JWT deve ser removido do localStorage.	CT012
US-AUTH-004-R01	Usuário autenticado deve poder visualizar no perfil os campos: nome, e-mail e função da conta.	CT008
US-AUTH-004-R02	Usuário deve poder editar o nome.	CT009
US-AUTH-004-R03	Usuário deve poder editar a senha e o sistema valida se a senha atual está correta.	CT010, CT011
US-AUTH-004-R04	Sistema deve validar alterações feitas e indicar visualmente os campos modificados.	CT009, CT010
US-AUTH-004-R05	Mensagem de sucesso deve ser exibida após atualização de perfil.	CT009, CT010
US-HOME-001-R01	Página inicial deve exibir um banner com informações sobre o cinema.	CT014
US-HOME-001-R02	Página inicial deve listar os filmes em cartaz.	CT014
US-HOME-001-R03	Layout da página deve ser responsivo, adaptando-se a diferentes tamanhos de tela, ter links rápidos para as principais áreas e cabeçalho.	CT013
US-HOME-001-R04	Usuários autenticados devem poder ver opções personalizadas no menu.	CT013
US-MOVIE-001-R01	Usuário deve conseguir visualizar uma lista de filmes em layout grid, com pôster e detalhes	CT015

	básicos.	
US-MOVIE-001-R02	Sistema deve permitir filtros por título e gênero.	CT015
US-MOVIE-002-R01	Usuário deve conseguir visualizar detalhes do filme com um clique.	CT016
US-MOVIE-002-R02	Página de detalhes deve exibir horários de sessões disponíveis para o filme e link para reserva.	CT016
US-SESSION-001-R01	Usuário deve conseguir visualizar os horários de sessões disponíveis, com data, hora, teatro e disponibilidade.	CT017
US-RESERVE-001-R01	Usuário logado deve poder visualizar o layout de assentos com códigos de cor por disponibilidade.	CT018
US-RESERVE-001-R02	Sistema deve impedir seleção de assentos já ocupados.	CT019
US-RESERVE-001-R03	Usuário pode selecionar múltiplos assentos disponíveis.	CT018
US-RESERVE-001-R04	Sistema deve calcular e exibir o subtotal da reserva conforme os assentos selecionados.	CT018
US-RESERVE-002-R01	Após selecionar os assentos, usuário deve ser redirecionado para a página de checkout.	CT018
US-RESERVE-002-R02	Página de checkout deve exibir um resumo da compra, com valor total, assentos e método de pagamento.	CT018
US-RESERVE-002-R03	Sistema deve processar o pagamento (simulado) e confirmar a reserva.	CT018
US-RESERVE-002-R04	Assentos selecionados devem ser marcados como ocupados após confirmação da reserva.	CT018
US-RESERVE-003-R01	Usuário deve acessar seu histórico de reservas via menu através do link "Minhas Reservas".	CT020
US-RESERVE-003-R02	Reservas devem ser exibidas em formato de card com informações claras.	CT020
US-RESERVE-003-R03	Usuário pode visualizar o pôster do filme associado à reserva.	CT020
US-RESERVE-003-R04	Sistema exibe indicadores visuais de status da reserva (confirmada, pendente, cancelada).	CT020
US-RESERVE-003-R05	Usuário pode acessar página dedicada de reservas separada das informações de perfil.	CT020
US-NAV-001-R01	Cabeçalho de navegação deve estar presente em todas as páginas, com links para áreas principais.	CT013
US-NAV-001-R02	Menu deve ser responsivo, com versão mobile adequada.	CT013

