Plano de Testes para Cinema App

1. Identificação do Plano de Testes:

- Nome do projeto: Cinema App API & Frontend
- Responsável: Raique Alfredo Pereira de Ramos
- Data de criação: 24/06/2025

---

2. Objetivo do Plano:

Este plano de testes visa garantir a qualidade das funcionalidades do **Cinema App**, abrangendo o backend e o frontend. Ele cobre testes funcionais, fluxos de usuário, e validações críticas, incluindo cenários negativos e alternativos. O objetivo é validar que os recursos implementados atendem aos requisitos funcionais especificados na documentação técnica e nas User Stories. Além disso, busca identificar melhorias possíveis e assegurar a escalabilidade do sistema com base nos testes automatizados.

---

3. Escopo dos Testes:

Backend:

- **Endpoints Funcionais**:
  - Autenticação e Autorização (`/auth`).
  - Gerenciamento de usuários (`/users`).
  - Filmes (`/movies`).
  - Sessões (`/sessions`).
  - Reservas (`/reservations`).
  - Cinemas (`/theaters`).
- **Validações**:
  - Status HTTP adequados para cada tipo de requisição.
  - Fluxos completos, como criação de reservas e autenticação.
  - Cenários alternativos e negativos, como requisições com dados inválidos ou usuários sem autorização.
- **Regras de Negócio**:
  - Verificar restrições de acesso com base no nível de permissão (admin/usuário).
  - Validar consistência de dados, como não permitir sessões em horários conflitantes ou reservas em assentos ocupados.

Frontend:

- **Fluxos Funcionais**:
  - Login e registro de usuários.
  - Navegação de filmes e detalhes.
  - Processo de reserva (seleção de sessão, assentos e confirmação).
  - Gerenciamento de perfil do usuário.
- **Interface e Usabilidade**:
  - Responsividade para diferentes tamanhos de tela.
  - Mensagens de erro claras para validações no frontend.

Fora do escopo:

- Testes de carga e estresse no backend e frontend.
- Testes de integração com sistemas externos.
- Automação de testes não funcionais, como performance.

---

4. Análise do Cinema App

Backend:

A API do Cinema App fornece suporte robusto para autenticação, gerenciamento de filmes, sessões, reservas e cinemas. A análise inclui os seguintes pontos:

1. **Autenticação e Autorização**:
   - **Endpoints:** `/auth/register`, `/auth/login`, `/auth/me`.
   - Validações:
     - Prevenção de registros duplicados (e-mails únicos).
     - Uso de JWT para manter sessões seguras.
   - Fluxos principais:
     - Registro e login de usuários.
     - Logout para invalidação de sessão.
   - Cenários negativos:
     - Tentativas de login com credenciais inválidas.
     - Acesso a rotas protegidas sem token ou com token inválido/expirado.
2. **Gerenciamento de Usuários**:
   - **Endpoints:** `/users`, `/users/:id`.
   - Recursos:
     - Exibição e atualização de perfil de usuário.
     - Operações de CRUD com permissões específicas para administradores.
3. **Filmes**:
   - **Endpoints:** `/movies`, `/movies/:id`.
   - Recursos:
     - Listagem de filmes com filtros opcionais.
     - Detalhamento de filmes, incluindo informações ricas como sinopse, elenco e diretor.
   - Validações:
     - Garantia de dados consistentes e ausência de duplicidade de entradas.
4. **Sessões**:
   - **Endpoints:** `/sessions`, `/sessions/:id`.
   - Recursos:
     - Exibição de horários de sessões para filmes específicos.
     - Integração com o gerenciamento de reservas e cinemas.
   - Dependências:
     - Sessões estão vinculadas a filmes e cinemas existentes.
5. **Reservas**:
   - **Endpoints:** `/reservations`, `/reservations/me`, `/reservations/:id`.
   - Recursos:
     - Criação, listagem e cancelamento de reservas.

        - Visualização de reservas do usuário atual e todas as reservas (admin).
    - Validações:
        - Prevenção de reservas duplicadas para o mesmo assento/sessão.
        - Assentos ocupados não podem ser reservados novamente.
6. **Cinemas**:
    - **Endpoints:** `/theaters`, `/theaters/:id`.
    - Recursos:
        - Cadastro e detalhamento de cinemas.
    - Cenários:
        - Garantir que o cinema seja criado corretamente e que todos os detalhe do cinema estejam visíveis.

Front-end:

A interface do Cinema App apresenta fluxos organizados para usuários e administradores. A análise inclui os seguintes pontos:

1. **Navegação e Experiência do Usuário**:
    - O menu principal exibe opções claras e é responsivo.
    - Navegação intuitiva e feedback visual.
    - Diferenciação de funcionalidades para visitantes, usuários autenticados e administradores.
2. **Gerenciamento de Filmes**:
    - Listagem de filmes com informações claras.
    - Detalhamento com sinopse, elenco e horários disponíveis.
    - Cards responsivos e visualmente atraentes.
3. **Reserva de Assentos**:
    - Visualização do layout de assentos com codificação de cores.
    - Feedback visual ao selecionar/desmarcar assentos.
    - Integração com o processo de checkout.
4. **Checkout e Confirmação**:
    - Fluxo de pagamento simulado com suporte a múltiplos métodos (cartão, PIX).
    - Resumo das reservas com valores claros.
5. **Painel Administrativo**:
    - Recursos para gerenciar filmes, sessões e cinemas.
    - Visualização de todas as reservas e relatórios básicos.
6. **Responsividade e Usabilidade**:
    - Suporte para diferentes tamanhos de tela, adaptando a experiência tanto para dispositivos móveis quanto para desktops.
    - Mensagens claras de erro e sucesso em ações do usuário.

Dependências Identificadas

1. **Autenticação**:
    - Acesso a recursos como reservas e gerenciamento de sessões depende de login bem-sucedido.
2. **Filmes e Sessões**:
    - Reservas dependem de sessões associadas a filmes válidos.
    - Sessões só podem ser criadas se o filme e o cinema existirem.
3. **Cinemas e Reservas**:
    - Layouts de assentos nos cinemas precisam ser precisos para evitar inconsistências nas reservas.
4. **Frontend e Backend**:
    - Comunicação robusta entre a interface e a API via endpoints documentados.

---

5. Técnicas Aplicadas:

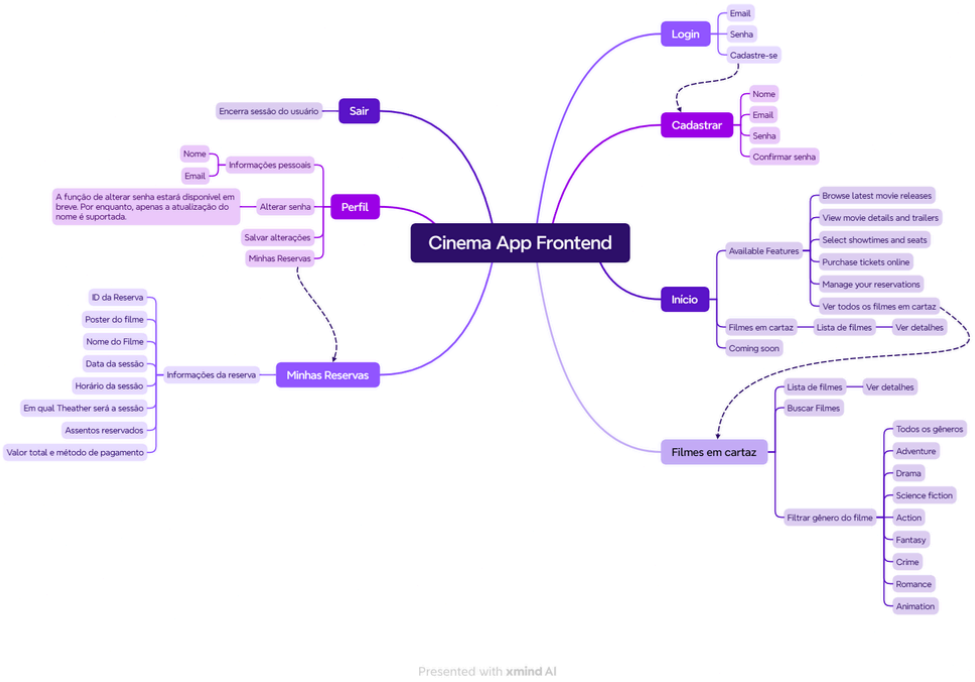
- **Testes Funcionais:** Verificação de funcionalidades principais conforme requisitos.
- **Testes de Fluxos Completo:** Validação de cenários ponta a ponta, como criar uma reserva e gerenciar filmes.
- **Testes Baseados em Requisitos:** Testes planejados com base nas User Stories e documentação.
- **Particionamento de Equivalência:** Para entradas de dados em campos como datas e IDs.
- **Testes de Interface:** Usabilidade e layout, usando ferramentas como Playwright.

