## Plano de Testes - Cinema

## 1. Identificação do Plano de Testes 🕖

• Nome do projeto: Cinema

• Responsável: Cassia Yumi Iwamoto Basso

• Data de criação: 23/06/2025

Data	Revisão
26/06/2025	Adicionado número dos casos de testes
27/06/2025	Revisado Mapa mental

### 2. Apresentação 🖉

Este documento apresenta o planejamento de testes da API Cinema, que é uma RESTful API para reservas de tickets de cinema ticket criada com Node.js, Express, and MongoDB e uma interface Web responsiva.

Serão realizados testes tanto no Backend quanto no Frontend:

- No backend serão realizados testes manuais e automatizados através do Postman, com foco nos endpoints, autenticação JWT, consistência dos dados e validações de entradas, de acordo com as histórias de usuário apresentadas e documentação da API disponível em <a href="Swagger-backend">Swagger-backend</a>.
- No frontend serão realizados testes manuais e automatizados utilizando o Robot Framework, com a finalidade de garantir que os fluxos funcionais principais (login, registro, seleção de assentos, e outras funcionalidades) atendam às necessidades dos usuários.

## 3. Escopo dos Testes 🕖

O escopo inclui as funcionalidades descritas nas histórias de usuário, em especial:

✓ Autenticação (registro, login, logout, perfil). Ø

✓ Navegação e visualização de filmes. Ø

✓ Horários de sessão. Ø

✓ Reservas de assentos e checkout. Ø

✓ Consulta das reservas realizadas. Ø

✓ Navegabilidade da interface. ②

X Fora do escopo: Testes de integração externa de pagamento, Envio de e-mails, Teste de carga, Funcionalidades administrativas não acessíveis.

#### 4. Funcionalidades a serem testadas 🖉

Registro de usuário (US-AUTH-001)

- ✓ Login/logout (US-AUTH-002 e US-AUTH-003)
- Gerenciamento de perfil (US-AUTH-004)
- ✓ Home e exibição de filmes (US-HOME-001, US-MOVIE-001)
- ✓ Detalhes de filmes (US-MOVIE-002)
- Sessões (US-SESSION-001)
- ☑Seleção e reserva de assentos (US-RESERVE-001, US-RESERVE-002)
- ✓ Histórico de reservas (US-RESERVE-003)
- ✓ Navegação geral da aplicação (US-NAV-001)

### 5. Estratégia de Testes 🕖

Backend (Postman): @

Serão aplicados os seguintes testes:

- Testes de métodos: GET, POST, PUT e DELETE.
- Testes de autenticação com JWT.
- Teste de validação de dados (campos obrigatórios, tipos).
- Testes de fluxo de reserva (selecionar assentos, checkout, bloquear assentos reservados).

Em relação à autenticação e controle e acesso serão validados o perfil admin para rotas protegidas, verificação de restrições de acesso sem token e testes com token expirado ou ausente.

Frontend (Robot Framework): @

Serão aplicados os seguintes testes:

- Testes manuais.
- Validação de navegação entre telas.
- Verificação visual de mensagens, feedbacks e responsividade.
- Fluxos de usuário: registro, login, perfil, reserva, logout.
- Teste de interface para módulos "Filmes em cartaz", "Detalhes do filme", "Minhas reservas".

Será realizado a análise da cobertura com base no Path Coverage, utilizando-se o seguinte critério:

- Total de URIs disponíveis: X
- URIs testadas: Y
- Cobertura: Path Coverage= (Y / X) \* 100%

Será realizada a análise dos riscos, com foco em:

- Falhas em login, seleção de filmes ou processo de reserva.
- Utilização de matriz de risco para priorização dos testes.

Serão analisados os testes candidatos à automação com base em repetição, criticidade e estabilidade da funcionalidade.

Os dados de testes utilizados poderão ser usados da seguinte forma:

- Usuários com diferentes permissões (visitante, usuário autenticado, administrador)
- Filmes com diferentes classificações, datas e gêneros
- Tokens válidos e inválidos

• Carrinhos com múltiplos produtos ou sessões simultâneas

## 6. Critérios de Aceitação dos Testes ⊘

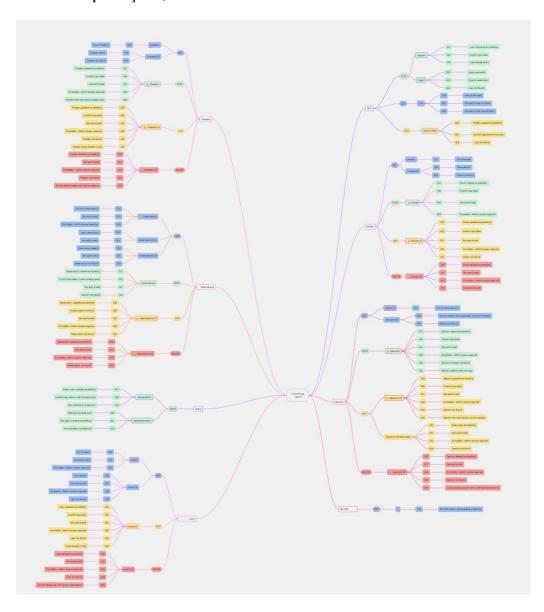
- Todos os endpoints devem retornar códigos HTTP apropriados;
- Validações de dados devem estar presentes (ex: formato de e-mail, campos obrigatórios);
- Interface deve exibir mensagens claras em casos de erro ou sucesso;
- Tokens

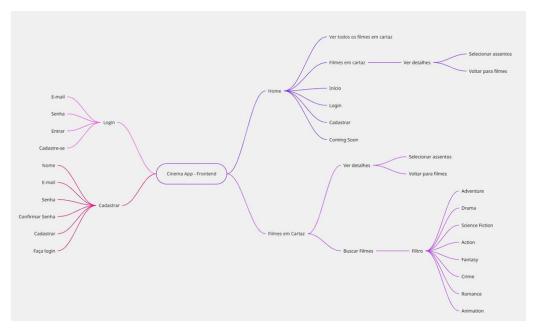
## 7. Técnicas aplicadas 🕖

Com base na documentação da API e nas users stories disponibilizadas, foram selecionadas as seguintes técnicas de teste:

- Partição de Equivalência: testes de entradas válidas e inválidas;
- Análise de Valor Limite: verificação de limites mínimos e máximos das entradas;
- Teste de Valores Nulos e Vazios
- Testes baseados em regras de negócio: ex: reserva só com login, bloqueio de assentos;
- Testes exploratórios: visando encontrar comportamento não previstos nos requisitos ou na documentação;
- Técnica de transição de estados: especialmente quanto ao token, autenticação e navegação da sessão.