US-NAV-001-R03	Usuário logado deve ter acesso às opções personalizadas: “Minhas Reservas” e “Perfil”.	CT013
US-NAV-001-R04	Sistema deve fornecer breadcrumbs ou elementos visuais que indiquem o caminho atual.	CT013
US-NAV-001-R05	Links devem permitir fácil retorno à página anterior, quando apropriado.	CT013
US-MOVIE-ADMIN-001	Admin deve conseguir criar um novo filme com campos obrigatórios.	CT021
US-MOVIE-ADMIN-002	Admin deve conseguir atualizar um filme.	CT021
US-MOVIE-ADMIN-003	Admin deve conseguir excluir um filme.	CT021
US-MOVIE-ADMIN-004	Sistema deve validar payload inválido ao criar ou editar filme.	CT022
US-MOVIE-ADMIN-005	Sistema deve impedir que usuários comuns criem, editem ou excluam filmes.	CT023
US-THEATER-ADMIN-001	Admin deve conseguir criar um novo teatro.	CT024
US-THEATER-ADMIN-002	Admin deve conseguir atualizar um teatro.	CT024
US-THEATER-ADMIN-003	Admin deve conseguir excluir um teatro sem sessões.	CT024
US-THEATER-ADMIN-004	Sistema deve impedir exclusão de teatro com sessões ativas.	CT027
US-THEATER-ADMIN-005	Sistema deve validar payload inválido ao criar ou editar um teatro.	CT025
US-THEATER-ADMIN-006	Sistema deve impedir que usuários comuns criem, editem ou excluam teatros.	CT026
US-SESSION-ADMIN-001	Admin deve conseguir criar uma nova sessão de filme.	CT028
US-SESSION-ADMIN-002	Admin deve conseguir atualizar os detalhes de uma sessão existente.	CT029
US-SESSION-ADMIN-003	Admin deve conseguir excluir uma sessão.	CT030
US-SESSION-ADMIN-004	Admin deve conseguir resetar os assentos de uma sessão para o status disponível.	CT031
US-RESERVATION-ADMIN-001	Admin deve conseguir visualizar a lista completa de reservas de todos os usuários.	CT032
US-RESERVATION-ADMIN-002	Admin deve conseguir atualizar o status de uma reserva.	CT033
US-RESERVATION-ADMIN-003	Admin deve conseguir excluir uma reserva existente.	CT034

US-USER-ADMIN-001	Admin deve conseguir listar todos os usuários via GET /users e visualizar os dados de um usuário específico via GET /users/{id}.	CT035
US-USER-ADMIN-002	Admin deve conseguir editar os dados de um usuário via PUT /users/{id}.	CT036
US-USER-ADMIN-003A	Admin deve conseguir excluir um usuário via DELETE /users/{id} quando não houver reservas ativas.	CT037
US-USER-ADMIN-003B	Sistema deve impedir a exclusão de usuário com reservas ativas, retornando 409 Conflict.	CT038
US-USER-ADMIN-004	Sistema deve impedir que usuários comuns acessem os endpoints de usuários, retornando 403 Forbidden.	CT039

Cobertura de testes

Path Coverage (input)

✅ (14 / 14) × 100 = 100%

Operator Coverage (input)

✅ (26 / 26) × 100 = 100%

Coverage by Requirements (Cobertura por Requisitos)

✅ (65 / 65) × 100 = 100%

Cenários de Teste Planejados com Priorização

ID	Prioridade	Caso de Teste	Passos	Resultado Esperado
CT001	Alta	Registro de usuário com sucesso	Pré-condições: Nenhuma Execução: API 1. Enviar POST /auth/register com nome, e-mail e senha válidos	201 Created com token e redirecionamento ao login (no teste web)
CT002	Alta	Registro com e-mail duplicado	Pré-condições: Usuário já existente	409 Conflict com mensagem de e-mail duplicado

			Execução: API 1. Enviar POST /auth/register com o mesmo e-mail	
CT003	Alta	Registro com e-mail inválido	Pré-condições: Nenhuma Execução: API 1. Enviar POST /auth/register com e-mail mal formatado	400 Bad Request com mensagem de validação
CT004	Alta	Registro com senha inválida	Pré-condições: Nenhuma Execução: API 1. Enviar POST /auth/register com senha abaixo do mínimo permitido	400 Bad Request com mensagem de política de senha
CT005	Alta	Login com sucesso	Pré-condições: Usuário existente Execução: API 1. Enviar POST /auth/login com credenciais válidas	200 OK com token JWT
CT006	Alta	Login com senha incorreta	Pré-condições: Usuário existente Execução: API 1. Enviar POST /auth/login com senha incorreta	400 Bad Request com mensagem de credenciais inválidas
CT007	Média	Login com e-mail não cadastrado	Pré-condições: Nenhuma Execução: API 1. Enviar POST /auth/login com e-mail inexistente	404 Not Found

