

Cinema App

Autor: Mário Queiroz

1. Apresentação

Este documento detalha o planejamento estratégico para os testes da aplicação **Cinema App**, cobrindo tanto o **Frontend** quanto o **Backend**. O objetivo é guiar a equipe de Qualidade na validação completa das funcionalidades, da interface do usuário à lógica de negócio no servidor, com base nas Histórias de Usuário.

Este plano servirá como a principal fonte de referência para a criação de cenários, execução de testes e reporte de resultados.

- **Aplicação-alvo:** Cinema App
- **Ambientes de Teste:**
 - **Frontend:** Navegador Chrome/Chromium.
 - **Backend:** Ambiente de desenvolvimento local com Node.js e MongoDB.
- **Ferramentas:**
 - **Frontend:** Ferramentas de desenvolvedor do navegador, Robot (para automação E2E).
 - **Backend:** Postman, Insomnia (para testes de API).
 - **Gestão:** Jira e Confluence.

2. Objetivo

O objetivo principal é **validar a qualidade, funcionalidade e usabilidade da aplicação Cinema App de ponta a ponta**, garantindo que a experiência do usuário seja coesa e que a integração entre o frontend e o backend funcione perfeitamente, conforme os critérios de aceitação das Histórias de Usuário.

Objetivos Secundários:

- Garantir uma experiência de usuário intuitiva e visualmente consistente em diferentes navegadores e tamanhos de tela.
- Validar a segurança e a integridade dos dados desde a entrada no frontend até o armazenamento no backend.
- Identificar e documentar defeitos funcionais, de usabilidade e de integração.
- Criar uma base de testes de regressão (manuais e automatizados) para garantir a

estabilidade da aplicação em futuras atualizações.

3. Escopo

3.1. Funcionalidades em Escopo:

- **Testes de Backend (API):**
 - Validação de todos os endpoints (`/auth`, `/users`, `/movies`, `/theaters`, `/sessions`, `/reservations`).
 - Testes de contrato (request/response).
 - Validação de regras de negócio no servidor.
 - Testes de segurança de rotas (autenticação e autorização por roles).
- **Testes de Frontend (UI/UX):**
 - Validação de todos os fluxos de usuário descritos nas histórias.
 - Testes de componentes visuais e interativos.
 - Testes de responsividade (desktop, tablet, mobile).
 - Testes de usabilidade e navegação.
- **Testes de Integração (E2E - Ponta a Ponta):**
 - Fluxos completos que simulam a jornada do usuário, como: Registro -> Login -> Seleção de Filme -> Escolha de Assentos -> Checkout -> Verificação em "Minhas Reservas".

3.2. Funcionalidades Fora do Escopo:

- Testes de performance, carga e estresse.
- Integração real com gateways de pagamento (o pagamento é simulado).
- Testes de compatibilidade com navegadores legados (ex: Internet Explorer).

4. Análise de Testes (Estratégia)

A estratégia será dividida em três camadas:

1. **Testes de API (Backend):** Foco na lógica de negócio, integridade dos dados e segurança. Serão os primeiros a serem executados para garantir que a "base" da aplicação está sólida.
2. **Testes de UI (Frontend):** Foco nos componentes visuais, interatividade e experiência do usuário. Valida se a "vitrine" da loja é funcional e agradável.
3. **Testes de Ponta a Ponta (E2E):** Foco na integração entre as camadas. Valida se um cliente consegue entrar na loja, escolher um produto, pagar e sair com ele, simulando o fluxo real.

5. Técnicas de Teste Aplicadas

- **Particionamento de Equivalência e Análise de Valor Limite:** Usadas tanto no

backend (para testar a API) quanto no frontend (para testar formulários).