---

6. Mapa Mental:

Cinema App Api:

Cinema App Frontend:



---

7. Cenários de Testes Planejados:

Cenários de Testes para Autenticação

US-AUTH-001: Registro de Usuário

Back-end

- **ID:** AUTH-BE-001
- **Título:** Registro de usuário com dados válidos
- **Pré-condição:** Nenhum usuário com o mesmo e-mail deve existir.
- **Passos:**
  a. Fazer uma requisição `POST` na rota `/api/v1/auth/register`.
  b. Enviar no payload os campos `name`, `email` e `password` com valores válidos.
- **Resultado Esperado:**
  - Retorna **status 201 (Created)**.
  - A resposta contém os dados do usuário criado, exceto a senha.

Front-end

- **ID:** AUTH-FE-001
- **Título:** Registro de usuário pelo formulário na interface
- **Pré-condição:** O campo e-mail não deve estar registrado.
- **Passos:**
  a. Acessar a página de registro.
  b. Preencher os campos com dados válidos.
  c. Submeter o formulário.
- **Resultado Esperado:**
  - O usuário é registrado e redirecionado para a página inicial.

- Mensagem de sucesso exibida.

US-AUTH-002: Login de Usuário

Back-end

- **ID:** AUTH-BE-002
- **Título:** Login com credenciais válidas
- **Pré-condição:** O usuário deve estar registrado.
- **Passos:**
  a. Fazer uma requisição `POST` na rota `/api/v1/auth/login`.
  b. Enviar no payload os campos `email` e `password` corretos.
- **Resultado Esperado:**
  - Retorna **status 200 (OK)**.
  - A resposta contém um **token JWT** válido.

Front-end

- **ID:** AUTH-FE-002
- **Título:** Login pela interface
- **Pré-condição:** O usuário deve estar registrado.
- **Passos:**
  a. Acessar a página de login.
  b. Inserir e-mail e senha corretos.
  c. Submeter o formulário.
- **Resultado Esperado:**
  - O usuário é redirecionado para a página inicial.

US-AUTH-003: Logout de Usuário

Back-end

- **ID:** AUTH-BE-003
- **Título:** Logout do usuário autenticado
- **Pré-condição:** O usuário deve estar autenticado e possuir um token JWT válido.
- **Passos:**
  a. Fazer uma requisição `POST` na rota `/api/v1/auth/logout`.
  b. Garantir que o token JWT é enviado no cabeçalho da requisição.
- **Resultado Esperado:**
  - Retorna **status 200 (OK)**.
  - A resposta confirma que o logout foi concluído.
  - Token JWT é invalidado no sistema.

Front-end

- **ID:** AUTH-FE-003
- **Título:** Logout através do menu de navegação
- **Pré-condição:** O usuário deve estar logado.
- **Passos:**
  a. Clicar na opção "Sair" no menu de navegação.
  b. Verificar se o usuário é redirecionado para a página de login.
  c. Tentar realizar uma reserva após logout.
- **Resultado Esperado:**

- O usuário é desconectado e redirecionado para a página de login.
- Ao tentar concluir a reserva, é redirecionado para página de login.

US-AUTH-004: Visualizar informações do Usuário

Back-end

- **ID:** AUTH-BE-004
- **Título:** Visualizar informações do perfil
- **Pré-condição:** O usuário deve estar autenticado.
- **Passos:**
  a. Fazer uma requisição `GET` na rota `/api/v1/auth/me`.
- **Resultado Esperado:**
  - Retorna **status 200 (OK)**.
  - A resposta contém informações como nome, e-mail e função do usuário.

Front-end

- **ID:** AUTH-FE-004
- **Título:** Exibir informações do perfil
- **Pré-condição:** O usuário deve estar logado.
- **Passos:**
  a. Navegar até a página de perfil.
  b. Confirmar se as informações do usuário (nome e e-mail) são exibidas corretamente.
- **Resultado Esperado:**
  - As informações do perfil são exibidas corretamente na interface.

US-AUTH-005: Atualizar informações do Usuário

Back-end

- **ID:** AUTH-BE-005
- **Título:** Atualizar informações do perfil
- **Pré-condição:** O usuário deve estar autenticado.
- **Passos:**
  a. Fazer uma requisição `PUT` na rota `/api/v1/auth/profile`.
  b. Enviar no payload os campos que o usuário deseja modificar (exemplo: `name`).
- **Resultado Esperado:**
  - Retorna **status 200 (OK)**, confirmando a atualização.
  - Retorna mensagem que contém os dados com a atualização.

Front-end

- **ID:** AUTH-FE-005
- **Título:** Editar e salvar alterações no perfil
- **Pré-condição:** O usuário deve estar logado.
- **Passos:**
  a. Navegar até a página de perfil.
  b. Editar o campo desejado (exemplo: nome) e salvar alterações.
  c. Retornar à página de perfil para verificar se as informações atualizadas estão refletidas.
- **Resultado Esperado:**
  - Mensagem de sucesso exibida após a edição.
  - Alterações no perfil são salvas e refletem na interface.

US-AUTH-006: Registro de Usuário com E-mail Já Existente

Back-end

ID: AUTH-BE-006
Título: Registro de usuário com e-mail já existente
Pré-condição: Deve existir um usuário registrado com o mesmo e-mail.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/auth/register`.
2. Enviar no payload os campos `name`, `email` (já registrado) e `password` com valores válidos.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém uma mensagem de erro indicando que o e-mail já está em uso.

Front-end

ID: AUTH-FE-006
Título: Registro de usuário com e-mail já existente pela interface
Pré-condição: Deve existir um usuário registrado com o mesmo e-mail.
Passos:

1. Acessar a página de registro.
2. Preencher o formulário com um e-mail já registrado e outros campos válidos.
3. Submeter o formulário.

Resultado Esperado:

- Uma mensagem de erro é exibida indicando que o e-mail já está em uso.
- O registro não é realizado, e o usuário permanece na página de registro.

US-AUTH-007: Login com Credenciais Inválidas

Back-end

ID: AUTH-BE-007
Título: Login com e-mail ou senha inválidos
Pré-condição: O usuário deve estar registrado.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/auth/login`.
2. Enviar no payload um `email` inexistente ou uma `password` incorreta.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém uma mensagem de erro indicando credenciais inválidas.

Front-end

ID: AUTH-FE-007
Título: Login com e-mail ou senha inválidos pela interface
Pré-condição: O usuário deve estar registrado.
Passos:

1. Acessar a página de login.
2. Inserir um e-mail inexistente ou uma senha incorreta.
3. Submeter o formulário.

Resultado Esperado:

- Uma mensagem de erro é exibida informando credenciais inválidas.
- O login não é realizado, e o usuário permanece na página de login.

US-AUTH-008: Atualizar Informações do Perfil com Dados Inválidos

Back-end

ID: AUTH-BE-008
Título: Atualizar perfil com informações inválidas
Pré-condição: O usuário deve estar autenticado.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/auth/profile`.
2. Enviar no payload campos inválidos (exemplo: `name` vazio ou `senha` com formato inválido).

Resultado Esperado:

- Retorna status 401 (Unauthorized).
- A resposta contém mensagens de erro específicas para os campos inválidos.

Front-end

ID: AUTH-FE-008
Título: Atualizar perfil com informações inválidas pela interface
Pré-condição: O usuário deve estar logado.
Passos:

1. Navegar até a página de perfil.
2. Inserir valores inválidos nos campos editáveis (exemplo: deixar o campo `name` vazio ou inserir um e-mail inválido).
3. Salvar as alterações.

Resultado Esperado:

- Mensagem de erro é exibida para cada campo inválido.
- As alterações não são salvas, e os dados antigos permanecem na interface.

Cenários de Testes para Usuários

US-USER-001: Listar todos os usuários

Back-end
ID: USER-BE-001
Título: Listar todos os usuários
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição GET na rota `/api/v1/users`.
2. Incluir o token JWT do administrador no cabeçalho.

Resultado Esperado:

- Retorna status 200 (OK).
- A resposta contém uma lista de usuários cadastrados.

US-USER-002: Obter detalhes de um usuário por ID

Back-end
ID: USER-BE-002
Título: Obter detalhes de um usuário por ID

Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição GET na rota `/api/v1/users/{id}`.
2. Substituir `{id}` pelo ID do usuário desejado.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 200 (OK).
- A resposta contém os detalhes do usuário especificado.

