**🎬 Plano de Teste – Cinema App**

Qualidade de Software • Testes Funcionais e Automatizados

**📑 Sumário**

1. Apresentação

2. Objetivo

3. Escopo

4. Ambiente de Teste

5. Análise

6. Cenários de Teste

7. Técnicas de Teste Aplicadas

8. Casos de Teste

9. Matriz de Risco

10. Cobertura de Testes

**Plano de Teste Cinema App**

Apresentação

Este plano de testes tem como objetivo validar funcionalidades críticas da aplicação **Cinema App**, garantindo que o sistema responda corretamente às solicitações dos usuários. Através da execução de testes funcionais, buscamos assegurar a qualidade da entrega, a estabilidade do sistema e a confiabilidade dos principais fluxos de uso, como consulta de filmes, seleção de sessões e compra de ingressos.

Objetivo

Assegurar que todos os endpoints funcionem corretamente de acordo com as especificações da aplicação, com foco em:

* Validação de entradas (dados válidos e inválidos)
* Segurança e controle de acesso (token Bearer)

Escopo

O escopo deste plano de testes abrange a validação dos principais **endpoints** e funcionalidades da aplicação *Cinema App*, com foco em assegurar que as operações essenciais funcionem corretamente. Serão testadas as seguintes funcionalidades:

* Cadastrar usuário
* Autenticação (login)
* Atualização de dados do usuário
* Listar filmes em cartaz
* Buscar de filmes por critérios específicos
* Realizar reservas de ingressos
* Consulta de reservas de ingressos
* Não permitir reserva duplicada

Não testaremos funcionalidades administrativas, performance ou integração com meios de pagamento nesta fase.

Ambiente de teste

Atualmente, os testes da aplicação **Cinema App** serão executados em ambiente local de desenvolvimento, com a seguinte configuração mínima:

* **Servidor local** rodando via localhost
* **URL base**: http://localhost:3000 (frontend) e http://localhost:3001/api (exemplo para API REST)
* **Sistema Operacional**: Windows 10 ou superior
* **Banco de Dados**: MongoDB local ou em nuvem (ex: MongoDB Atlas)
* **Ferramentas utilizadas**: Postman (para testes manuais), Node.js, navegador Google Chrome

Análise

Para garantir a qualidade e confiabilidade da aplicação **Cinema App**, será adotada uma abordagem mista de testes, combinando diferentes técnicas e ferramentas

**1. Testes Manuais (via Postman)**

* Os principais endpoints da API REST serão testados manualmente no **Postman**.
* Serão validadas respostas esperadas, status codes, mensagens de erro, e fluxos completos como cadastro, login, listagem e reservas.
* Casos de teste serão documentados para referência e reexecução.

**2. Testes Automatizados (via Robot Framework)**

* Scripts de automação serão desenvolvidos em **Robot Framework**, visando garantir repetibilidade, agilidade e regressão constante dos testes da API.
* Utilização de arquivos resource e fixtures para organizar melhor os testes.
* Execução local inicialmente, com possibilidade futura de integração em pipeline CI/CD.

**3. Testes Exploratórios**

* Serão realizados testes exploratórios manuais para identificar falhas inesperadas, comportamentos anômalos ou pontos cegos não cobertos por casos de teste pré-definidos.
* O foco será em descobrir erros por meio da interação direta com o sistema, utilizando a experiência e intuição do testador.

**Mapa mental da aplicação**

Diagrama, Esquemático

O conteúdo gerado por IA pode estar incorreto.

**Cenários de Teste**

| **ID** | **Funcionalidade** | **Cenário** | **Resultado Esperado** |
| --- | --- | --- | --- |
| CT-01 | Cadastro | Usuário se cadastra com dados válidos | Cadastro concluído com sucesso |
| CT-02 | Cadastro | Tentativa de cadastro com email já existente | Mensagem de erro informando email duplicado |
| CT-03 | Login | Login com email e senha corretos | Usuário autenticado com sucesso |
| CT-04 | Login | Login com senha incorreta | Acesso negado com mensagem de erro |
| CT-05 | Perfil do Usuário | Alterar nome com sucesso | Nome atualizado corretamente |
| CT-06 | Perfil do Usuário | Tentar alterar nome para campo em branco | Exibição de mensagem de erro |
| CT-07 | Filmes | Listar todos os filmes | Lista de filmes em cartaz exibida corretamente |
| CT-08 | Filmes | Buscar filme por nome existente | Filme correspondente exibido |
| CT-09 | Filmes | Buscar filme inexistente | Mensagem informando ausência de resultados |
| CT-10 | Reserva | Realizar reserva com sucesso | Reserva confirmada |
| CT-11 | Reserva | Tentar reservar assento ocupado | Erro informando que o assento já está reservado |
| CT-12 | Reserva | Buscar reservas com usuário autenticado | Lista de reservas do usuário exibida corretamente |
| CT-13 | Reserva | Buscar reservas sem estar logado | Acesso negado; redirecionamento para login |