- **Teste Baseado em Estado:** Para validar as transições de estado da aplicação (ex: Visitante -> Logado -> Admin).
- **Testes Exploratórios:** Sessões livres onde o testador explora a aplicação sem um roteiro, buscando encontrar falhas inesperadas em ambos os ambientes (front e back).
- **Teste de Usabilidade:** Avaliação da facilidade de uso da interface, clareza da navegação e feedback visual ao usuário.
- **Teste de contrato(API):** Garantir o pleno funcionamento de todos os endpoints, especificando tipo da requisição, envio e recebimento de dados.

6. Cenários de Teste Planejados

A seguir, uma lista de cenários de alto nível planejados, agrupados por funcionalidade.

US-AUTH - /auth

ID	Cenário	Resultado Esperado
US-AUTH-TC01	[Happy Path] POST - Cadastrar um novo usuário (não administrador) com dados válidos.	Status 201 Created. O usuário é criado no banco de dados.
US-AUTH-TC02	POST - Cadastrar um usuário com e-mail já existente.	Status 400 Bad Request com a mensagem "User already exists or invalid data".
US-AUTH-TC03	POST - Cadastrar um usuário com e-mail em formato inválido (ex: "teste.com").	Status 400 Bad Request com mensagem de erro de validação de e-mail.
US-AUTH-TC04	POST - Cadastrar um usuário com senha fora do limite (ex: 4 caracteres).	Status 400 Bad Request com mensagem de erro sobre o tamanho da senha.

US-AUTH-TC05	[Happy Path] POST - Tentar realizar login com dados de um usuário válido.	Status 200 OK. Realiza o login.
US-AUTH-TC06	POST - Tentar realizar login com dados de um usuário inválido.	Status 400 Invalid credentials.
US-AUTH-TC07	GET - Acessar o perfil do usuário logado.	Status 200 OK, com os dados do usuário.
US-AUTH-TC08	GET - Acessar o perfil do usuário sem token nos headers.	401 - Not authorized, no token
US-AUTH-TC09	GET - Buscar um usuário com token de autenticação inválido.	Status 403 - Not authorized, invalid token.
US-AUTH-TC10	PUT - Atualizar dados do usuário com dados válidos (alteração de senha).	Status 200 OK Profile updated successfully
US-AUTH-TC11	PUT - Atualizar dados do usuário com dados inválidos.	Status 401 Current password is incorrect.
US-AUTH-TC12	PUT -Atualizar um usuário inexistente.	Status 404 User not found.

US-AUTH - /users

US-AUTH-TC13	[Happy Path] GET - Listar todos os usuários cadastrados.	Status 200 OK. Retorna uma lista com todos os usuários.
US-AUTH-TC14	GET - Tentar listar todos os	Status 401 Unauthorized.

	usuários cadastrados sem um token válido.	
US-AUTH-TC15	GET - Tentar listar todos os usuários cadastrados com usuário não admin.	Status 403 Forbidden.
US-AUTH-TC16	GET - Buscar um usuário por um ID existente.	Status 200 OK. Retorna os dados do usuário específico.
US-AUTH-TC17	GET - Buscar um usuário sem token de autenticação.	Status 401 Unauthorized.
US-AUTH-TC18	GET - Buscar um usuário com usuário não admin.	Status 403 Forbidden.
US-AUTH-TC19	GET - Buscar um usuário por um ID inexistente.	Status 404 not found.
US-AUTH-TC20	PUT - [Happy Path] Atualizar um usuário existente com dados válidos.	Status 200 OK. Os dados do usuário são atualizados.
US-AUTH-TC21	PUT - Atualizar um usuário informando dados inválidos.	400 Invalid input data.
US-AUTH-TC22	PUT - Atualizar um usuário sem ter um token de autenticação nos headers.	401 Not authorized
US-AUTH-TC23	PUT - Atualizar um usuário por meio de um usuário não admin.	403 Forbidden - Admin access required
US-AUTH-TC24	PUT - Atualizar um usuário informando um ID inexistente.	Status 404 not found.
US-AUTH-TC25	PUT - Atualizar para um e-mail que já está em uso por	Status 409 E-mail already in use com a mensagem "Este