US-USER-003: Excluir um usuário por ID

Back-end
ID: USER-BE-003
Título: Excluir um usuário por ID
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição DELETE na rota `/api/v1/users/{id}`.
2. Substituir `{id}` pelo ID do usuário a ser excluído.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 200 (OK).
- Retorna mensagem informando que o usuário foi deletado com sucesso.

US-USER-004: Obter detalhes de um usuário por ID com ID incorreto ou inexistente

Back-end

ID: USER-BE-004
Título: Obter detalhes de um usuário com ID incorreto ou inexistente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição GET na rota `/api/v1/users/{id}`.
2. Substituir `{id}` por um ID inválido ou inexistente.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 404 (Not Found).
- A resposta contém uma mensagem de erro indicando que o usuário não foi encontrado ou o ID é inválido.

US-USER-005: Excluir um usuário por ID incorreto ou inexistente

Back-end

ID: USER-BE-005
Título: Excluir um usuário com ID incorreto ou inexistente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição DELETE na rota `/api/v1/users/{id}`.
2. Substituir `{id}` por um ID inválido ou inexistente.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 404 (Not Found).
- A resposta contém uma mensagem de erro indicando que o usuário não foi encontrado.

US-USER-006: Atualizar informações de um usuário existente

Back-end

ID: USER-BE-006
Título: Atualizar informações de um usuário existente
Pré-condição:

- O usuário deve estar autenticado como administrador.
- O ID do usuário a ser atualizado deve existir no sistema.

Passos:

1. Fazer uma requisição **PUT** na rota `/api/v1/users/{id}`.
2. Substituir `{id}` pelo ID válido do usuário a ser atualizado.
3. Incluir o token JWT do administrador no cabeçalho da requisição.
4. Enviar no **request body** os campos a serem modificados.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém a mensagem de sucesso e os dados atualizados do usuário.

Cenários de Testes para Filmes

US-HOME-001: Página Inicial Atrativa

Front-end

- **ID:** HOME-FE-001
- **Título:** Verificar layout da página inicial
- **Pré-condição:** O usuário deve acessar a página inicial.
- **Passos:**
  a. Navegar até a página inicial.
  b. Verificar se o banner com informações do cinema está presente.
  c. Confirmar a exibição da seção "Filmes em Cartaz" com pôsteres em tamanho adequado.
  d. Testar a responsividade redimensionando a janela do navegador.
  e. Verificar links rápidos e menu de navegação.
- **Resultado Esperado:**
  - Banner e seção "Filmes em Cartaz" estão visíveis.
  - Layout ajusta-se corretamente para diferentes tamanhos de tela.
  - Links rápidos e menu principal funcionam conforme esperado.

US-MOVIE-001: Navegar na Lista de Filmes

Back-end

- **ID:** MOVIE-BE-001
- **Título:** Listar todos os filmes disponíveis
- **Pré-condição:**
  - Deve haver filmes cadastrados.
  - Parameters: title; genre; sort; limit; page.
- **Passos:**
  a. Fazer uma requisição `GET` na rota `/api/v1/movies`.

- **Resultado Esperado:**
  - Retorna **status 200 (OK)**.
  - A resposta contém uma lista de filmes com detalhes: título, sinopse, diretor, gêneros, duração e data de lançamento.

Front-end

- **ID:** MOVIE-FE-001
- **Título:** Exibir lista de filmes na página inicial
- **Pré-condição:** O back-end deve retornar filmes na listagem.
- **Passos:**
  a. Acessar a página inicial.
  b. Verificar se os filmes são exibidos em formato de grid.
  c. Verificar se cada card de filme exibe título, sinopse, diretor, gêneros, duração e data de lançamento.
  d. Clicar em um card de filme.
- **Resultado Esperado:**
  - Filmes são exibidos corretamente com título, classificação, gêneros, duração e data de lançamento.
  - Cada card é clicável e redireciona para os detalhes do filme.
  - O layout é responsivo.

US-MOVIE-002: Visualizar Detalhes do Filme

Back-end

- **ID:** MOVIE-BE-002
- **Título:** Detalhar informações de um filme
- **Pré-condição:** O ID do filme deve existir.
- **Passos:**
  a. Fazer uma requisição `GET` na rota `/api/v1/movies/{id}`.
- **Resultado Esperado:**
  - Retorna **status 200 (OK)**.
  - A resposta contém detalhes como title, sinopse, elenco, diretor, gênero, data de lançamento e duração.

Front-end

- **ID:** MOVIE-FE-002
- **Título:** Exibir detalhes do filme na interface
- **Pré-condição:** O filme deve existir.
- **Passos:**
  a. Ir até a página filmes em cartaz.
  b. Clicar em ver detalhes em um filme.
  c. Verificar se a página de detalhes exibe title, sinopse, elenco, diretor, data de lançamento, duração e horários disponíveis.
  d. Clicar no horário de uma sessão.
- **Resultado Esperado:**
  - A página exibe todos os detalhes do filme, incluindo sinopse, elenco, diretor, data de lançamento, duração, horários e pôster do filme.
  - É possível navegar para a reserva a partir dos horários disponíveis.

US-MOVIE-003: Criar um novo filme

Back-end

ID: MOVIE-BE-003

Título: Criar um novo filme com dados válidos
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/movies`.
2. Enviar no payload os campos `title`, `synopsis`, `director`, `genres`, `duration`, `classification`, `poster` e `releaseDate` com valores válidos.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 201 (Created).
- A resposta contém os dados do filme criado.

US-MOVIE-004: Atualizar um filme existente

Back-end
ID: MOVIE-BE-004
Título: Atualizar um filme existente com dados válidos
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/movies/{id}`.
2. Substituir `{id}` pelo ID do filme a ser atualizado.
3. Enviar no payload os campos a serem modificados.
4. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 200 (OK).
- A resposta contém os dados atualizados do filme.

US-MOVIE-005: Excluir um filme por ID

Back-end
ID: MOVIE-BE-005
Título: Excluir um filme por ID
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição DELETE na rota `/api/v1/movies/{id}`.
2. Substituir `{id}` pelo ID do filme a ser excluído.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 200 (OK).
- Retorna a mensagem informando que o filme foi removido com sucesso.

US-MOVIE-006: Criar um filme com informações inválidas ou inexistentes

Back-end

ID: MOVIE-BE-006
Título: Criar um novo filme com payload inválido ou ausente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/movies`.
2. Enviar no payload campos incompletos, inválidos ou ausentes.

3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém mensagens de erro indicando os campos inválidos ou ausentes.

US-MOVIE-007: Atualizar um filme existente com informações inválidas ou inexistentes

Back-end

ID: MOVIE-BE-007
Título: Atualizar um filme existente com payload inválido ou ausente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/movies/{id}`.
2. Substituir `{id}` pelo ID válido do filme.
3. Enviar no payload campos inválidos ou ausentes.
4. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém mensagens de erro indicando os campos inválidos ou ausentes.

US-MOVIE-008: Excluir um filme com ID incorreto ou inexistente

Back-end

ID: MOVIE-BE-008
Título: Excluir um filme com ID incorreto ou inexistente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição DELETE na rota `/api/v1/movies/{id}`.
2. Substituir `{id}` por um ID inválido ou inexistente.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 404 (Not Found).
- A resposta contém uma mensagem de erro indicando que o filme não foi encontrado.

Cenários de Testes para Cinemas

US-THEATERS-001: Listar todos os cinemas

Back-end
ID: THEATERS-BE-001
Título: Listar todos os cinemas cadastrados
Pré-condição: Nenhuma.
Passos:

1. Fazer uma requisição GET na rota `/api/v1/theaters`.

Resultado Esperado:

- Retorna status 200 (OK).
- A resposta contém uma lista de cinemas com detalhes como nome, localização e capacidade.

US-THEATERS-002: Obter Detalhes do Cinema

Back-end
ID: THEATERS-BE-002
Título: Obter detalhes de um cinema existente

Pré-condição: O ID do cinema deve existir.

Passos:

1. Fazer uma requisição **GET** na rota `/api/v1/theaters/{id}`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém os detalhes completos do cinema (nome, localização, capacidade, etc.).

US-THEATERS-003: Criar um novo cinema

Back-end
ID: THEATERS-BE-003
Título: Criar um novo cinema com dados válidos
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/theaters`.
2. Enviar no payload os campos `name`, `capacity` e `type` com valores válidos.
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 201 (Created).
- A resposta contém os dados do cinema criado.

