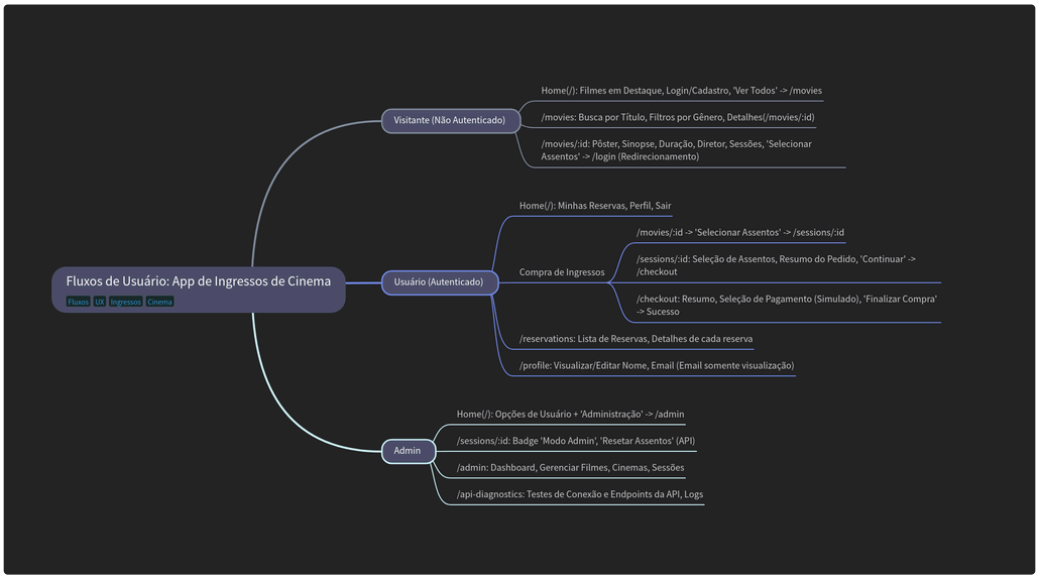
Plano de Testes: Cinema App Frontend

1. Introdução

Este plano de testes detalha a estratégia e a abordagem para validar a aplicação **Cinema App Frontend**. O foco é garantir a qualidade da interface do usuário (UI), da experiência do usuário (UX) e da correta integração com a Cinema App API, assegurando que a aplicação atenda a todos os requisitos funcionais e não funcionais.



(Mapa mental do APP; incluíndo fluxos de navegação organizados para melhor visualização e aproveitamento. Clique aqui para expandir.)

---

2. Objetivo Geral

O objetivo principal é estabelecer um processo de validação sistemático para a aplicação, garantindo que todas as páginas, funcionalidades e fluxos de usuário operem conforme o esperado. Este plano visa identificar falhas funcionais, bugs visuais, problemas de usabilidade e inconsistências de dados, resultando na entrega de um produto final estável, confiável e de alta qualidade.

2.1 Escopo dos Testes

- **Dentro do Escopo:**
  - **Testes Funcionais E2E (End-to-End):** Validação completa dos fluxos de usuário, desde o registro e login até a seleção de assentos e finalização da reserva.
  - **Testes de Interface do Usuário (UI):** Verificação da renderização correta de todos os componentes, consistência visual entre as páginas e conformidade com o design proposto.
  - **Testes de Compatibilidade e Responsividade:** Garantir que a aplicação funcione corretamente nos principais navegadores (Chromium, Firefox, WebKit) e se adapte a diferentes tamanhos de tela (desktop, mobile).
  - **Validação de Integração com API:** Assegurar que o frontend lide adequadamente com as respostas da API, incluindo estados de carregamento, sucesso e erro.
- **Fora do Escopo:**
  - Testes de unidade ou de integração da API do backend (cobertos pelo plano de testes do backend).
  - Testes de performance, carga e estresse do servidor.

- Validação de infraestrutura de backend, como configuração de servidor e banco de dados.