# 8. Mapa Mental da aplicação 🖉





Mapa mental - Frontend

## 9. Cenários de Testes Planejados 🕖

Com base nas users stories apresentadas e na documentação da API ServeRest foram planejados os seguintes cenários de testes:

## **1** US-AUTH-001 *⊘*

#### Cenário de Teste US-AUTH-001

Descrição: Realizar o cadastro de um novo usuário com sucesso no Cinema App.

#### Pré-condições:

- 1. O usuário ainda **não está cadastrado** na base de dados.
- 2. O formulário de registro está disponível e acessível.
- 3. O endpoint /auth/register (método POST) está ativo e funcional.

### Condições:

- Campos obrigatórios: nome, email, senha;
- O sistema deve validar o formato do e-mail;
- O sistema não deve permitir registro com e-mails duplicados;
- O sistema deve validar a complexidade mínima da senha (por exemplo: mínimo 6 caracteres);
- Em caso de sucesso, o sistema deve criar o usuário e redirecionar para a página de login (no frontend);
- O backend deve retornar HTTP 201 Created em caso de sucesso;
- O backend deve retornar HTTP 400 Bad Request em caso de erros de validação.

#### Passo a passo: 🖉

1. Acessar a tela de cadastro no frontend;

- 2. Preencher os campos obrigatórios com dados válidos e enviar o formulário;
- 3. Verificar se a resposta da API é 201 Created;
- 4. Validar que o usuário foi de fato persistido no banco;
- 5. Verificar se a interface redireciona para a tela de login;
- 6. Realizar novas tentativas com dados inválidos e verificar o comportamento do sistema.

## Ø Casos de Teste Associados: Ø

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Component	Versio n
CIN-1 Registro com dados válidos	Cadastro com nome, e- mail e senha corretos	Preencher todos os campos e enviar formulário	Nome: Ana SilvaEmail : ana@test e.comSen ha: Abc12345	HTTP 201 Created; redireciona mento para login	Frontend/A PI	1.0
CIN-2 Registro com e- mail já cadastrad o	Testar bloqueio de e-mails duplicado s	Enviar formulário com e- mail já existente	Email: ana@test e.com	HTTP 400 com mensagem "E-mail já cadastrado"	Frontend/A PI	1.0
CIN-3 Registro com e- mail inválido	Verificar validação de formato de e-mail	Enviar e- mail inválido no formulário	Email: øt este.co m	HTTP 400 com mensagem "E-mail inválido"	Frontend	1.0
CIN-4 Registro com senha fraca	Verificar se senha com menos de 6 caractere s é rejeitada	Enviar senha com poucos caractere s	Senha: 123	HTTP 400 com mensagem "Senha muito curta"	Frontend/A PI	1.0

CIN-5 Registro com nome vazio	Validação de campo obrigatóri o (nome)	Enviar formulário sem preencher o nome	Nome: ""	HTTP 400 com mensagem "Campo obrigatório"	Frontend/A PI	1.0
CIN-6 Registro com campos em branco	Testar envio de todos os campos vazios	Submeter formulário em branco	Nome, Email e Senha vazios	HTTP 400 com múltiplas mensagens de erro	Frontend	1.0
CIN-8 Resposta da API	Validar se resposta contém ID e e-mail após cadastro	Observar resposta JSON	-	JSON com propriedade s id, email	API	1.0

## **2** US-AUTH-002 *⊘*

#### Cenário de Teste US-AUTH-002

Descrição: Realizar autenticação com sucesso no Cinema App.

Pré-condição: Usuário previamente cadastrado e validado no sistema com nome, e-mail e senha corretos.

#### Condições:

- Visitantes (usuários não cadastrados) não devem conseguir autenticar;
- Usuários cadastrados com senha incorreta não devem conseguir autenticar;
- Falhas de autenticação devem retornar HTTP 401 Unauthorized;
- Usuários existentes com senha correta devem ser autenticados com sucesso;
- A autenticação bem-sucedida deve retornar um token do tipo Bearer;

- Enviar uma requisição HTTP POST para o endpoint /auth/login com email e password;
- 2. Verificar se o **status da resposta** é 200 OK (para sucesso) ou 401 Unauthorized (para falha);
- 3. Validar a presença de um token JWT no corpo da resposta em caso de sucesso;
- 4. Confirmar que, após o login, o usuário é redirecionado à página inicial e vê o menu personalizado.

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen t	Versio n
CIN-10 Login com credenciai s válidas	Verifica login com e-mail e senha corretos	Acessar login → Inserir credenciai s → Clicar em Entrar	E-mail: ana@test e.com Senha: Teste123	Usuário é redirecionad o para a página inicial	Frontend/A PI	1.0
CIN-11 Login com senha incorreta	Verifica resposta para senha inválida	Inserir e- mail correto com senha errada e tentar logar	E-mail: ana@test e.com Senha: Errada123	Mensagem "Invalid email or password" é exibida	Frontend/A PI	1.0
CIN-12 Login com campo vazio	Validação de formulário de login com campos vazios	Deixar campos em branco e submeter o formulário	E-mail:"" Senha:""	Validação impede envio; mensagens de campo obrigatório são exibidas	Frontend/A PI	1.0

#### ■US -AUTH-003 @

#### Cenário de Teste US-AUTH-003

Descrição: Encerrar a sessão de um usuário autenticado no Cinema App.

#### Pré-condições:

- 1. Usuário deve estar logado/autenticado com um token JWT válido armazenado no navegador (localStorage ou sessionStorage);
- 2. O menu de navegação com a opção de logout deve estar visível.

#### Condições:

- O botão ou link de logout deve estar disponível após o login;
- Ao acionar o logout, o token deve ser removido do localStorage;
- Após o logout, o **usuário não deve conseguir acessar rotas protegidas**;
- O sistema pode redirecionar o usuário para a página de login ou home pública;

- Nenhuma requisição futura deve conter o token anterior;
- A sessão deve ser considerada encerrada para efeitos de autorização.