US-THEATERS-004: Atualizar um cinema existente

Back-end
ID: THEATERS-BE-004
Título: Atualizar um cinema existente com dados válidos
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/theaters/{id}`.
2. Substituir `{id}` pelo ID do cinema a ser atualizado.
3. Enviar no payload os campos a serem modificados.
4. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 200 (OK).
- A resposta contém os dados atualizados do cinema.

US-THEATERS-005: Excluir um cinema por ID

Back-end
ID: THEATERS-BE-005
Título: Excluir um cinema por ID
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição DELETE na rota `/api/v1/theaters/{id}`.
2. Substituir `{id}` pelo ID do cinema a ser excluído.

3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 204 (No Content).
- O cinema é removido do sistema.

US-THEATERS-006: Criar um novo cinema com payload inválido ou ausente

Back-end

ID: THEATERS-BE-006
Título: Tentar criar um novo cinema com payload inválido ou ausente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/theaters`.
2. Enviar no payload campos incompletos, inválidos ou ausentes (ex.: `name` vazio ou `capacity` negativo).
3. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém mensagens de erro detalhando os campos inválidos ou ausentes.

US-THEATERS-007: Atualizar um cinema existente com payload inválido ou ausente

Back-end

ID: THEATERS-BE-007
Título: Tentar atualizar um cinema com payload inválido ou ausente
Pré-condição: O usuário deve estar autenticado como administrador.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/theaters/{id}`.
2. Substituir `{id}` por um ID válido.
3. Enviar no payload campos inválidos (ex.: `capacity` como string ou `type` fora dos valores permitidos) ou omitir campos obrigatórios.
4. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém mensagens de erro indicando os problemas no payload.

Cenários de Testes para Sessões

US-SESSIONS-001: Listar Todas as Sessões

Back-end
ID: SESSIONS-BE-001
Título: Listar todas as sessões disponíveis

Pré-condição:

- Devem existir sessões cadastradas no sistema.
- Paramters:
  - Filtrar por ID do filme e teatro
  - Filtrar por data (YYYY-MM-DD)
  - Número de sessões a retornar - *Valor padrão* : 10

- Número da página - *Valor padrão* : 1

Passos:

- Fazer uma requisição GET na rota /api/v1/sessions.

Resultado Esperado:

- Retorna status 200 (OK).
- A resposta contém uma lista de sessões de filmes.

Front-end
ID: SESSIONS-FE-001
Título: Visualizar detalhes de uma sessão na interface

Pré-condição: A sessão deve estar cadastrada.

Passos:

1. Acessar a interface principal do sistema.
2. Navegar até a seção de **Filmes em Cartaz**.
3. Selecionar um filme e clicar em **Ver Detalhes**.
4. Na página de detalhes do filme, verificar a lista de **Sessões Disponíveis**.
5. Observar as informações apresentadas para cada sessão.

Resultado Esperado:
A interface exibe corretamente os detalhes das sessões disponíveis, incluindo:

- Nome do cinema (theater),
- Tipo de sala (por exemplo, 3D),
- Valor do ingresso **inteiro** e **meia-entrada**.

US-SESSIONS-002: Obter Detalhes de uma Sessão

Back-end
ID: SESSIONS-BE-002
Título: Obter detalhes de uma sessão específica

Pré-condição: O ID da sessão deve existir.

Passos:

1. Fazer uma requisição **GET** na rota `/api/v1/sessions/{id}`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém os detalhes completos da sessão.

US-SESSIONS-003: Criar uma Nova Sessão

Back-end
ID: SESSIONS-BE-003
Título: Criar uma nova sessão (administrador)

Pré-condição: O usuário deve ser um administrador.

Passos:

1. Fazer uma requisição **POST** na rota `/api/v1/sessions`.
2. Enviar no payload os dados da sessão (filme, cinema, horário, etc.).

Resultado Esperado:

- Retorna status **201 (Created)**.
- A resposta contém os detalhes da sessão criada.

US-SESSIONS-004: Atualizar uma Sessão

Back-end
ID: SESSIONS-BE-004
Título: Atualizar informações de uma sessão existente (administrador)

Pré-condição: A sessão deve existir e o usuário deve ser um administrador.

Passos:

1. Fazer uma requisição **PUT** na rota `/api/v1/sessions/{id}`.
2. Enviar no payload os dados atualizados da sessão.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém "Session updated successfully" e dados da sessão atualizada.

US-SESSIONS-005: Excluir uma Sessão

Back-end
ID: SESSIONS-BE-005
Título: Excluir uma sessão existente (administrador)

Pré-condição: A sessão deve existir e o usuário deve ser um administrador.

Passos:

1. Fazer uma requisição **DELETE** na rota `/api/v1/sessions/{id}`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- Retorna a mensagem: "Session deleted successfully".

US-SESSIONS-006: Criar uma nova sessão com payload inválido ou ausente

Back-end

ID: SESSIONS-BE-006

Título: Tentar criar uma nova sessão com payload inválido ou ausente
Pré-condição: O usuário deve ser um administrador.
Passos:

- Fazer uma requisição POST na rota /api/v1/sessions.
- Enviar no payload campos obrigatórios omitido.
- Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 404 (Not Foundt).
- Retorna mensagem de erro adequada.

US-SESSIONS-007: Atualizar uma sessão com payload inválido ou ausente

Back-end

ID: SESSIONS-BE-007
Título: Tentar atualizar uma sessão com payload inválido ou ausente
Pré-condição: A sessão deve existir, e o usuário deve ser um administrador.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/sessions/:id`.
2. Substituir `:id` por um ID válido.
3. Enviar no payload campos inválidos (ex.: `data` em formato errado ou `cinema` não existente) ou omitir campos obrigatórios.
4. Incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém “Invalid input data” / mensagens de erro indicando os problemas no payload.

US-SESSIONS-008: Resetar os assentos de uma sessão existente para disponíveis

Back-end

ID: SESSIONS-BE-008

Título: Resetar os assentos de uma sessão existente para disponíveis

Pré-condição: A sessão deve existir, e o usuário deve ser autenticado como administrador.

Passos:

1. Fazer uma requisição **PUT** na rota `/api/v1/sessions/{id}/reset-seats`.
2. Substituir `{id}` por um ID válido de sessão.
3. Incluir o token JWT no cabeçalho de autorização.

Resultado Esperado:

- Retorna status 200 (OK)
- Retorna uma mensagem de sucesso informando que os assentos foram resetados para disponíveis.

Front-end

ID: SESSIONS-FE-008

Título: Resetar os assentos de uma sessão existente para disponíveis pela interface

Pré-condição: O usuário deve estar logado como administrador e a sessão deve existir.

Passos:

1. Fazer login com uma conta de administrador.
2. Acessar a lista de filmes disponíveis.
3. Selecionar um filme e visualizar seus detalhes.
4. Acessar uma sessão vinculada ao filme e ir para a visualização de assentos.
5. Clicar na opção de "Resetar assentos" (exibida apenas para administradores).
6. Confirmar a ação de reset.

Resultado Esperado:

Uma mensagem de sucesso é exibida informando que todos os assentos da sessão foram resetados para o status "disponível".
A visualização dos assentos é atualizada na interface, refletindo os novos estados.

Cenários de Testes para Reservas

US-RESERVATIONS-001: Listar Todas as Reservas

Back-end
ID: RESERVATIONS-BE-001
Título: Listar todas as reservas (somente administrador)

Pré-condição:

- O usuário deve ser um administrador e devem existir reservas no sistema.
- Parameters:
  - Authorization; page; limit.

Passos:

1. Fazer uma requisição **GET** na rota `/api/v1/reservations`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém uma lista de todas as reservas.

US-RESERVATIONS-002: Obter Reservas do Usuário Atual

Back-end
ID: RESERVATIONS-BE-002
Título: Obter reservas do usuário autenticado

Pré-condição: O usuário deve estar autenticado e ter reservas registradas.

Passos:

1. Fazer uma requisição **GET** na rota `/api/v1/reservations/me`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém uma lista de reservas do usuário.

Front-end
ID: RESERVATIONS-FE-002
Título: Visualizar reservas pessoais na interface

Pré-condição: O usuário deve estar autenticado e ter reservas registradas.

Passos:

1. Navegar até a página de "Minhas Reservas".
2. Confirmar se as reservas do usuário são exibidas corretamente.

Resultado Esperado:

- A página exibe a lista de reservas do usuário com detalhes.

US-RESERVATIONS-003: Obter Detalhes de uma Reserva

Back-end
ID: RESERVATIONS-BE-003
Título: Obter detalhes de uma reserva específica

Pré-condição: A reserva deve existir.

Passos:

1. Fazer uma requisição **GET** na rota `/api/v1/reservations/{id}`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta contém os detalhes da reserva.