	outro usuário.	email já está sendo usado".
US-AUTH-TC26	DELETE - [Happy Path] Deletar um usuário existente.	Status 200 OK com a mensagem "User deleted succesfully".
US-AUTH-TC27	DELETE - Deletar um usuário sem um token válido nos headers.	Status 401 Not authorized
US-AUTH-TC28	DELETE - Deletar um usuário sem permissão.	403 Unauthorized Forbidden - Admin access required
US-AUTH-TC29	DELETE - Deletar um usuário inexistente.	404 User not found
US-AUTH-TC30	DELETE - Deletar um usuário com reservas ativas.	Status 409 Cannot delete user with active reservations.

US-MOVIE - /movies

ID	Cenário	Resultado Esperado
USMOV-TC31	GET - [Happy Path] Consultar lista de filmes.	Status 200 OK.
USMOV-TC32	POST - [Happy Path] Criar um filme com dados válidos.	Status 201 Created. Movie created successfully.
USMOV-TC33	POST - Criar um filme com dados inválidos.	Status 400 Invalid input data.
USMOV-TC34	POST - Criar um filme com com um token inválido.	Status 401 Not Authorized.

USMOV-TC35	POST - Criar um filme com um usuário não admin.	Status 403 Forbidden - Admin access required
USMOV-TC36	GET - [Happy Path] Consultar um filme.	200 Movie details
USMOV-TC37	GET - Consultar filme com fomato inválido.	400 Invalid movie ID format "Invalid movie ID format or movie not found"
USMOV-TC38	GET - Consultar filme inexistente.	404 Movie not found
USMOV-TC39	PUT - [Happy Path] Atualizar um filme existente com dados válidos.	Status 200 Movie updated successfully
USMOV-TC40	PUT - Atualizar um filme informando dados inválidos.	400 Invalid input data.
USMOV-TC41	PUT - Atualizar um filme sem ter um token de autenticação nos headers.	401 Not authorized
USMOV-TC42	PUT - Atualizar um filme por meio de um usuário não admin.	403 Forbidden - Admin access required
USMOV-TC43	PUT - Atualizar um filme informando um ID inexistente.	Status 404 Movie not found.
USMOV-TC44	DELETE - [Happy Path] Deletar um filme existente.	Status 200 OK com a mensagem "Movie deleted successfully".
USMOV-TC45	DELETE - Deletar um usuário sem um token válido nos headers.	Status 401 Not authorized

USMOV-TC46	DELETE - Deletar um usuário sem permissão.	403 Forbidden - Admin access required
USMOV-TC47	DELETE - Deletar um usuário inexistente.	404 User not found

US-RESERVE - /reserve

ID	Cenário	Resultado Esperado
USRES-TC48	POST - Happy Path] Criar uma reserva com dados válidos.	Status 201 Created. Reservation created successfully.
USRES-TC49	POST - Criar uma reserva com dados inválidos.	Status 400 Invalid input data or seats already taken.
USRES-TC50	POST - Criar uma reserva com com um token inválido.	Status 401 Not Authorized.
USRES-TC51	POST - Criar uma reserva com sessão inválida.	Status 404 Session not found
USRES-TC52	GET - [Happy Path] Consultar reservas do usuário logado.	Status 200 OK User's reservations.
USRES-TC53	GET - Consultar reservas do usuário logado com token inválido.	Status 401 Not authorized.
USRES-TC54	GET - [Happy Path] Listar todas as reservas.	Status 200 OK. Retorna uma lista com todos as reservas.
USRES-TC55	GET - Tentar listar todas as reservas sem um token	Status 401 Unauthorized.