CT008	Baixa	Visualizar perfil autenticado	Pré-condições: Usuário autenticado Execução: API 1. Enviar GET /auth/me	200 OK com nome, e-mail e função
CT009	Baixa	Editar nome do perfil	Pré-condições: Usuário autenticado Execução: API 1. Enviar PUT /auth/profile alterando o nome	200 OK com confirmação de atualização
CT010	Baixa	Editar senha com sucesso	Pré-condições: Usuário autenticado Execução: API 1. Enviar PUT /auth/profile alterando a senha com senha atual correta	200 OK com confirmação de atualização
CT011	Baixa	Editar senha com senha atual incorreta	Pré-condições: Usuário autenticado Execução: API 1. Enviar PUT /auth/profile com senha atual incorreta	401 Unauthorized
CT012	Baixa	Logout e proteção de sessão	Pré-condições: Usuário autenticado Execução: Front-End 1. Realizar logout pelo menu 2. Tentar acessar rotas protegidas	Sessão encerrada, token removido, rotas protegidas inacessíveis
CT013	Média	Navegação personalizada e	Pré-condições: Usuário	Navegação correta e layout

		responsiva	autenticado Execução: Front-End 1. Navegar por Home, Filmes, Perfil e Minhas Reservas 2. Redimensionar tela para mobile e desktop	adaptável
CT014	Baixa	Exibir banner e lista de filmes na home	Pré-condições: Filmes cadastrados Execução: Front-End 1. Acessar página inicial	Banner exibido, filmes listados com pôsteres
CT015	Média	Listagem de filmes com filtros e paginação	Pré-condições: Múltiplos filmes cadastrados Execução: Front-End 1. Ir para a página de buscar filmes 2. Usar os filtros por título e gênero	Lista de filmes filtrada em layout grid, com pôster e detalhes básicos.
CT016	Média	Detalhes de filme com sessões	Pré-condições: Filme com sessões ativas Execução: Front-End 1. Acessar detalhes de um filme	Exibição de sinopse, elenco, diretor, duração, data de lançamento, pôster e horários de sessões
CT017	Média	Listagem de sessões por filme	Pré-condições: Nenhuma Execução: Front-End 1. Acessar detalhes de um filme	Horários de sessões disponíveis, com data, hora, teatro e disponibilidade

			2. Visualisar as sessões do filme	
CT018	Alta	Fluxo de reserva com sucesso	Pré-condições: Sessão com assentos disponíveis Execução: Front-End 1. Selecionar assentos 2. Ir para checkout 3. Confirmar reserva	Subtotal calculado, reserva confirmada, assentos ocupados
CT019	Média	Tentar reservar assentos ocupados	Pré-condições: Assentos já reservados Execução: Front-End 1. Tentar selecionar assentos ocupados	400 Bad Request com mensagem de assento indisponível
CT020	Alta	Histórico de reservas	Pré-condições: Reservas existentes Execução: Front-End 1. Acessar Minhas Reservas	Cards com detalhes de filme, data, horário, cinema, assentos, status e método de pagamento
CT021	Alta	CRUD de filme por Admin (sucesso)	Pré-condições: Admin autenticado Execução: API 1. Criar filme (POST /movies) 2. Editar filme (PUT /movies/{id}) 3. Excluir filme (DELETE /movies/{id})	201 Created, 200 OK e 200 OK