- 1. Realizar login com e-mail e senha válidos;
- 2. Aguardar carregamento da interface autenticada;
- 3. Clicar na opção **"Logout"** no menu superior ou lateral;
- 4. Verificar se o token JWT foi removido do navegador;
- 5. Tentar acessar uma rota protegida, como /perfil ou /minhas-reservas;
- 6. Confirmar que o sistema impede o acesso (ex: redireciona para login ou mostra mensagem de acesso negado).
- Ø Casos de Teste Associados: 
   Ø

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen	Versio n
CIN-17 Logout visível após login	Verificar se botão de logout aparece após autenticaç ão	Fazer login e observar o menu	-	Opção de logout visível no menu	Frontend	1.0
CIN-18 Logout remove token	Verificar se token é removido ao sair	Logar → Clicar em Logout	-	localStora ge não deve conter o token	Frontend	1.0
CIN-19 Acesso a rota protegida após logout	Verificar se usuário é impedido de acessar rotas privadas	Logout → Acessar /auth/p rofile	-	Redirecioname nto para login ou mensagem de acesso negado	Frontend/A PI	1.0

CIN-21	Verificar se menu	Logout → Verificar	-	Menu padrão para visitantes	Frontend	1.0
Navegaçã o limpa após	personaliz ado não	opções no menu		é exibido		
logout	aparece mais					

#### **4** US-AUTH-004 €

#### Cenário de Teste US-AUTH-004

Descrição: Permitir que o usuário logado visualize e edite suas informações de perfil no Cinema App.

#### Pré-condições:

- 1. O usuário deve estar logado e autenticado;
- 2. A rota /perfil ou interface de perfil deve estar acessível apenas para usuários autenticados;
- 3. O backend deve prover um endpoint (ex: GET /users/profile e PUT /users/profile) que retorne e atualize os dados do usuário.

#### Condições:

- A página de perfil deve exibir: nome, e-mail e tipo de conta (usuário ou admin);
- O nome completo do usuário pode ser editado;
- O e-mail e o tipo de conta não podem ser editados;
- O sistema deve indicar visualmente os campos alterados (ex: mudança de cor ou borda);
- Após salvar, deve ser exibida uma mensagem de sucesso;
- O frontend deve atualizar os dados exibidos após alteração bem-sucedida;
- A edição de perfil deve estar disponível apenas para o próprio usuário autenticado.

- 1. Realizar login com um usuário válido;
- 2. Navegar até a página de **Perfil** através do menu;
- 3. Verificar se os dados (nome, e-mail, tipo) são exibidos corretamente;
- 4. Alterar o nome e clicar em Salvar;
- 5. Verificar se a resposta da API (PUT /users/profile) é 200 OK;
- 6. Verificar se o nome foi atualizado corretamente no frontend e backend;
- 7. Verificar se a mensagem de sucesso é exibida;
- 8. Tentar alterar e-mail ou tipo e verificar se os campos estão desabilitados ou bloqueados.

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Component	Versio n
CIN-23 Visualizar dados do perfil	Verificar se os dados do usuário são exibidos corretame nte	Logar → Acessar página de perfil	Usuário autenticad o	Nome, e- mail e tipo de conta visíveis	Frontend/AP I	1.0
CIN-24 Editar nome com sucesso	Alterar nome e salvar alterações	Alterar nome no campo e clicar em salvar	Novo nome: João Silva	Nome atualizado; resposta HTTP 200; mensagem de sucesso exibida	Frontend/AP	1.0
CIN-25 Campo e- mail não editável	Verificar que o campo de e-mail está desabilita do	Tentar editar o e- mail diretamen te na interface	-	Campo bloqueado ou em modo somente leitura	Frontend	1.0
CIN-26 Confirmaç ão visual de alterações	Verificar se campos alterados mudam visualmen te	Modificar o campo "nome"	-	Campo com destaque (ex: borda azul, ícone de modificação )	Frontend	1.0
CIN-29 Atualizaçã o inválida (nome vazio)	Tentar salvar nome em branco	Apagar conteúdo do campo "nome" e salvar	Nome: ""	Validação bloqueia ação; mensagem "Campo obrigatório"	Frontend/AP	1.0

CIN-30	Testar	Tentar	Token de	HTTP 403	API	1.0
Acesso ao	acesso com token	editar perfil de	usuário X para ID	Forbidden ou		
outro usuário	de outro usuário	outro usuário via token alternado	de usuário Y	redireciona mento		

## **5**US-HOME-001 €

#### Cenário de Teste US-HOME-001

Descrição: Verificar se a página inicial apresenta uma visão geral clara e atrativa para o usuário, com foco em conteúdo visual, navegação e responsividade.

#### Pré-condições:

- 1. A aplicação está acessível via navegador;
- 2. O frontend está carregando corretamente os dados da API (filmes em cartaz, usuário autenticado ou não);
- 3. Backend disponibiliza os filmes em cartaz.

#### Condições:

- Um **banner principal** deve ser exibido com informações do cinema;
- A seção "Filmes em Cartaz" deve mostrar pôsteres em tamanho adequado e com qualidade visual;
- O layout deve ser responsivo, adaptando-se a diferentes tamanhos de tela (mobile, tablet, desktop);
- A página deve conter links de navegação rápidos para áreas como login, reservas, perfil, etc.;
- O cabeçalho da página deve permitir navegação principal clara;
- **Usuários logados** devem ver opções personalizadas como "Minhas Reservas", "Perfil" ou "Logout".

- 1. Acessar a home page da aplicação ( / );
- 2. Observar a presença do banner institucional;
- 3. Verificar a seção de **filmes em cartaz** com pôsteres organizados;
- 4. Redimensionar a tela ou abrir em dispositivo móvel para verificar responsividade;
- 5. Observar se os **links rápidos** para áreas principais estão visíveis e funcionais;
- 6. Fazer login como usuário e verificar se o menu muda dinamicamente.



Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen	Versio n
CIN-31 Exibir banner principal	Verificar se banner institucion al aparece na home	Acessar página inicial	-	Banner visível com texto/imagem informativa sobre o cinema	Frontend	1.0
CIN-32 Exibir seção Filmes em Cartaz	Verificar seção de filmes com pôsteres visuais	Rolagem até a seção de filmes	-	Grid com pôsteres, títulos e informações básicas	Frontend/A PI	1.0
CIN-33 Layout responsivo	Validar adaptaçã o da home em telas diferentes	Redimens ionar janela ou abrir em celular	-	Elementos se ajustam; layout legível e funcional	Frontend	1.0
CIN-34 Exibir links de navegação principais	Verificar links como login, reservas, perfil	Observar o cabeçalh o	-	Links rápidos presentes e funcionais	Frontend	1.0
CIN-37 Navegaçã o funcional a partir da home	Verificar se os links da home levam para as seções corretas	Clicar nos botões/lin ks da página inicial	-	Redirecioname ntos corretos e sem erro 404	Frontend	1.0
CIN-38  Carregame nto da home sem autenticaç ão	Validar exibição para visitantes	Abrir página inicial sem login	-	Home carregada normalmente com conteúdo público	Frontend/A PI	1.0

#### Cenário de Teste US-MOVIE-001

Descrição: Permitir que visitantes ou usuários autenticados visualizem a lista de filmes disponíveis, com informações claras e navegação acessível.