	válido.	
USRES-TC56	GET - Tentar listar todas as reservas com usuário não admin.	Status 403 Forbidden.
USRES-TC57	GET - Buscar as reservas de um usuário por um ID existente.	Status 200 OK. Retorna os dados do usuário específico.
USRES-TC58	GET - Buscar as reservas de um usuário sem token de autenticação.	Status 401 Unauthorized.
USRES-TC59	GET - Buscar uma reserva inválida.	Status 404 Reservation not found
USRES-TC60	PUT - [Happy Path] Atualizar uma reserva existente com dados válidos.	Status 200 Reservation updated successfully
USRES-TC61	PUT - Atualizar uma reserva informando dados inválidos.	400 Invalid status transition.
USRES-TC62	PUT - Atualizar uma reserva sem ter um token de autenticação nos headers.	401 Not authorized
USRES-TC63	PUT - Atualizar uma reserva por meio de um usuário não admin.	403 Forbidden - Admin access required
USRES-TC64	PUT - Atualizar um filme informando um ID inexistente.	Status 404 Reservation not found.
USRES-TC65	DELETE - [Happy Path] Deletar uma reserva existente.	Status 200 OK com a mensagem "Reservation deleted successfully".

USRES-TC66	DELETE - Deletar uma reserva sem um token válido nos headers.	Status 401 Not authorized
USRES-TC67	DELETE - Deletar uma reserva sem permissão.	403 Forbidden - Admin access required
USRES-TC68	DELETE - Deletar uma reserva inexistente.	404 Reservation not found

US-SESSIONS - /sessions

ID	Cenário	Resultado Esperado
USSNS-TC69	POST - [Happy Path] Criar uma sessão com dados válidos.	Status 201. Session created successfully.
USSNS-TC70	POST - Criar uma sessão com dados inválidos.	Status 400 Invalid input data.
USSNS-TC71	POST - Criar uma reserva com um token inválido.	Status 401 Not Authorized.
USSNS-TC72	POST - Criar uma sessão por meio de um usuário não admin.	Status 403 Forbidden.
USSNS-TC73	POST - Criar uma sessão com filme ou cinema inválidos.	Status 404 Movie or Theater not found.
USSNS-TC74	POST - Criar uma sessão com horários coincidentes.	Status 409 - Session conflict (time overlap).
USSNS-TC75	GET - [Happy Path]	Status 200 OK List of movie

	Consultar lista de sessões.	sessions.
USSNS-TC76	GET - [Happy Path] Consultar sessão pelo ID.	Status 200 Session details with populated movie and theater.
USSNS-TC77	GET - Consultar sessão por um ID inválido.	Status 404 Session not found.
USSNS-TC78	PUT - [Happy Path] Atualizar uma sessão existente com dados válidos.	Status 200 Session updated successfully
USSNS-TC79	PUT - Atualizar uma sessão informando dados inválidos.	400 Invalid input data.
USSNS-TC80	PUT - Atualizar uma sessão sem ter um token de autenticação nos headers.	401 Not authorized
USSNS-TC81	PUT - Atualizar uma sessão por meio de um usuário não admin.	403 Forbidden - Admin access required
USSNS-TC82	PUT - Atualizar uma sessão informando um ID inexistente.	Status 404 Session not found.
USSNS-TC83	PUT - Atualizar uma sessão com reservas existentes.	Status 409 Session has reservations, cannot update.
USSNS-TC84	DELETE - [Happy Path] Deletar uma sessão existente.	Status 200 OK com a mensagem "Session deleted successfully".
USSNS-TC85	DELETE - Deletar uma sessão sem um token válido nos headers.	Status 401 Not authorized
USSNS-TC86	DELETE - Deletar uma sessão sem permissão.	403 Forbidden - Admin access required

USSNS-TC87	DELETE - Deletar uma reserva inexistente.	404 Session not found
USSNS-TC88	DELETE - Deletar uma sessão com reservas confirmadas.	Status 409 com a mensagem Cannot delete session with confirmed reservations.
USSNS-TC89	PUT - Resetar assentos de uma sessão.	200 - Seats reset successfully
USSNS-TC90	PUT - Resetar assentos de uma sessão sem ter um token de autenticação nos headers.	401 Not authorized
USSNS-TC91	PUT - Resetar assentos de uma sessão por meio de um usuário não admin.	403 Forbidden - Admin access required
USSNS-TC92	PUT - Resetar assentos de uma sessão informando um ID inexistente.	Status 404 Session not found.