US-RESERVATIONS-004: Criar uma Nova Reserva

Back-end
ID: RESERVATIONS-BE-004
Título: Criar uma nova reserva

Pré-condição: O usuário deve estar autenticado.

Passos:

1. Fazer uma requisição **POST** na rota `/api/v1/reservations`.
2. Enviar no payload os dados da reserva (assentos, sessão, etc.).

Resultado Esperado:

- Retorna status **201 (Created)**.
- A resposta contém os detalhes da reserva criada.

Front-end
ID: RESERVATIONS-FE-004
Título: Criar reserva na interface

Pré-condição: O usuário deve estar logado.

Passos:

1. Navegar até a página de filmes em cartaz.
2. Selecionar ver detalhes em qualquer filme.
3. Nas sessões disponíveis, clicar em selecionar assentos.
4. Escolher os assentos e continuar para pagamento.
5. Selecionar método de pagamento e clicar em finalizar compra.

Resultado Esperado:

- A reserva é criada com sucesso.
- Uma mensagem de confirmação é exibida.

US-RESERVATIONS-005: Atualizar Status de uma Reserva

Back-end
ID: RESERVATIONS-BE-005
Título: Atualizar status de uma reserva (somente administrador)

Pré-condição: A reserva deve existir e o usuário deve ser um administrador.

Passos:

1. Fazer uma requisição **PUT** na rota `/api/v1/reservations/{id}`.
2. Enviar no payload o novo status da reserva e o status do pagamento.

Resultado Esperado:

- Retorna status **200 (OK)**.
- A resposta confirma a atualização da reserva e retornar seus dados.

US-RESERVATIONS-006: Excluir uma Reserva

Back-end

ID: RESERVATIONS-BE-006

Título: Excluir uma reserva existente (somente administrador)

Pré-condição: A reserva deve existir e o usuário deve ser um administrador.

Passos:

1. Fazer uma requisição **DELETE** na rota `/api/v1/reservations/{id}`.

Resultado Esperado:

- Retorna status **200 (OK)**.
- Retorna a mensagem de sucesso “Reservation deleted successfully“.

US-RESERVATIONS-007: Obter Reservas do Usuário Atual sem auth

Back-end

ID: RESERVATIONS-BE-007

Título: Tentar obter reservas do usuário atual sem autenticação

Pré-condição: O usuário não deve estar autenticado.

Passos:

1. Fazer uma requisição GET na rota `/api/v1/reservations/me` sem incluir o token JWT no cabeçalho.

Resultado Esperado:

- Retorna status 401 (Unauthorized).
- A resposta contém uma mensagem de erro informando a necessidade de autenticação.

Front-end

ID: RESERVATIONS-FE-007

Título: Tentar acessar reservas pessoais sem autenticação

Pré-condição: O usuário não deve estar logado.

Passos:

1. Navegar até a página de filmes em cartaz.
2. Selecionar ver detalhes em qualquer filme.
3. Nas sessões disponíveis, clicar em selecionar assentos.
4. Escolher os assentos e continuar para pagamento.
5. Selecionar método de pagamento e clicar em finalizar compra.

Resultado Esperado:

- O sistema redireciona o usuário para a página de login.
- Uma mensagem de erro ou alerta informa que é necessário estar logado para acessar reservas.

US-RESERVATIONS-008: Obter Detalhes de uma Reserva com ID inválido ou inexistente

Back-end

ID: RESERVATIONS-BE-008

Título: Tentar obter detalhes de uma reserva inexistente

Pré-condição: A reserva com o ID fornecido não deve existir.

Passos:

1. Fazer uma requisição GET na rota `/api/v1/reservations/{id}` substituindo `{id}` por um ID inválido ou inexistente.

Resultado Esperado:

- Retorna status 404 (Not Found).
- A resposta contém uma mensagem de erro informando que a reserva não foi encontrada.

US-RESERVATIONS-009: Criar uma Nova Reserva com payload inválido ou ausente

Back-end

ID: RESERVATIONS-BE-009
Título: Tentar criar uma nova reserva com dados inválidos ou ausentes
Pré-condição: O usuário deve estar autenticado.
Passos:

1. Fazer uma requisição POST na rota `/api/v1/reservations`.
2. Enviar no payload dados inválidos (ex.: assentos inexistentes ou sessão inválida) ou omitir campos obrigatórios.

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém mensagens de erro indicando os problemas no payload.

Front-end

ID: RESERVATIONS-FE-009
Título: Tentar criar uma nova reserva com dados inválidos pela interface
Pré-condição: O usuário deve estar logado.
Passos:

1. Navegar até a página de seleção de assentos.
2. Escolher assentos inválidos (ex.: assentos já ocupados ou fora do mapa da sessão).

Resultado Esperado:

- O sistema não permite selecionar assentos já ocupados.
- O sistema não permite clicar em “continuar para pagamento” sem selecionar assentos.

US-RESERVATIONS-010: Atualizar Status de uma Reserva com dados inválidos

Back-end

ID: RESERVATIONS-BE-010
Título: Tentar atualizar o status de uma reserva com valor inválido
Pré-condição: A reserva deve existir, e o usuário deve ser um administrador.
Passos:

1. Fazer uma requisição PUT na rota `/api/v1/reservations/{id}` substituindo `{id}` por um ID válido.
2. Enviar no payload um status inválido (ex.: valores fora do padrão permitido).

Resultado Esperado:

- Retorna status 400 (Bad Request).
- A resposta contém uma mensagem de erro detalhando que o status fornecido é inválido.

Cenários de Testes para Setup

US-SETUP-001: Criar um administrador no ambiente de desenvolvimento

Back-end

ID: SETUP-BE-001
Título: Criar um administrador no ambiente de desenvolvimento
Pré-condição: O ambiente deve estar configurado como "development".
Passos:

1. Fazer uma requisição **POST** na rota `/api/v1/setup/admin`.
2. Enviar no **payload** os seguintes campos: name; email; password.

Resultado Esperado:

- Status: 201 (Created)
- Mensagem de sucesso contendo os dados criados, exceto senha e sua role.

US-SETUP-002: Criar usuários de teste no ambiente de desenvolvimento

Back-end

ID: SETUP-BE-002
Título: Criar usuários de teste no ambiente de desenvolvimento
Pré-condição: O ambiente deve estar configurado como "development".
Passos:

1. Fazer uma requisição **POST** na rota `/api/v1/setup/test-user`.

Resultado Esperado:

- Status: 200 (OK)
- Mensagem de sucesso adequada.

---

8. Matriz de Risco:

Matriz de Risco: Cenários de Testes para Autenticação

ID	Título	Probabilidade	Impacto	Classificação	Justificativa
US-AUTH-001	Registro de usuário com dados válidos	Baixa	Alta	Média	Erros no registro impactam novos usuários, mas possuem baixa ocorrência devido a validações existentes.
US-AUTH-002	Login de usuário	Moderada	Muito Alta	Alta	Login é funcionalidade crítica, falhas impedem o uso do sistema.
US-AUTH-003	Logout de usuário	Baixa	Moderada	Baixa	Falhas no logout têm impacto limitado, já que a sessão expira naturalmente.
US-AUTH-004	Visualizar informações do usuário	Moderada	Moderada	Média	Problemas na visualização afetam a experiência do usuário, mas não

					bloqueiam outras funcionalidades.
US-AUTH-005	Atualizar informações do usuário	Baixa	Moderada	Baixa	Problemas na atualização de dados impactam a experiência, mas têm consequências limitadas.
US-AUTH-006	Registro de usuário com e-mail já existente	Moderada	Alta	Alta	E-mails duplicados afetam a experiência do usuário e geram inconsistências.
US-AUTH-007	Login com credenciais inválidas	Alta	Moderada	Alta	Tentativas falhas frequentes podem frustrar usuários e comprometer a segurança.
US-AUTH-008	Atualizar perfil com informações inválidas	Baixa	Moderada	Média	Informações inválidas não afetam diretamente os dados existentes do sistema.
US-SETUP-001	Criar um administrador no ambiente de desenvolvimento	Alta	Baixa	Média	A funcionalidade é crucial para configuração inicial, mas seu uso restrito ao ambiente de desenvolvimento reduz a probabilidade de falhas graves.
US-SETUP-002	Criar usuários de teste no ambiente de desenvolvimento	Moderada	Baixa	Baixa	A ausência de usuários de teste pode ser contornada, mas afeta a validação de outros cenários, impactando menos o ambiente de desenvolvimento.