---

3. Análise dos Testes

3.1 Base para Testes

A criação dos cenários será fundamentada nos seguintes artefatos:

- **Histórias de Usuário (`USER-STORIES.md`):** Fornecem os critérios de aceitação e os fluxos de valor para cada papel (Visitante, Usuário, Administrador).
- **Estrutura do Código-Fonte (`repomix-output-juniorschmitz-cinema-challenge-front.xml`):** Análise das páginas, componentes, contextos de estado (`AuthContext`, `AlertContext`) e serviços de API para mapear todas as interações possíveis.
- **Documentação da API (`API-DOCUMENTATION.md`):** Utilizada para entender os contratos de dados e os resultados esperados das chamadas de backend.

3.2 Cobertura de Testes

A estratégia de cobertura visa:

- **100% de cobertura das Histórias de Usuário:** Todos os critérios de aceitação definidos no `USER-STORIES.md` devem ser validados.
- **Cobertura dos Principais Fluxos:** Testar os caminhos críticos da aplicação, como o fluxo de reserva completo, registro e login.
- **Validação de Componentes Reutilizáveis:** Garantir que componentes como `Header`, `Footer` e `Alert` funcionem de forma consistente em todas as páginas onde são utilizados.
- **Testes de Estados da UI:** Validar todos os estados visuais, incluindo telas de carregamento (`loading`), sucesso, erro e estados vazios (ex: "Nenhuma reserva encontrada").

---

4. Técnicas Aplicadas

- **Teste Exploratório:** Navegação livre pela aplicação para identificar bugs de usabilidade e de interface não cobertos por cenários roteirizados.
- **Teste de Fluxo de Usuário:** Execução de jornadas completas que simulam o uso real da aplicação por um usuário final.
- **Teste de Compatibilidade:** Validação da aplicação em diferentes navegadores e resoluções de tela para garantir consistência visual e funcional.
- **Teste de Regressão Visual:** Comparação de screenshots da UI para detectar alterações visuais indesejadas após novas implementações (executado principalmente via automação).

---

5. Cenários de Teste Planejados

Os cenários são agrupados por funcionalidade principal da aplicação.

5.1 Autenticação e Gerenciamento de Perfil

Papel	Cenário	Ação	Resultado Esperado
Visitante	**Registro com sucesso**	Preencher o formulário de registro com dados válidos e submeter.	Usuário é redirecionado para a página de login com uma mensagem de sucesso.
Visitante	**Registro com senhas divergentes**	Preencher o formulário de registro com senhas que não coincidem.	Exibir um alerta de erro "As senhas não coincidem".
Usuário	**Login com credenciais válidas**	Preencher o formulário de login com e-mail e senha corretos.	Usuário é autenticado e redirecionado para a página inicial. O header exibe as opções de usuário logado.
Usuário	**Login com senha incorreta**	Preencher o formulário de login com uma senha errada.	Exibir um alerta de erro "Credenciais inválidas".
Usuário	**Atualização de nome no perfil**	Na página de perfil, alterar o nome e clicar em "Salvar".	Exibir modal de sucesso e o nome no header deve ser atualizado.
Usuário	**Logout da conta**	Clicar no botão "Sair" no header.	Usuário é deslogado e redirecionado para a página de login. O header volta ao estado de visitante.

5.2 Navegação e Fluxo de Reserva

Papel	Cenário	Ação	Resultado Esperado
Visitante	**Visualização de detalhes do filme**	Na página inicial, clicar em um card de filme.	Redirecionado para a página de detalhes do filme (`/movies/:id`), que exibe sinopse e sessões disponíveis.
Usuário	**Seleção de assentos**	Na página de detalhes do filme, selecionar uma sessão e, na tela seguinte, clicar em assentos disponíveis.	Os assentos clicados mudam para o estado "Selecionado" e o resumo do pedido é atualizado.
Usuário	**Tentar selecionar assento ocupado**	Na tela de seleção, clicar em um assento com status "Ocupado".	O assento não pode ser selecionado, mantendo seu estado original.
Usuário	**Finalização da compra**	Após selecionar os assentos, clicar em "Continuar para Pagamento", escolher um método e finalizar a compra.	Exibir a tela de confirmação da reserva com todos os detalhes do pedido.
Usuário	**Visualização de reservas**	Acessar a página "Minhas Reservas" pelo menu.	A página exibe um card com os detalhes da reserva recém-criada.
Admin	**Acesso ao painel de Admin**	Fazer login como administrador.	O link "Administração" deve aparecer no header.