/theaters

ID	Cenário	Resultado Esperado
USTHE-TC93	GET - [Happy Path] Consultar lista de cinemas.	Status 200 OK. List of theaters
USTHE-TC94	POST - [Happy Path] Criar um cinema com dados válidos.	Status 201 Created. Theater created successfully.
USTHE-TC95	POST - Criar um cinema com dados inválidos.	Status 400 Invalid input data.

USTHE-TC96	POST - Criar um cinema com com um token inválido.	Status 401 Not Authorized.
USTHE-TC97	POST - Criar um cinema com um usuário não admin.	Status 403 Forbidden - Admin access required
USTHE-TC98	POST - Criar um cinema com nome repetido.	Status 409 Theater with that name already exists
USTHE-TC99	GET - [Happy Path] Consultar um cinema.	200 Movie details
USTHE-TC100	GET - Consultar cinema inexistente.	404 Theater not found
USTHE-TC101	PUT - [Happy Path] Atualizar um cinema existente com dados válidos.	Status 200 Movie updated successfully
USTHE-TC102	PUT - Atualizar um cinema informando dados inválidos.	400 Invalid input data.
USTHE-TC103	PUT - Atualizar um cinema sem ter um token de autenticação nos headers.	401 Not authorized
USTHE-TC104	PUT - Atualizar um cinema por meio de um usuário não admin.	403 Forbidden - Admin access required
USTHE-TC105	PUT - Atualizar um cinema informando um ID inexistente.	Status 404 Theater not found.
USTHE-TC106	PUT - Atualizar um cinema com nome que já está sendo utilizado.	Status 409 Theater name already in use