#### Pré-condições:

- 1. O backend deve estar populado com filmes disponíveis na rota (ex: GET /movies);
- 2. A home ou a página dedicada a "Filmes em Cartaz" deve estar acessível;

#### Condições:

- A lista de filmes deve ser exibida em layout em grid com boa organização visual;
- Cada card deve apresentar um pôster de alta qualidade, o título, a classificação indicativa e os gêneros do filme;
- Deve incluir duração e data de lançamento nos cards ou em tooltips/popups;
- O layout da lista de filmes deve ser responsivo (adaptável a diferentes tamanhos de tela);
- Cada card deve ser clicável, levando o usuário à página de detalhes do respectivo filme.

- Acessar a página de filmes em exibição (na home ou em rota dedicada /filmes);
- 2. Observar a disposição dos cards em grid;
- 3. Verificar se os cards contêm as informações mínimas esperadas;
- 4. Clicar sobre um card e verificar se o redirecionamento para a página de detalhes ocorre;
- 5. Testar o comportamento da página em resoluções diferentes.



Test Name	Descriptio n	Test Step	Test Data	Expected Result	Component	Versio n
CIN-39 Exibição de grid de filmes	Verificar se os filmes aparecem organizad os em grid	Acessar página de filmes	-	Lista de filmes exibida com espaçamento e alinhamento adequados	Frontend	1.0

CIN-40 Exibir pôster grande e visível	Validar que cada card exibe imagem de qualidade	Observar cards na página	-	Imagens visíveis, proporcionais e sem quebra	Frontend	1.0
CIN-41 Exibir título, classifica ção, gêneros	Verificar campos básicos de cada filme	Analisar informaçõ es visíveis nos cards	-	Título, classificação indicativa e gêneros corretamente exibidos	Frontend	1.0
CIN-42 Exibir duração e data de lançament o	Verificar informaçõ es adicionais de cada filme	Observar detalhes nos cards	-	Duração e data visíveis no card ou acessíveis	Frontend	1.0
CIN-43 Card de filme é clicável	Verificar se clique no card leva à página de detalhes	Clicar em qualquer card	-	Navegação para página de detalhes do filme corresponden te	Frontend	1.0

## **☑**US-MOVIE-002 *②*

#### Cenário de Teste US-MOVIE-002

Descrição: Permitir que visitantes ou usuários autenticados vejam informações detalhadas sobre um filme, com opção de seguir para a reserva.

#### Pré-condições:

- 1. O backend deve disponibilizar os dados detalhados do filme via rota (ex: GET /movies/:id);
- 2. Deve haver sessões cadastradas no sistema para o filme em questão;
- 3. A página de detalhes deve estar acessível a partir da lista de filmes ou via rota direta.

#### Condições:

- A página deve exibir: sinopse, elenco, diretor, duração e data de lançamento;
- O pôster do filme deve estar visível de forma destacada;

- Devem ser exibidos os horários de sessão disponíveis para aquele filme;
- O usuário deve conseguir clicar em um horário para iniciar a reserva de assentos;
- A página deve ser acessível tanto por visitantes quanto por usuários autenticados.

## Passo a passo: 🖉

- 1. Acessar a lista de filmes e clicar em um card;
- 2. Ser redirecionado para a **página de detalhes** do filme selecionado;
- 3. Verificar a presença dos campos: sinopse, elenco, diretor, duração, lançamento;
- 4. Confirmar que o pôster está visível;
- 5. Verificar a lista de **sessões disponíveis** (com datas e horários);
- 6. Clicar em uma sessão e verificar se é redirecionado para a seleção de assentos.

## 

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen	Versio n
CIN-109 Exibir detalhes do filme	Verificar campos principais do filme	Acessar página de detalhes	ID de filme válido	Sinopse, elenco, diretor, duração e lançamento visíveis	Frontend/A PI	1.0
CIN-48 Exibir horários de sessão	Verificar sessões disponívei s para o filme	Observar seção "Horários"	-	Lista com datas, horários, sala e disponibilidade	Frontend/A PI	1.0
CIN-49 Acessar página de reserva pela sessão	Validar link do horário para reservar	Clicar em um horário	-	Redirecioname nto para seleção de assentos	Frontend	1.0
CIN-50 Acesso com ID de filme inválido	Testar erro ao acessar um filme inexistent e	Acessar /filmes /abc123 invalid o	ID inválido	Página de erro ou mensagem "Filme não encontrado"	API	1.0

#### Cenário de Teste US-SESSION-001

Descrição: Permitir que visitantes e usuários autenticados visualizem os horários disponíveis de exibição para um filme específico e naveguem até a seleção de assentos.

#### Pré-condições:

- 1. O backend deve disponibilizar as sessões cadastradas com associação ao filme via rota (ex: GET /sessions?movieId=...);
- 2. O usuário deve acessar a página de detalhes do filme;
- 3. Deve haver pelo menos uma sessão futura cadastrada no banco.

#### Condições:

- Os horários de sessão devem exibir: data, hora, teatro (sala) e disponibilidade de assentos;
- A exibição deve ser clara, com layout que facilite comparação de opções;
- Os horários devem estar ordenados por data e hora;
- Cada item de horário deve ser clicável e redirecionar para a tela de seleção de assentos;
- A funcionalidade deve estar disponível tanto para visitantes quanto para usuários autenticados (reserva exige login posteriormente).

- 1. Acessar a página de detalhes de um filme com sessões disponíveis;
- 2. Observar a lista de **horários de sessão** exibida na tela;
- 3. Verificar se os campos de data, hora, sala e disponibilidade estão visíveis;
- 4. Clicar em um dos horários disponíveis;
- 5. Validar o redirecionamento para a página de seleção de assentos.



Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen	Versio n
CIN-53 Sessões ordenada s cronologic amente	Validar ordenaçã o das sessões por data e hora	Observar ordem da lista	-	Sessões ordenadas do mais próximo para o mais distante	Frontend	1.0

CIN-54 Visualizaç ão de sala e disponibili dade	Verificar se sala e número de assentos disponívei s são exibidos	Observar campos da sessão	-	Campo "Sala" e "Disponibilidad e" visíveis	Frontend	1.0
CIN-55 Link para seleção de assentos	Verificar redirecion amento ao clicar em um horário	Clicar em um botão de horário	-	Usuário é levado à tela de seleção de assentos	Frontend	1.0
CIN-56 Sessão sem disponibili dade	Testar se sessões lotadas são desativad as ou indicadas corretame nte	Visualizar sessão com todos assentos ocupados	Sessão lotada	Sessão marcada como indisponível ou botão desativado	Frontend/A PI	1.0

### ■US-RESERVE-001 ②

#### Cenário de Teste US-RESERVE-001

Descrição: Permitir que o usuário logado visualize o mapa de assentos e selecione as posições disponíveis para reservar em uma sessão específica.

### Pré-condições:

- 1. O usuário deve estar logado com token válido;
- 2. Deve existir uma **sessão com assentos disponíveis** no sistema;
- 3. O frontend deve exibir um layout interativo do mapa de assentos;
- 4. Backend deve disponibilizar as informações de assentos via rota (ex: GET /sessions/:id/seats).

#### Condições:

- O sistema deve exibir o layout do teatro com os assentos organizados visualmente;
- Assentos devem estar **codificados por cor**, indicando:
  - o Disponível

- Selecionado
- Reservado
- O usuário pode selecionar múltiplos assentos disponíveis;
- Assentos já reservados não podem ser selecionados;
- O sistema deve calcular e exibir o **subtotal da compra** com base nos assentos selecionados;
- O botão de "Próximo" ou "Continuar" deve ser ativado somente após seleção válida.

## Passo a passo: 🖉

- 1. Fazer login com um usuário válido;
- 2. Acessar uma sessão disponível via horários de exibição;
- 3. Verificar o **layout gráfico** dos assentos do teatro;
- 4. Observar a **legenda de cores** para status dos assentos;
- 5. Selecionar múltiplos assentos disponíveis;
- 6. Verificar se os assentos são destacados e se o subtotal aparece;
- 7. Tentar clicar em assentos já reservados e verificar o bloqueio;
- 8. Confirmar que é possível prosseguir apenas com seleção válida.

## 

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Component	Versio n
CIN-58 Visualizar layout de assentos	Verificar exibição do mapa gráfico da sala	Acessar página de reserva após escolher sessão	Sessão com layout ativo	Grid de assentos visível, organizado por fileira	Frontend	1.0
CIN-59 Legenda de cores	Validar se há legenda clara para status dos assentos	Observar layout de assentos	-	Cores distintas para disponível, selecionado e reservado	Frontend	1.0
CIN-60 Selecionar assento disponível	Verificar se assento pode ser	Clicar em assento disponível	-	Assento muda de cor para "selecionado"	Frontend	1.0

	seleciona do					
CIN-61 Selecionar múltiplos assentos	Verificar seleção de mais de um assento	Selecionar 2 ou mais assentos	-	Todos os assentos ficam destacados e adicionados ao subtotal	Frontend	1.0
CIN-62 Impedir seleção de assento reservado	Testar clique em assentos já ocupados	Clicar em assento marcado como "reservad o"	Assento indispon ível	Nada acontece ou mensagem "assento já reservado"	Frontend	1.0
CIN-63  Calcular subtotal dinamica mente	Verificar atualizaçã o do valor total da reserva	Selecionar e desseleci onar assentos	Assento: R\$ 30,00	Subtotal atualizado corretamente com base na seleção	Frontend	1.0
CIN-64 Acesso sem login	Testar tentativa de acesso à tela de reservas sem estar autentica do	Acessar /reserv a/:id sem estar logado	-	Redirecioname nto para login ou erro HTTP 401	Frontend/A PI	1.0

## **1 0** - US-RESERVE-002 *𝔞*

#### Cenário de Teste US-RESERVE-002

Descrição: Permitir que o usuário finalize a compra de ingressos após selecionar os assentos, processando o pagamento e confirmando a reserva.

## Pré-condições:

- 1. O usuário deve estar logado;
- 2. O usuário deve ter **selecionado um ou mais assentos disponíveis**;
- 3. A aplicação deve redirecionar o usuário para a página de **checkout** após a seleção;
- 4. O backend deve aceitar simulação de pagamento e registrar reserva.

## Condições:

- A tela de checkout deve mostrar um resumo claro dos assentos selecionados (fileira, número, quantidade);
- O valor total da compra deve ser calculado e exibido corretamente;
- O usuário deve poder escolher o método de pagamento (cartão, débito, PIX, transferência);
- Ao confirmar a compra, o sistema deve **processar o pagamento (simulado)**;
- O sistema deve exibir confirmação visual de sucesso (mensagem ou tela de sucesso);
- Os assentos selecionados devem ser marcados como ocupados na base de dados e na próxima visualização da sessão.

## Passo a passo: 🖉

- 1. Fazer login e selecionar assentos em uma sessão;
- 2. Ser redirecionado para a página de checkout;
- 3. Verificar os assentos selecionados e o valor total exibido;
- 4. Escolher um método de pagamento e confirmar a compra;
- 5. Verificar se o sistema retorna status de sucesso;
- 6. Visualizar mensagem de confirmação na interface;
- 7. Voltar à página da sessão e verificar que os assentos escolhidos agora estão ocupados.

## 🔗 Casos de Teste Associados: ∅

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen	Versio n
CIN-66 Redirecion amento após seleção	Verificar se o usuário vai para o checkout após escolher assentos	Selecionar assentos → clicar em "Próximo"	2 assentos	Usuário vai para /checkout com os dados carregados	Frontend	1.0
CIN-67 Exibir resumo dos assentos	Validar apresenta ção das poltronas seleciona das	Observar o resumo do pedido na página de checkout	Assento s A2, A3	Resumo exibe corretamente fileiras, quantidade e numeração	Frontend	1.0

CIN-68 Exibir valor total	Verificar cálculo total com base nos assentos	Selecionar múltiplos assentos	R\$ 20,00 cada	Total exibido = soma dos preços	Frontend	1.0
CIN-69 Selecionar método de pagament o	Validar interface de seleção de pagament o	Escolher cartão de crédito ou PIX	PIX	Campo de método ativo e selecionável	Frontend	1.0
CIN-70 Confirmar reserva	Verificar envio e resposta de confirmaç ão da reserva	Confirmar	Assento s + método	Requisição enviada com sucesso; resposta HTTP 201 ou equivalente	Frontend/A PI	1.0
CIN-71 Confirmaç ão visual após pagament o	Verificar mensage m de sucesso após concluir reserva	Observar tela ou modal pós- checkout	-	Mensagem como "Reserva confirmada com sucesso"	Frontend	1.0
CIN-72 Atualizaçã o do status dos assentos	Verificar que os assentos são marcados como ocupados após o checkout	Voltar à mesma sessão após compra	-	Assentos aparecem como "reservados"	Frontend/A PI	1.0

# ■1 US-RESERVE-003 @

## Cenário de Teste US-RESERVE-003

Descrição: Permitir que o usuário autenticado visualize seu histórico de reservas com detalhes claros e organizados.