Matriz de Risco: Cenários de Testes para Usuários

ID	Título	Probabilidade	Impacto	Classificação	Justificativa
US-USER-001	Listar todos os usuários	Baixa	Muito Alta	Média	Erros na listagem comprometem a gestão administrativa, afetando operações críticas.
US-USER-002	Obter detalhes de um usuário por ID	Moderada	Alta	Alta	Informações incorretas ou ausentes impactam decisões administrativas e operações relacionadas.
US-USER-003	Excluir um usuário por ID	Baixa	Muito Alta	Alta	Falhas na exclusão comprometem a segurança e podem gerar inconsistências no sistema.
US-USER-004	Obter detalhes de usuário com ID inválido	Moderada	Alta	Média	Erros ao buscar dados críticos podem causar impacto em operações administrativas.
US-USER-005	Excluir um usuário com ID inválido	Moderada	Muito Alta	Alta	Exclusão incorreta compromete a integridade dos dados do sistema.
US-USER-006	Atualizar informações de um usuário existente	Moderada	Alta	Alta	Atualizações incorretas podem comprometer a integridade dos dados do usuário, impactando fluxos que dependem de informações corretas.

Matriz de Risco: Cenários de Testes para Filmes

ID	Título	Probabilidade	Impacto	Classificação	Justificativa

US-HOME-001	Página Inicial Atrativa	Baixa	Moderado	Baixa	Problemas de layout podem prejudicar a navegação inicial, mas o impacto direto na funcionalidade é limitado.
US-MOVIE-001	Navegar na Lista de Filmes	Moderada	Alto	Média	Falhas na exibição podem confundir o usuário e dificultar a descoberta de conteúdo.
US-MOVIE-002	Visualizar Detalhes do Filme	Alta	Muito Alto	Alta	Informações incorretas ou ausentes afetam a decisão do usuário de assistir ao filme.
US-MOVIE-003	Criar um novo filme	Moderada	Moderado	Média	Falhas na criação afetam o gerenciamento do catálogo pelo administrador.
US-MOVIE-004	Atualizar um filme existente	Baixa	Moderado	Baixa	Erros na atualização podem atrasar correções de dados, mas o impacto direto é limitado.
US-MOVIE-005	Excluir um filme por ID	Baixa	Moderado	Baixa	Problemas na exclusão podem gerar atrasos no gerenciamento do catálogo, mas o impacto é reversível.
US-MOVIE-006	Criar um filme com informações inválidas	Alta	Alta	Alta	Filmes inválidos geram inconsistências que podem comprometer exibições.
US-MOVIE-007	Atualizar um filme com informações inválidas	Alta	Alta	Alta	Alterações inválidas comprometem a exibição de

					informações corretas ao público.
US-MOVIE-008	Excluir um filme com ID inválido	Moderada	Muito Alta	Alta	Erros na exclusão podem gerar inconsistências nos dados exibidos.

Matriz de Risco: Cenários de Testes para Cinemas

ID	Título	Probabilidade	Impacto	Classificação	Justificativa
US-THEATERS-001	Listar todos os cinemas cadastrados	Baixa	Moderada	Baixa	Problemas na listagem dificultam a experiência, mas não impactam a integridade.
US-THEATERS-002	Obter detalhes de um cinema existente	Moderada	Alta	Média	Dados inconsistentes ou ausentes podem comprometer a tomada de decisões.
US-THEATERS-003	Criar um novo cinema com dados válidos	Moderada	Muito Alta	Alta	Falhas podem gerar retrabalho e impactar a operação administrativa.
US-THEATERS-004	Atualizar informações de um cinema existente	Moderada	Alta	Média	Alterações incorretas afetam a confiabilidade das informações gerenciais.
US-THEATERS-005	Excluir um cinema por ID	Moderada	Muito Alta	Alta	Exclusões incorretas podem causar inconsistências no sistema e afetar exibições.
US-THEATERS-006	Criar um novo cinema com payload inválido ou ausente	Alta	Alta	Alta	Informações inválidas geram falhas críticas na operação do sistema.
US-THEATERS-007	Atualizar um cinema com	Alta	Alta	Alta	Atualizações inválidas podem

	payload inválido ou ausente				comprometer a consistência dos dados do sistema.

Matriz de Risco: Cenários de Testes para Sessões

ID	Título	Probabilidade	Impacto	Classificação	Justificativa
US-SESSIONS-001	Listar todas as sessões disponíveis	Baixa	Moderada	Baixa	Problemas na listagem dificultam a navegação, mas não impactam dados críticos.
US-SESSIONS-002	Obter detalhes de uma sessão específica	Moderada	Alta	Média	Dados ausentes ou incorretos afetam a confiabilidade das informações exibidas.
US-SESSIONS-003	Criar uma nova sessão	Moderada	Muito Alta	Alta	Falhas na criação impactam o planejamento e vendas de ingressos.
US-SESSIONS-004	Atualizar informações de uma sessão existente	Moderada	Alta	Média	Alterações incorretas podem comprometer o fluxo de reservas e exibição de dados.
US-SESSIONS-005	Excluir uma sessão existente	Moderada	Muito Alta	Alta	Exclusões indevidas podem causar falhas graves no sistema de reservas.
US-SESSIONS-006	Criar uma nova sessão com payload inválido ou ausente	Alta	Alta	Alta	Dados inválidos impedem o uso correto do sistema e podem causar retrabalho.
US-SESSIONS-007	Atualizar uma sessão com payload inválido ou ausente	Alta	Alta	Alta	Atualizações inválidas comprometem a integridade e confiabilidade do sistema.
US-SESSIONS-008	Resetar os assentos de uma	Moderada	Muito Alta	Alta	Resetar incorretamente os

	sessão existente para disponíveis				assentos de uma sessão pode levar a inconsistências no sistema, como reservas inválidas ou falhas operacionais.

Matriz de Risco: Cenários de Testes para Reservas

ID	Título	Probabilidade	Impacto	Classificação	Justificativa
US-RESERVATIONS-001	Listar todas as reservas	Moderada	Moderada	Média	Falhas na listagem administrativa podem dificultar a auditoria e o gerenciamento de reservas.
US-RESERVATIONS-002	Obter reservas do usuário atual	Baixa	Moderada	Baixa	Impacta apenas a experiência individual do usuário, mas não afeta operações críticas do sistema.
US-RESERVATIONS-003	Obter detalhes de uma reserva	Moderada	Alta	Média	Falhas podem afetar a experiência do usuário ao tentar acessar informações específicas de sua reserva.
US-RESERVATIONS-004	Criar uma nova reserva	Alta	Muito Alta	Alta	Erros na criação podem impactar diretamente o uso do sistema e frustrar usuários durante o processo de reserva.
US-RESERVATIONS-005	Atualizar status de uma reserva	Moderada	Alta	Alta	Atualizações incorretas afetam a confiabilidade das informações e o gerenciamento de reservas.

US-RESERVATIONS-006	Excluir uma reserva	Moderada	Muito Alta	Alta	Exclusões indevidas podem causar perda de informações importantes ou impactar negativamente os usuários.
US-RESERVATIONS-007	Tentar obter reservas do usuário atual sem autenticação	Alta	Moderada	Alta	Falhas nesse cenário expõem vulnerabilidades de segurança ao lidar com sessões não autenticadas.
US-RESERVATIONS-008	Obter detalhes de uma reserva com ID inválido	Moderada	Moderada	Média	Impacta a navegação do usuário, mas sem consequências críticas para o sistema.
US-RESERVATIONS-009	Criar uma nova reserva com payload inválido ou ausente	Alta	Muito Alta	Alta	Dados inválidos podem comprometer o fluxo de reservas e causar inconsistências no sistema.
US-RESERVATIONS-010	Atualizar status de uma reserva com dados inválidos	Alta	Alta	Alta	Falhas na validação do status comprometem a integridade do gerenciamento de reservas e podem confundir administradores.

---

9. Priorização da Execução dos Cenários de Testes:

A execução dos cenários de testes foi priorizada com base na **Matriz de Risco**, assegurando que funcionalidades mais críticas e de maior impacto sejam validadas primeiro.

Classificação e Critérios de Priorização:

1. **Alta (Prioridade Máxima):**
   - **Definição:** Funcionalidades críticas cuja falha pode bloquear o sistema ou causar grande impacto ao negócio.
   - **Objetivo:** Garantir a estabilidade dos fluxos principais.
2. **Média (Prioridade Intermediária):**

- **Definição:** Funcionalidades importantes que impactam significativamente a experiência do usuário, mas não bloqueiam o sistema.
- **Objetivo:** Validar componentes complementares ao funcionamento básico.