6. Matriz de Risco

Risco	Probabilidade	Impacto	Mitigação
Falha na comunicação com a API	Média	Alto	Testar o comportamento da UI em todos os pontos de integração (login, busca de filmes, etc.) quando a API está offline ou retorna erro 500. A aplicação deve exibir alertas de erro amigáveis e não quebrar.
Inconsistência de estado de autenticação	Média	Médio	Validar se, após o login/logout, todos os componentes (como o `Header`) reagem corretamente e se as rotas protegidas ficam inacessíveis/acessíveis conforme o esperado.
Bugs de layout em telas responsivas	Alta	Médio	Executar testes visuais em diferentes viewports (desktop, tablet, mobile) para garantir que elementos como grids de filmes e a tela de seleção de assentos não quebrem.
Falha ao passar dados entre páginas	Média	Alto	No fluxo de reserva, garantir que os dados da sessão e os assentos selecionados são passados corretamente da página de `SeatSelection` para `Checkout` via `location.state`.

---

7. Testes Candidatos à Automação

Para garantir a estabilidade e agilizar os ciclos de teste, os seguintes cenários são fortes candidatos à automação, pois cobrem os fluxos mais críticos e repetitivos da aplicação.

7.1 Cenários Automatizados com Robot Framework

- **Autenticação:**
  - Validar o fluxo completo de registro de um novo usuário.
  - Testar o login com credenciais válidas e verificar se o `Header` é atualizado.
  - Testar o login com credenciais inválidas e validar a exibição do alerta de erro.
- **Fluxo de Reserva (E2E):**
  - Automatizar a jornada: Login → Selecionar filme → Selecionar sessão → Escolher assentos → Validar resumo → (Simular) Checkout → Verificar se a reserva aparece em "Minhas Reservas".
- **Controle de Acesso:**
  - Tentar acessar a página de perfil (`/profile`) sem estar logado e validar o redirecionamento para `/login`.
  - Fazer login como usuário comum e garantir que o link de "Administração" não esteja visível no `Header`.

---

8. Estrutura de Testes

A automação com Robot Framework será organizada da seguinte forma para garantir manutenibilidade e reuso:

```
/cinema-frontend-tests
|
|--- tests/
|   |--- test_authentication.robot
|   |--- test_booking_flow.robot
|   |--- test_profile_and_reservations.robot
|
|--- resources/
|   |--- common_keywords.robot
|   |--- page_objects/
|   |   |--- home_page.robot
|   |   |--- login_page.robot
|   |   |--- movie_detail_page.robot
|
|--- variables/
|   |--- test_data.py
|   |--- environment_variables.robot
```

- `tests/`: Contém os casos de teste de alto nível, escritos em Gherkin.
- `resources/`: Agrupa keywords reutilizáveis e Page Objects para abstrair a interação com os elementos da UI.
- `variables/`: Centraliza dados de teste, como credenciais de usuários e URLs de ambiente.

---

9. Ferramentas de Teste

- **Robot Framework:** Ferramenta principal para automação dos testes E2E.
  - **Browser Library:** Para interação e automação do navegador.
  - **RequestsLibrary:** Para chamadas de API auxiliares (ex: criar dados de teste antes de uma validação de UI).
- **QAlity for Jira:** Para documentar os casos de teste manuais, registrar evidências e controlar os resultados.
- **Jira:** Para registrar e rastrear os bugs identificados, vinculando-os aos cenários de teste correspondentes.
- **Navegadores (Chrome, Firefox, WebKit):** Para execução dos testes manuais exploratórios e de compatibilidade.

---

10. Conclusão

Este plano de testes fornece uma abordagem estruturada e completa para validar a aplicação Cinema App Frontend. A combinação de testes manuais exploratórios com uma suíte de automação robusta para os fluxos críticos garantirá a detecção precoce de falhas e a entrega de um produto de alta qualidade, funcional e com excelente usabilidade.