CT022	Média	Validação de payload inválido ao criar filme	Pré-condições: Admin autenticado Execução: API 1. Criar filme sem campo obrigatório	400 Bad Request com mensagem de validação
CT023	Baixa	Bloqueio de CRUD de filme por usuário comum	Pré-condições: Usuário comum autenticado Execução: API 1. Tentar criar filme 2. Tentar editar filme 3. Tentar excluir filme	403 Forbidden em todas as ações
CT024	Alta	CRUD de teatro por Admin (sucesso)	Pré-condições: Admin autenticado Execução: API 1. Criar teatro (POST /theaters) 2. Editar teatro (PUT /theaters/{id}) 3. Excluir teatro sem sessões (DELETE /theaters/{id})	201 Created, 200 OK e 200 OK
CT025	Média	Validação de payload inválido ao criar ou editar teatro	Pré-condições: Admin autenticado Execução: API 1. Criar teatro com payload incompleto	400 Bad Request
CT026	Baixa	Bloqueio de CRUD de teatro por usuário comum	Pré-condições: Usuário comum autenticado Execução: API 1. Tentar criar teatro	403 Forbidden em todas

			2. Tentar editar teatro 3. Tentar excluir teatro	
CT027	Média	Bloqueio ao tentar excluir teatro com sessões ativas	Pré-condições: Teatro com sessões Execução: API 1. Admin tenta excluir teatro (DELETE /theaters/{id})	409 Conflict com mensagem de dependência
CT028	Média	Criar nova sessão de filme por Admin	Pré-condições: Admin autenticado Execução: API 1. Enviar POST /sessions com movieId, theaterId, data e horário válidos	201 Created com detalhes da nova sessão
CT029	Baixa	Atualizar sessão existente por Admin	Pré-condições: Sessão existente Execução: API 1. Enviar PUT /sessions/{id} com novos dados válidos (ex: alterar horário)	200 OK com confirmação da atualização
CT030	Média	Excluir sessão por Admin	Pré-condições: Sessão existente e sem reservas vinculadas Execução: API 1. Enviar DELETE /sessions/{id}	200 OK com confirmação da exclusão
CT031	Baixa	Reset de assentos de uma sessão (Admin)	Pré-condições: Sessão com assentos ocupados Execução: API 1. Enviar PUT /sessions/{id}/	200 OK com todos os assentos resetados para disponível

			reset-seats	
CT032	Baixa	Listar todas as reservas (Admin)	Pré-condições: Reservas existentes Execução: API 1. Enviar GET /reservations	200 OK com lista completa de reservas, incluindo status, usuário e detalhes de assentos
CT033	Média	Atualizar status de uma reserva (Admin)	Pré-condições: Reserva existente Execução: API 1. Enviar PUT /reservations/{id} alterando o status da reserva (ex: de "pendente" para "confirmada")	200 OK com confirmação da alteração
CT034	Baixa	Excluir uma reserva (Admin)	Pré-condições: Reserva existente Execução: API 1. Enviar DELETE /reservations/{id}	200 OK com confirmação da exclusão da reserva
CT035	Baixa	Listar e visualizar dados de usuários como admin	Pré-condições: Admin autenticadoExecução: API1. Enviar GET /users2. Enviar GET /users/{id} com ID válido	200 OK com lista de usuários completa200 OK com dados detalhados do usuário
CT036	Alta	Editar dados de um usuário para promovê-lo à admin	Pré-condições: Admin autenticado e usuário comum existenteExecução: API1. Enviar PUT /users/{id} com o campo `"role": "admin"`	200 OK com confirmação da atualização e usuário promovido à função de admin
CT037	Média	Excluir usuário **sem** reservas como admin	Pré-condições: Admin autenticado e usuário sem reservas	200 OK com confirmação da exclusão

			existenteExecução: API1. Enviar DELETE /users/{id}	
CT038	Média	Tentar excluir usuário **com** reservas como admin	Pré-condições: Admin autenticado e usuário com reservas existenteExecução: API1. Enviar DELETE /users/{id}	409 Conflict com mensagem de dependência
CT039	Alta	Bloqueio de acesso aos endpoints de usuários por usuário comum	Pré-condições: Usuário comum autenticadoExecução: API1. Enviar GET /users2. Enviar GET /users/{id}3. Enviar PUT /users/{id}4. Enviar DELETE /users/{id}	403 Forbidden em todas as operações