3. **Baixa (Prioridade Mínima):**
   - **Definição:** Funcionalidades de impacto limitado ou uso eventual.
   - **Objetivo:** Assegurar o funcionamento correto após validações críticas e intermediárias.

A Priorização ficou da seguinte maneira:

<details open>
<summary>Alta (Prioridade máxima):</summary>

- **Autenticação:**
  - US-AUTH-002: Login de usuário
  - US-AUTH-006: Registro de usuário com e-mail já existente
  - US-AUTH-007: Login com credenciais inválidas
- **Usuário:**
  - US-USER-002: Obter detalhes de um usuário por ID
  - US-USER-003: Excluir um usuário por ID
  - US-USER-005: Excluir um usuário com ID inválido
  - US-USER-006: Atualizar informações de um usuário existente
- **Filmes:**
  - US-MOVIE-002: Visualizar Detalhes do Filme
  - US-MOVIE-006: Criar um filme com informações inválidas
  - US-MOVIE-007: Atualizar um filme com informações inválidas
  - US-MOVIE-008: Excluir um filme com ID inválido
- **Cinemas:**
  - US-THEATERS-003: Criar um novo cinema com dados válidos
  - US-THEATERS-005: Excluir um cinema por ID
  - US-THEATERS-006: Criar um novo cinema com payload inválido ou ausente
  - US-THEATERS-007: Atualizar um cinema com payload inválido ou ausente
- **Sessões:**
  - US-SESSIONS-003: Criar uma nova sessão
  - US-SESSIONS-005: Excluir uma sessão existente
  - US-SESSIONS-006: Criar uma nova sessão com payload inválido ou ausente
  - US-SESSIONS-007: Atualizar uma sessão com payload inválido ou ausente
  - US-SESSIONS-008: Resetar os assentos de uma sessão existente para disponíveis
- **Reservas:**
  - US-RESERVATIONS-004: Criar uma nova reserva
  - US-RESERVATIONS-005: Atualizar status de uma reserva
  - US-RESERVATIONS-006: Excluir uma reserva
  - US-RESERVATIONS-007: Tentar obter reservas do usuário atual sem autenticação
  - US-RESERVATIONS-009: Criar uma nova reserva com payload inválido ou ausente
  - US-RESERVATIONS-010: Atualizar status de uma reserva com dados inválidos

</details>

<details open>
<summary>Média (Prioridade Intermediária):</summary>

- **Autenticação:**
  - US-AUTH-001: Registro de usuário com dados válidos
</details>

- US-AUTH-004: Visualizar informações do usuário
- **Usuário:**
  - US-USER-001: Listar todos os usuários
  - US-USER-004: Obter detalhes de usuário com ID inválido
- **Filmes:**
  - US-MOVIE-001: Navegar na Lista de Filmes
  - US-MOVIE-003: Criar um novo filme
- **Cinemas:**
  - US-THEATERS-002: Obter detalhes de um cinema existente
  - US-THEATERS-004: Atualizar informações de um cinema existente
- **Sessões:**
  - US-SESSIONS-002: Obter detalhes de uma sessão específica
  - US-SESSIONS-004: Atualizar informações de uma sessão existente
- **Reservas:**
  - US-RESERVATIONS-001: Listar todas as reservas
  - US-RESERVATIONS-003: Obter detalhes de uma reserva
  - US-RESERVATIONS-008: Obter detalhes de uma reserva com ID inválido
- **Setups:**
  - US-SETUP-001: Criar um administrador no ambiente de desenvolvimento

Baixa (Prioridade Mínima):

- **Autenticação:**
  - US-AUTH-003: Logout de usuário
  - US-AUTH-005: Atualizar informações do usuário
  - US-AUTH-008: Atualizar perfil com informações inválidas
- **Filmes:**
  - US-HOME-001: Página Inicial Atrativa
  - US-MOVIE-004: Atualizar um filme existente
  - US-MOVIE-005: Excluir um filme por ID
- **Cinemas:**
  - US-THEATERS-001: Listar todos os cinemas cadastrados
- **Sessões:**
  - US-SESSIONS-001: Listar todas as sessões disponíveis
- **Reservas:**
  - US-RESERVATIONS-002: Obter reservas do usuário atual
- **Setups:**
  - US-SETUP-002: Criar usuários de teste no ambiente de desenvolvimento

---

10. Testes candidatos à automação:

Testes da API para automatizar

Autenticação (Backend)

1. **AUTH-BE-002 (Alta)**: Login de usuário
   Motivo: Fluxo crítico e de alta frequência. A automação garante que o core do sistema (autenticação) funcione após cada deploy, evitando bloqueios para outros testes.

2. **AUTH-BE-006 (Alta)**: Registro com e-mail já existente
   Motivo: Cenário negativo comum que valida regras de negócio (e-mails únicos). Automatizar evita testes manuais repetitivos.
3. **AUTH-BE-007 (Alta)**: Login com credenciais inválidas
   Motivo: Validação de segurança e mensagens de erro. Testes manuais consomem tempo para simular falhas.

Usuários (Backend)

4. **USER-BE-002 (Alta)**: Obter detalhes de usuário por ID
   Motivo: Fluxo essencial para perfis e admin. Automatizar assegura consistência na recuperação de dados sensíveis.
5. **USER-BE-003 (Alta)**: Excluir usuário por ID
   Motivo: Operação crítica (especialmente para admin). Automação previne erros em deleções acidentais.

Filmes (Backend)

6. **MOVIE-BE-002 (Alta)**: Visualizar detalhes do filme
   Motivo: Funcionalidade de alta frequência (usuários e admin). Automação garante que informações-chave (como sinopse) estejam sempre corretas.
7. **MOVIE-BE-006 (Alta)**: Criar filme com dados inválidos
   Motivo: Cenário negativo que valida regras de negócio (ex.: campos obrigatórios). Reduz retrabalho manual.

Cinemas (Backend)

8. **THEATERS-BE-003 (Alta)**: Criar cinema com dados válidos
   Motivo: Fluxo base para reservas/sessões. Automatizar assegura que cinemas novos não quebrem integrações.
9. **THEATERS-BE-006 (Alta)**: Criar cinema com payload inválido
   Motivo: Validação de erros esperados. Testes manuais são tediosos para múltiplos cenários.

Sessões (Backend)

10. **SESSIONS-BE-003 (Alta)**: Criar nova sessão
    Motivo: Impacta diretamente nas reservas. Automação previne conflitos de horários e vincula corretamente a filmes/cinemas.
11. **SESSIONS-BE-006 (Alta)**: Criar sessão com payload inválido
    Motivo: Cenário essencial para evitar sessões corrompidas. Automatizar valida mensagens de erro e status HTTP.

Reservas (Backend)

12. **RESERVATIONS-BE-004 (Alta)**: Criar nova reserva
    Motivo: Fluxo mais crítico do sistema (impacta receita). Automação garante que assentos não sejam duplicados e regras sejam aplicadas.
13. **RESERVATIONS-BE-007 (Alta)**: Tentar obter reservas sem autenticação
    Motivo: Validação de segurança (acesso não autorizado). Evita testes manuais repetitivos.
14. **RESERVATIONS-BE-009 (Alta)**: Criar reserva com payload inválido
    Motivo: Fluxo crítico para o negócio. Automatizar previne reservas em assentos inexistentes ou com dados incompletos, reduzindo inconsistências..

11. Cobertura de Testes:

Cobertura de teste dos endpoints

Utilize um heatmap que marca endpoints testados com cores.

- Verde: Testado.
- Amarelo: Testado parcialmente ou em repetição.
- Vermelho: Não testado.

Endpoint	Prioridade	Cobertura
/auth/register	Altíssima	🟢
/auth/login	Altíssima	🟢
/auth/me	Alta	🟢
/auth/profile	Alta	🟢
/movies	Altíssima	🟢
/movies/{id}	Alta	🟢
/reservations/me	Alta	🟢
/reservations	Altíssima	🟢
/reservations/{id}	Alta	🟢
/sessions	Altíssima	🟢
/sessions/{id}	Alta	🟢
/sessions/{id}/reset-seats	Alta	🟢
/setup/admin	Média	🟢
/setup/test-users	Média	🟢
/theaters	Altíssima	🟢
/theaters/{id}	Alta	🟢
/users	Altíssima	🟢
/users/{id}	Alta	🟢

No Cinema App API temos 18 endpoints, que são os mesmos mostrados na coluna “Endpoint”

Para calcular a cobertura: pelo menos um teste em cada endpoint (18) / quantidade de endpoints na API (18) x 100.

Cobertura: 18/18 x 100 = 100%

Cobertura dos testes manuais da API

Utilizei um heatmap que marca os cenários de teste testados com cores.