#### Pré-condições:

- 1. O usuário deve estar **logado** com sessão ativa (token válido);
- 2. O backend deve expor um endpoint para listar reservas por usuário (ex: GET /reservas/minhas);
- 3. O frontend deve disponibilizar a página Minhas Reservas, acessível via menu.

#### Condições:

- O menu de navegação deve conter um item "Minhas Reservas", visível apenas para usuários logados;
- A página de reservas deve apresentar as informações em cards visuais;
- Cada card deve conter:
  - Título do filme
  - o Data e horário da sessão
  - Cinema (sala ou unidade)
  - Lista de assentos reservados
  - Status da reserva (ex: confirmada, pendente, cancelada)
  - Método de pagamento
  - Pôster do filme
- Os status das reservas devem estar destacados com indicadores visuais (cor, ícone, selo, etc.);
- A página deve ser separada da tela de perfil e não acessível por visitantes.

#### Passo a passo: 🖉

- 1. Fazer login como usuário que possui reservas ativas;
- 2. Acessar o menu e clicar em "Minhas Reservas";
- 3. Verificar se a rota leva para a página de histórico de reservas;
- 4. Observar se as reservas são exibidas corretamente em cards;
- 5. Validar presença e legibilidade de todas as informações relevantes;
- 6. Observar se o status está visualmente destacado (por cor ou selo);
- 7. Confirmar que pôsteres dos filmes são exibidos;
- 8. Validar o comportamento da tela para usuários sem reservas.

## 🔗 Casos de Teste Associados: ∅

Test Name	Descriptio n	Test Step	Test Data	Expected Result	Componen	Versio n
CIN-75	Verificar presença	Fazer login e	-	Link "Minhas Reservas"	Frontend	1.0

Link "Minhas Reservas" visível	do link após login	abrir o menu		visível no menu		
CIN-76 Acesso à rota de reservas	Validar que a página é carregada ao clicar no link	Clicar em "Minhas Reservas"	-	Redirecionam ento para /reservas	Frontend	1.0
CIN-81 Exibir cards de reservas	Verificar se reservas são exibidas em cards	Acessar página com histórico	Reservas registrad as	Cards exibidos com organização visual clara	Frontend	1.0
CIN-77 Exibir detalhes da reserva	Validar campos como filme, data, hora, cinema, assentos e pagament o	Observar os dados de cada card	-	Todos os dados visíveis e bem formatados	Frontend	1.0
CIN-78 Exibir pôster do filme	Verificar se imagem do pôster é carregada em cada reserva	Observar visual de cada card	-	Imagem do pôster exibida sem erro	Frontend	1.0
CIN-79 Status da reserva com destaque visual	Verificar destaque de status	Observar selo, cor ou ícone em cada card	Confirma da, Pendente	Indicação visual clara (ex: verde para confirmada, cinza para cancelada)	Frontend	1.0

CIN-80	Verificar	Fazer	-	Mensagem	Frontend	1.0
Usuário sem reservas	comporta mento quando não há reservas	login com usuário sem reservas		"Você ainda não fez nenhuma reserva" exibida	The state of the s	
	registrada s					

## 1 2 -US-NAV-001 €

#### Cenário de Teste US-NAV-001

Descrição: Garantir que o usuário, autenticado ou visitante, consiga navegar facilmente entre as páginas do Cinema App, com clareza visual, feedback e responsividade.

#### Pré-condições:

- 1. O sistema está acessível via navegador com interface carregada;
- 2. O usuário pode estar logado ou não;
- 3. As páginas da aplicação estão implementadas (ex: Home, Filmes, Perfil, Reservas, Login, etc.).

#### Condições:

- O cabeçalho (header) deve estar presente em todas as páginas visíveis;
- O menu de navegação deve conter os principais links (ex: Home, Filmes, Login/Logout, Minhas Reservas, Perfil);
- O menu deve ser **responsivo**, adaptando-se a dispositivos móveis;
- Usuários logados devem ver opções adicionais como "Minhas Reservas" e "Perfil";
- Indicadores de caminho atual devem ser visíveis em páginas internas;
- Deve haver feedback visual claro de qual página está ativa (ex: link destacado);
- Em telas com fluxo mais profundo, deve haver **link de retorno** para a página anterior.

- 1. Acessar a aplicação como visitante e verificar o cabeçalho em diferentes páginas;
- 2. Fazer login como usuário e verificar se o menu é atualizado com opções extras;
- 3. Verificar o comportamento do menu em dispositivos móveis (versão responsiva);
- 4. Acessar páginas internas (ex: detalhes do filme, checkout, perfil) e verificar breadcrumbs ou caminhos;
- 5. Observar qual item do menu está realçado como ativo;

- 6. Navegar para uma página interna e voltar usando o link de retorno (back link ou botão);
- 7. Testar acessibilidade visual (ex: contraste e clareza dos elementos ativos).