---

Matriz de Risco

Risco	Impacto	Probabilidade	Ação de Mitigação
Ambiente instável durante a execução	Média	Média	Realizar check diário do ambiente e planejar reexecuções conforme necessidade.
Falta de requisitos detalhados para algumas validações	Média	Baixa	Aceitar passivamente; durante testes exploratórios, documentar as observações e buscar esclarecimentos para possíveis atualizações.
Retornos genéricos em erros (ausência de mensagens claras e status HTTP apropriados)	Alta	Média	Documentar todos os casos de erro genérico; reportar para equipe de desenvolvimento para melhoria do feedback da API.

Ambiguidade no tratamento de permissões entre usuários comuns e administradores	Médio	Baixa	Validar rigorosamente todos os fluxos com diferentes perfis; criar casos de teste específicos para verificar permissões e acessos indevidos.
Ausência de documentação sobre limites de campos (ex.: tamanho máximo)	Alta	Alta	Realizar testes exploratórios de boundary; documentar comportamentos inesperados e sugerir inclusão desses limites na documentação.

Cobertura de testes Automatizados

Path Coverage (input)

✅ (11 / 14) × 100 = 78%

Operator Coverage (input)

✅ (16 / 26) × 100 = 61%

Coverage by Requirements (Cobertura por Requisitos)

✅ (39 / 65) × 100 = 59%

Testes Candidatos à Automação

ID	Justificativa para Automação
CT001	Testa o fluxo principal de criação de usuário com dados válidos (API).
CT002	Validação de e-mail duplicado — regra de negócio fundamental para integridade (API).
CT003	Validação de formato de e-mail — cenário clássico de validação de dados (API).
CT004	Política de senha mínima — validação importante para segurança (API).

CT005	Login com sucesso — fluxo essencial de autenticação (API).
CT006	Login com senha incorreta — validação negativa de segurança (API).
CT007	Login com e-mail inexistente — importante para segurança e mensagens de erro (API).
CT015	Listagem de filmes com filtros e paginação — fluxo de busca essencial (Front-End - Web).
CT016	Detalhes de filme com sessões — garante exibição correta de dados de filmes e sessões (Front-End - Web).
CT018	Fluxo de reserva com sucesso — fluxo de negócio crítico (Front-End - Web).
CT019	Tentar reservar assentos ocupados — validação de bloqueio de reservas inválidas (Front-End - Web).
CT020	Histórico de reservas — essencial para acompanhamento de usuário (Front-End - Web).
CT021	CRUD de filme por Admin (sucesso) — operações administrativas core (API).
CT022	Validação de payload inválido ao criar filme — importante para validação de entrada de dados (API).
CT023	CRUD de filme por usuário comum - proteção de rota crítica.
CT024	CRUD de teatro por Admin (sucesso) — equivalente ao fluxo de filmes (API).
CT025	Validação de payload inválido ao criar ou editar teatro — similar ao de filmes (API).
CT026	CRUD de teatro por usuário comum - proteção de rota crítica.

CT027	Bloqueio ao tentar excluir teatro com sessões ativas — proteção de integridade de dados (API).
CT028	Criar nova sessão de filme por Admin — fluxo administrativo (API).
CT033	Atualizar status de uma reserva (Admin) — fluxo de gestão de reservas (API).
CT036	Testa a atualização de um usuário, incluindo mudança de permissões — fluxo crítico para controle de acesso.
CT037	Valida exclusão de usuários sem reservas — ação administrativa fundamental, de fácil automação via API.
CT039	Garante o bloqueio de acesso por usuários não autorizados — importante para segurança e controle de permissões.

Referências e Anexos

- Quadro do Jira (Link)
- Relatório de testes (Link)