- Verde: Testado.
- Amarelo: Testado parcialmente ou em repetição.

- Vermelho: Não testado.

ID	Prioridade	Cobertura
AUTH-BE-001	Média	🟢
AUTH-BE-002	Alta	🟢
AUTH-BE-003	Baixa	🔴
AUTH-BE-004	Média	🟢
AUTH-BE-005	Baixa	🔴
AUTH-BE-006	Alta	🟢
AUTH-BE-007	Alta	🟢
AUTH-BE-008	Baixa	🔴
SETUP-BE-001	Média	🟢
SETUP-BE-002	Baixa	🟢
USER-BE-001	Média	🟢
USER-BE-002	Alta	🟢
USER-BE-003	Alta	🟢
USER-BE-004	Média	🔴
USER-BE-005	Alta	🔴
USER-BE-006	Alta	🟢
MOVIE-BE-001	Média	🟢
MOVIE-BE-002	Alta	🟢
MOVIE-BE-003	Média	🟢
MOVIE-BE-004	Baixa	🟢
MOVIE-BE-005	Baixa	🟢
MOVIE-BE-006	Alta	🟢
MOVIE-BE-007	Alta	🟢
MOVIE-BE-008	Alta	🔴
THEATERS-BE-001	Baixa	🟢
THEATERS-BE-002	Média	🟢
THEATERS-BE-003	Alta	🟢
THEATERS-BE-004	Média	🟢
THEATERS-BE-005	Média	🟢

THEATERS-BE-006	Alta	🟢
THEATERS-BE-007	Alta	🟢
SESSIONS-BE-001	Baixa	🟢
SESSIONS-BE-002	Média	🟢
SESSIONS-BE-003	Alta	🟢
SESSIONS-BE-004	Média	🟢
SESSIONS-BE-005	Alta	🟢
SESSIONS-BE-006	Alta	🟢
SESSIONS-BE-007	Alta	🔴
SESSIONS-BE-008	Alta	🟢
RESERVATIONS-BE-001	Média	🟢
RESERVATIONS-BE-002	Baixa	🟢
RESERVATIONS-BE-003	Média	🟢
RESERVATIONS-BE-004	Alta	🟢
RESERVATIONS-BE-005	Alta	🟢
RESERVATIONS-BE-006	Alta	🟢
RESERVATIONS-BE-007	Alta	🟢
RESERVATIONS-BE-008	Média	🔴
RESERVATIONS-BE-009	Alta	🔴
RESERVATIONS-BE-010	Alta	🟢

Total de cenários de testes: 49.

Passaram 40 testes: 81.63%

Falharam 9 testes: 18.37%

Cobertura dos testes automatizados da API

Utilizei um heatmap que marca os cenários de teste testados com cores.

- Verde: Testado.
- Amarelo: Testado parcialmente ou em repetição.
- Vermelho: Não testado.

ID	Prioridade	Cobertura
AUTH-BE-002	Alta	🟢
AUTH-BE-006	Alta	🟢
AUTH-BE-007	Alta	🟢

MOVIE-BE-002	Alta	🟢
MOVIE-BE-006	Alta	🟢
RESERVATIONS-BE-004	Alta	🟢
RESERVATIONS-BE-007	Alta	🟢
RESERVATIONS-BE-009	Alta	🔴
SESSIONS-BE-003	Alta	🟢
SESSIONS-BE-006	Alta	🟢
THEATERS-BE-003	Alta	🟢
THEATERS-BE-006	Alta	🟢
USER-BE-002	Alta	🟢
USER-BE-003	Alta	🟢

Total de cenários de testes: 14.

Passaram 13 testes: 92.86%

Falharam 1 testes: 7.14%

Cobertura dos testes automatizados do Front-end

Utilizei um heatmap que marca os cenários de teste testados com cores.

- Verde: Testado.
- Amarelo: Testado parcialmente ou em repetição.
- Vermelho: Não testado.

ID	Prioridade	Cobertura
AUTH-FE-001	Média	🟢
AUTH-FE-002	Alta	🟢
AUTH-FE-003	Baixa	🟢
AUTH-FE-004	Média	🟢
AUTH-FE-005	Baixa	🟢
AUTH-FE-006	Alta	🟢
AUTH-FE-007	Alta	🟢
AUTH-FE-008	Média	🟢
SESSIONS-FE-001	Baixa	🟢
SESSIONS-FE-008	Alta	🟢
HOME-FE-001	Baixa	🟢
MOVIE-FE-001	Média	🟢

MOVIE-FE-002	Alta	🟢
RESERVATIONS-FE-002	Baixa	🟢
RESERVATIONS-FE-004	Alta	🟢
RESERVATIONS-FE-007	Alta	🟢
RESERVATIONS-FE-009	Alta	🟢

Total de cenários de testes: 17.

Passaram 17 testes: 100.00%

---

12. Interpretação dos Resultados de Testes Manuais e Automatizados:

Contexto e Ferramentas Utilizadas:
Os testes foram realizados utilizando abordagens complementares: manuais (principalmente para back-end) e automatizados (para back-end e front-end).

- **Testes Manuais (Back-end):** Executados via Postman, com acompanhamento e organização dos resultados no QAlity Plus (Jira).
- **Testes Automatizados (Back-end e Front-end):** Implementados em Robot Framework e executados no VSCode. O foco foi em fluxos essenciais e de alto impacto.
- Os resultados foram consolidados no QAlity, permitindo análise integrada da cobertura e qualidade da aplicação.

Panorama Geral dos Resultados

Com base na consolidação dos testes registrados:

Testes Manuais (API):

- Total de casos avaliados: **49**
- Casos passados: **40**
- Casos falhados: **9**
- **Taxa de sucesso:** 81.63%

Testes Automatizados (API):

- Total de casos avaliados: **14**
- Casos passados: **13**
- Casos falhados: **1**
- **Taxa de sucesso:** 92,86%

Testes Automatizados (Front-end):

- Total de casos avaliados: **17**
- Casos passados: **17**
- Casos falhados: **0**
- **Taxa de sucesso:** 100%

Abordagem	Testes Totais	Passados	Falhados	Taxa de Sucesso

Testes Manuais (Back-end)	49	40	9	81.63%
Testes Automatizados (API)	14	13	1	92,86%
Testes Automatizados (FE)	17	17	0	100%

Destaques Positivos

- **Autenticação e Sessões:** As rotas de login, autenticação e sessões tiveram desempenho consistente nos testes automatizados e manuais. Casos como `AUTH-BE-002`, `AUTH-BE-006`, `SESSIONS-BE-003`, `SESSIONS-BE-006`, `AUTH-FE-002` e `SESSIONS-FE-008` demonstraram robustez funcional.
- **Reserva de Ingressos:** As principais funcionalidades de reservas foram validadas com sucesso (`RESERVATIONS-BE-004`, `RESERVATIONS-BE-007`, `RESERVATIONS-FE-004`, `RESERVATIONS-FE-002`). A criação, visualização e proteção de acesso sem login funcionaram conforme esperado.
- **Filmes e Cinemas:** Testes como `MOVIE-BE-002`, `MOVIE-BE-006`, `MOVIE-FE-002`, `THEATERS-BE-003` e `THEATERS-BE-006` evidenciaram que as funcionalidades de exibição e navegação de conteúdo estão bem implementadas.

Pontos de Melhoria Identificados

Testes Manuais (Back-end):

- Algumas falhas foram recorrentes nas rotas de usuários, como nos casos `USER-BE-004` e `USER-BE-005`, associadas à validação de campos obrigatórios, e-mails duplicados ou ausência de restrições de segurança.
- A rota `MOVIE-BE-008` apresentou falha ao lidar com operações sensíveis, indicando a necessidade de ajustes em regras de negócios ou validações.
- `SESSIONS-BE-007` falhou, sugerindo que casos extremos ou dados inconsistentes durante atualização de sessões ainda não estão bem tratados.

Testes Automatizados (API):

- A falha registrada em `RESERVATIONS-BE-009` pode estar relacionada a tentativa de criar reservas com dados inválidos ou conflito de estados de assento, o que requer validação adicional na lógica de reserva.

Conclusão

A cobertura de testes está bem distribuída entre os módulos mais críticos da aplicação, como autenticação, sessões e reservas. As falhas encontradas foram pontuais e podem ser resolvidas com ajustes nas validações e regras de negócio.

Os testes automatizados demonstraram alta confiabilidade, especialmente no front-end, garantindo que os principais fluxos da aplicação estão funcionando corretamente. Com pequenos aprimoramentos, a aplicação pode alcançar níveis ainda maiores de robustez e estabilidade.