🔗 Casos de Teste Associados: ∅

Test	Description	Test Step	Test	Expected	Compone	Versio
Name			Data	Result	nt	n
CIN-82 Cabeçalh o presente em todas as páginas	Verificar visibilidade do header ao navegar entre rotas	Acessar várias rotas (home, login, perfil)	-	Cabeçalho visível e funcional em todas as páginas	Frontend	1.0
CIN-83 Menu com links principais	Validar presença de links como Home, Filmes, Login, etc.	Observar o menu em qualquer página	-	Links principais visíveis no cabeçalho	Frontend	1.0
CIN-88 Feedback visual da página ativa	Verificar destaque do link da página atual no menu	Navegar entre páginas	-	Link da página atual com cor diferente, sublinhado ou outra indicação visual	Frontend	1.0
CIN-86 Link de retorno para página anterior	Testar presença de botão/link de "voltar" em páginas internas	Acessar detalhes do filme ou checkout	-	Link funcional para voltar à página anterior (ex: lista de filmes)	Frontend	1.0
CIN-85 Menu personaliz ado para	Verificar se opções como "Minhas Reservas"	Fazer login e abrir o menu	-	Opções extras visíveis no menu após login	Frontend	1.0

usuário	e "Perfil"			
logado	aparecem			

## 10. Matriz de Risco 🕝

Para a elaboração da matriz de risco, foram utilizadas as seguintes escalas:

- Probabilidade de falha: Alta/Média/Baixa.
- Impacto da falha na experiência do usuário ou funcionamento do sistema: Alto/Médio/Baixo.
- Riscos Associados
- Prioridade de teste: combinação de probabilidade, impacto e riscos

Rota	Probabilida de	Impact o	Riscos Associados	Prioridad e
POST /users (Registro)	Média	Alta	<ul> <li>Validação incorreta de dados</li> <li>Permitir e-mails duplicados</li> <li>Falha no cadastro e retorno de erro</li> </ul>	Alta
POST /login (Autenticaçã o)	Alta	Alta	<ul> <li>Credenciais inválidas aceitas</li> <li>Falha na geração ou armazenamento do token JWT</li> <li>Ataques de brute force</li> </ul>	Alta
GET /users/profil e	Média	Média	<ul> <li>Exposição de dados incorretos</li> <li>Falha no carregamento do perfil</li> <li>Problemas de autorização</li> </ul>	Média
PUT /users/profil e	Média	Média	<ul> <li>Atualização incorreta dos dados</li> <li>Falta de validação</li> <li>Não confirmação de sucesso ao usuário</li> </ul>	Média
GET / (Página inicial)	Baixa	Média	<ul><li>Falha na exibição do banner ou filmes em cartaz</li><li>Problemas de responsividade</li></ul>	Média
GET /movies	Média	Alta	<ul><li>Lista incompleta ou incorreta</li><li>Imagens com baixa qualidade</li><li>Falha ao carregar lista</li></ul>	Alta

GET /movies/:id	Média	Alta	<ul> <li>Dados incompletos ou errados</li> <li>Falha ao carregar horários de sessões</li> <li>Navegação quebrada</li> </ul>	Alta
GET /sessions? movieId=	Média	Alta	<ul> <li>Horários desatualizados ou incorretos</li> <li>Sessões com disponibilidade errada</li> </ul>	Alta
GET /sessions/:id /seats	Alta	Alta	<ul> <li>Assentos incorretamente marcados como disponíveis ou ocupados</li> <li>Layout confuso ou inacessível</li> </ul>	Alta
POST /checkout ou /reservas	Alta	Crítica	<ul> <li>Pagamento não processado ou duplicado</li> <li>Overbooking de assentos</li> <li>Falha na confirmação da reserva</li> </ul>	Alta
GET /reservas/mi nhas	Baixa	Média	<ul> <li>Histórico incompleto ou errado</li> <li>Falha na exibição dos status ou dos métodos de pagamento</li> </ul>	Média
GET /nav/header (frontend)	Baixa	Baixa	<ul><li>Links quebrados</li><li>Problemas de responsividade</li><li>Feedback visual confuso</li></ul>	Baixa

# 11. Priorização da Execução dos Cenários de Teste $_{\mathscr{Q}}$

Considerando a realização dos testes com o Robot Framework e com base na documentação da API e, principalmente, as users stories apresentadas, bem como o fluxo da API e a matriz de risco apresentada no item anterior, os cenários de teste foram priorizados da seguinte forma:

Prioridad e	Rotas / Funcionalidades	Justificativa	Recomendações
Alta P	POST /login     (Autenticação)	Risco crítico: bloqueia acesso à aplicação, falha impacta todos os usuários	Executar primeiro, testar fluxos JWT e validação
	POST /users     (Registro)	Essencial para aquisição de novos usuários e dados corretos	Testes automáticos e de borda rigorosos

	• GET /movies (Lista de filmes)	Sem lista de filmes, usuário não pode explorar nem reservar	Testes funcionais
	• GET /movies/:id (Detalhes do filme)	Essencial para tomada de decisão de compra	Testar detalhamento, links para sessões
	<ul> <li>GET /sessions?     movieId=         (Horários)</li> </ul>	Horários corretos são vitais para reserva	Testar atualização e disponibilidade
	• GET /sessions/:id/seat s (Assentos)	Assentos incorretos impactam diretamente a experiência e reserva	Validar layout, cores e seleção
	<ul> <li>POST /checkout (Finalização de reserva)</li> </ul>	Crítico para garantir compra sem erros, evita overbooking	Testar métodos de pagamento e confirmações
Média €	<ul><li>GET /users/profile (Perfil)</li></ul>	Mantém dados do usuário atualizados e corretos	Testar leitura e controle de acesso
	<ul><li>PUT /users/profile (Editar perfil)</li></ul>	Importante para atualização de dados pessoais	Validar mensagens e persistência
	<ul> <li>GET         /reservas/minhas         (Histórico de         reservas)</li> </ul>	Facilita acompanhamento, mas não bloqueia outras funções	Testes funcionais e visual
	GET / (Página inicial)	Impacta percepção da aplicação, mas não impede funcionamento principal	Testes visuais e responsividade
Baixa P	GET /nav/header     (Navegação e menu)	Afeta usabilidade e navegação, porém com impacto menor	Testes de interface e responsividade

## 12. Cobertura de testes 🔗

A cobertura de testes foi medida considerando o Path Coverage, que "verifica a cobertura da suíte de testes de acordo com os endpoints que a API possui" (CREMA, Nayara. Como verificar a cobertura de testes de APIs REST, Medium ).