**Técnicas de teste aplicadas**

| **Técnica de Teste** | **Onde foi aplicada** | **Objetivo principal** |
| --- | --- | --- |
| Partição de Equivalência | Cadastro, login, criação de filmes, sessões, reservas | Separar entradas válidas e inválidas para reduzir a quantidade de testes |
| Análise de Valor Limite | Campos como senha, email, entradas com tamanho mínimo/máximo | Verificar comportamento nos limites aceitáveis de entrada |
| Teste de Autenticação/Autorização | Endpoints protegidos como /me, criação/edição de filmes e salas | Validar acesso com token, sem token e com permissões de admin |
| Teste de Fluxo (Caminho) | Cadastro → login → ações do usuário autenticado | Garantir que os passos da navegação estão corretos e encadeados |
| Teste de Transição de Estado | Atualização e exclusão de filmes, sessões, reservas | Verificar se o sistema muda corretamente de estado após ações |
| Teste Negativo | Erros esperados (400, 401, 403, 404, 409) | Validar como o sistema reage a entradas inválidas ou ações proibidas |
| Teste de Regressão | Após alterações ou adições de funcionalidades | Garantir que funcionalidades antigas continuam funcionando |
| Teste de CRUD | Em todos os recursos: filmes, salas, sessões, reservas | Validar operações de Criar, Ler, Atualizar e Deletar |
| Teste de Código de Status HTTP | Em todos os endpoints | Verificar se os status retornados estão corretos (200, 400 etc.) |

Casos de Teste

| **ID** | **Funcionalidade** | **Título do Caso de Teste** | **Dados de Entrada** | **Resultado Esperado** |
| --- | --- | --- | --- | --- |
| TC-001 | Cadastro | Cadastro com dados válidos | Nome, email, senha válidos | Cadastro realizado com sucesso |
| TC-002 | Cadastro | Cadastro com email já existente | Email já cadastrado | Erro: "Email já cadastrado" |
| TC-003 | Login | Login com credenciais válidas | Email e senha corretos | Autenticação bem-sucedida com token |
| TC-004 | Login | Login com senha incorreta | Email válido, senha incorreta | Erro: "Credenciais inválidas" |
| TC-005 | Perfil do Usuário | Alterar nome com sucesso | Novo nome | Nome atualizado corretamente |
| TC-006 | Perfil do Usuário | Tentar alterar nome para campo em branco | Nome vazio | Erro: "Nome é obrigatório" |
| TC-007 | Filmes | Listar todos os filmes | Nenhum | Lista de filmes em cartaz retornada |
| TC-008 | Filmes | Buscar filme por nome existente | Nome do filme válido (ex: "Batman") | Filme correspondente retornado |
| TC-009 | Filmes | Buscar filme inexistente | Nome de filme inexistente | Lista vazia ou mensagem: "Nenhum filme encontrado" |
| TC-010 | Reservas | Realizar reserva com sucesso | Filme, horário, assento disponíveis | Reserva realizada com sucesso |
| TC-011 | Reservas | Tentar reservar assento já ocupado | Mesmo assento e horário já reservado | Erro: "Assento indisponível" |
| TC-012 | Reservas | Buscar reservas com usuário autenticado | Token de autenticação | Lista de reservas do usuário exibida |
| TC-013 | Reservas | Buscar reservas sem estar logado | Nenhum token | Erro: "Autenticação necessária" |

**📌 Matriz de Risco**

| **ID** | **Caso de Teste** | **Nível de Risco** | **Prioridade** | **Ação Recomendada** |
| --- | --- | --- | --- | --- |
| TC-011 | Reserva de assento já ocupado | 🔴 Crítico | Alta | Validar duplicidade de assento |
| TC-010 | Realizar reserva com sucesso | 🟧 Alto | Alta | Validar o fluxo completo |
| TC-007 | Listar todos os filmes | 🟧 Alto | Alta | Garantir exibição correta |
| TC-013 | Buscar reservas sem login | 🟧 Alto | Alta | Exigir token e retorno 401 |
| TC-002 | Cadastro com email já existente | 🟧 Alto | Média | Validar erro de duplicidade |
| TC-004 | Login com senha incorreta | 🟨 Médio | Média | Garantir mensagem sem vazar info |
| TC-006 | Nome em branco no perfil | 🟨 Médio | Média | Validar campo obrigatório |
| TC-008 | Buscar filme por nome existente | 🟨 Médio | Média | Validar retorno parcial |
| TC-009 | Buscar filme inexistente | 🟩 Baixo | Baixa | Garantir retorno vazio amigável |

**✅ Cobertura de Testes**

| **🎯 Funcionalidade** | **🧪 Casos Testados** | **✅ Cobertura Visual** | **📈 %** |
| --- | --- | --- | --- |
| 🧍 Cadastro | TC-001, TC-002 | ██████████████████ | 100% |
| 🔐 Login | TC-003, TC-004 | ██████████████████ | 100% |
| 👤 Perfil do Usuário | TC-005, TC-006 | ██████████████████ | 100% |
| 🎞️ Listagem de Filmes | TC-007 | ████████░░░░░░░░░░ | 50% |
| 🔎 Busca de Filmes | TC-008, TC-009 | ██████████████████ | 100% |
| 🎟️ Reservas | TC-010, TC-011 | ██████████████████ | 100% |
| 📂 Consulta de Reservas | TC-012, TC-013 | ██████████████████ | 100% |