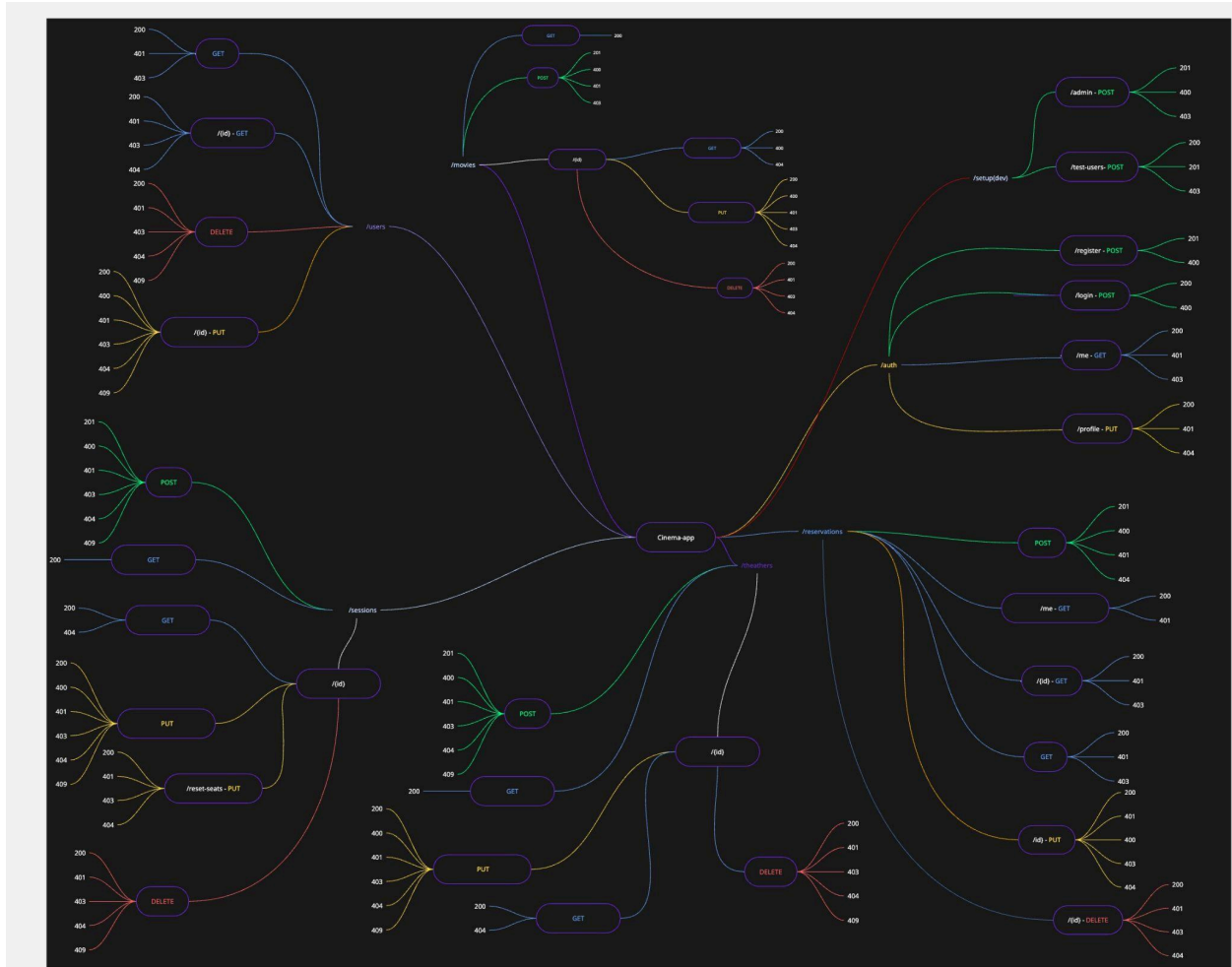
USTHE-TC107	DELETE - [Happy Path] Deletar um cinema existente.	Status 200 OK com a mensagem "Movie deleted successfully".
USTHE-TC108	DELETE - Deletar um cinema sem um token válido nos headers.	Status 401 Not authorized
USTHE-TC109	DELETE - Deletar um cinema sem permissão.	403 Unauthorized Forbidden - Admin access required
USTHE-TC110	DELETE - Deletar um cinema inexistente.	404 Theater not found
USTHE-TC111	DELETE - Deletar um cinema com sessões ativas.	409 Cannot delete theater with active sessions

7. Estratégia de Automação

- **Backend (API):** Utilizar o **Postman/Insomnia** para criar uma suíte de regressão automatizada que valide os contratos e regras de negócio da API. Executar essa suíte a cada nova build.
- **Frontend (E2E):** Utilizar Robot para automatizar os fluxos críticos do usuário (Happy Paths de registro, login e reserva). Esses testes garantem que as principais jornadas não foram quebradas.

8. Mapa Mental da Aplicação

A imagem a seguir, fornecida como base, ilustra a estrutura da aplicação Cinema-App e as funcionalidades a serem testadas.



9. Priorização da Execução dos Cenários

A execução seguirá uma priorização baseada na criticidade da funcionalidade para o negócio.

- **Prioridade Alta (Must Have):**

- Cenários de "Happy Path" de todas as funcionalidades (/users, /auth, /movies, /theaters, /sessions).
- Testes de autenticação e autorização.
- Validações críticas de negócio (e-mail único, nome de produto único).
- "Happy Path" do fluxo, desde cadastro até reserva.

- **Prioridade Média (Should Have):**

- Cenários de validação de dados (campos vazios, formatos inválidos, limites de senha).

- Fluxos alternativos.
- **Prioridade Baixa (Could Have):**
 - Cenários de teste para casos de borda menos comuns.
 - Testes exploratórios para encontrar falhas não previstas.
 - Cenário de cancelamento de compra.

10. Matriz de Risco

11. Cobertura de Testes

12. Testes Automatizados com Robot

13. Testes automatizados na collection

14. Cronograma

[illegible]