## ${\sl r}$ URIs disponíveis na API Cinema - Backend ${\ensuremath{\mathscr{O}}}$

Módulo	Método HTTP	URI	Descrição
Auth	POST	/register	Registro de novo usuário
		/login	Autenticação/login
	GET	/me	Visualiza perfil do usuário autenticado
	PUT	/auth/profile	Atualiza perfil do usuário
Movies	GET	/movies	Lista filmes
		/movies/:id	Traz informações sobre o filme selecionado
	POST	/movies	Cria um novo filme
	PUT	/movies/:id	Atualiza informações de um filme
	DELETE	/movies/:id	Exclui filme
Sessions	GET	/sessions	Lista sessões
		/sessions/:id	Lista informações da sessão
	POST	/sessions	Cria uma sessão
	PUT	/sessions/:id	Atualiza informações sobre sessão
		/sessions/:id/reset- seats	Disponibiliza todos os assentos da sessão
	DELETE	/sessions/:id	Exclui sessão
Theaters	GET	/theaters	Lista os cinemas
		/theaters/:id	Lista informações sobre um cinema
	POST	/theaters	Cria um cinema
	PUT	/theaters/:id	Atualiza informações sobre um cinema
	DELETE	/theaters/:id	Exclui um cinema

Reservatio ns	GET	/reservations	Lista as reservas
		/reservations/me	Lista as reservas do usuário autenticado
		/reservations/:id	Lista uma reserva
	POST	/reservations	Cria uma nova reserva
	PUT	/reservations/:id	Atualiza uma reserva
	DELETE	/reservations/:id	Cancela uma reserva
Setup	POST	/setup/admin	Cria um administrador teste
		/setup/test-users	Cria um usuário teste
Users	GET	/users	Lista os usuários
		/users/:id	Detalhes de um usuário específico
	PUT	/users/:id	Atualiza dados de um usuário
	DELETE	/users/:id	Exclui um usuário
API Info	GET	1	Informações da API

Total de URIs disponíveis: 33

# ✓ URIs testadas 𝒪

URI	Status
POST/register	Testada
POST/login	Testada
GET/me	Testada
PUT/auth/profile	Testada
GET/movies	Testada
GET/movies/:id	Testada
POST/movies	Testada
PUT/movies/:id	Testada
DELETE/movies/:id	Testada
GET/sessions	Testada

GET/sessions/:id	Testada
POST/sessions	Testada
PUT/sessions/:id	Testada
PUT/sessions/:id/reset-seats	Testada
DELETE/sessions/:id	Testada
GET/theaters	Testada
GET/theaters/:id	Testada
POST/theaters	Testada
PUT/theaters/:id	Testada
DELETE/theaters/:id	Testada
GET/reservations	Testada
GET/reservations/me	Testada
GET/reservations/:id	Testada
POST/reservations	Testada
PUT/reservations/:id	Testada
DELETE/reservations/:id	Testada
POST/setup/admin	Testada
POST/setup/test-users	Testada
GET/users	Testada
GET/users/:id	Testada
PUT/users/:id	Testada
DELETE/users/:id	Testada
1	Testada

Total de URIs testadas: 33

📊 Resultado 🖉

Path Coverage = (33 / 33) × 100 = 100% ₽

Funcionalidade / Área	Casos de Teste Relacionados	Status (Coberto / Faltando)
Autenticação e Autorização	CIN-1 a CIN-6, CIN-8, CIN-10 a CIN-12, CIN-17 a CIN-21	Coberto
Registro usuário	CIN-1, CIN-2, CIN-3, CIN-4, CIN-5, CIN-6, CIN-8	Coberto
Login / Logout	CIN-10, CIN-11, CIN-12, CIN-17, CIN-18, CIN-19, CIN-21	Coberto
Acesso a rotas protegidas	CIN-19	Coberto
Menu personalizado	CIN-85, CIN-21	Coberto
Navegação e Interface Geral	CIN-31 a CIN-38, CIN-82 a CIN-88	Coberto
Header e menu geral	CIN-31, CIN-32, CIN-33, CIN-34, CIN-37, CIN-38, CIN-82, CIN-83, CIN-85, CIN-86, CIN-88	Coberto
Home e Filmes	CIN-39 a CIN-43, CIN-109, CIN-48 a CIN-50	Coberto
• Grid e cards de filmes	CIN-39 a CIN-43	Coberto
Detalhes do filme	CIN-109, CIN-48, CIN-49, CIN-50	Coberto
Sessões e Reservas	CIN-53 a CIN-56, CIN-58 a CIN-64, CIN-66 a CIN-72, CIN-75 a CIN-80	Coberto
Listagem e ordenação de sessões	CIN-53, CIN-54, CIN-55, CIN-56	Coberto
Visualização e seleção de assentos	CIN-58, CIN-59, CIN-60, CIN-61, CIN-62, CIN-63	Coberto
Checkout e confirmação	CIN-66, CIN-67, CIN-68, CIN-69, CIN-70, CIN-71, CIN-72	Coberto
Reservas exibidas	CIN-75, CIN-76, CIN-77, CIN-78, CIN-79, CIN-80	Coberto
Comportamento para sem reservas	CIN-80	Coberto
Perfil do Usuário	CIN-23 a CIN-26, CIN-29, CIN-30	Coberto
• Visualização e edição	CIN-23, CIN-24, CIN-25, CIN-26	Coberto
Validações e restrições	CIN-29, CIN-30	Coberto

## 14. Testes candidatos a automação 🔗

Os testes candidatos à automação podem ser agrupados com base em diferentes tipos de validação e execução de requisições consecutivas.

Considerando o resultado dos testes manuais realizados no Postman, foram selecionados os casos de testes candidatos a automação:

Caso de Teste	Descrição	Justificativa para Automação
CT-Login-Valido	Testar login com credenciais válidas	Fluxo crítico, repetitivo, fundamental para acesso
CT-Login-Invalido	Testar login com credenciais inválidas	Regressão comum, previne acesso indevido
CT-Registro-Usuario	Testar cadastro de usuário com dados válidos	Processo obrigatório, validação automática facilita
CT-Registro- EmailDuplicado	Testar cadastro com email já cadastrado	Verificação de duplicidade, validação crítica
CT-Logout	Testar logout e invalidação do token	Teste essencial de segurança e sessão
CT-Visualizar-Perfil	Visualizar dados do perfil do usuário	Teste frequente e base para personalização
CT-Editar-Perfil	Atualizar dados do perfil e confirmar sucesso	Fluxo repetido, evita falhas na atualização
CT-Listar-Filmes	Listar filmes disponíveis com layout correto	Alta frequência e impacto visual
CT-Detalhes-Filme	Visualizar detalhes completos do filme	Facilita decisão do usuário, importante para UX
CT-Listar-Horarios	Consultar horários de sessões para um filme	Informação dinâmica e essencial para reservas
CT-Visualizar-Assentos	Visualizar layout e status dos assentos da sessão	Base para seleção e bloqueio, evita erros de disponibilidade
CT-Selecionar-Assentos	Selecionar múltiplos assentos disponíveis	Fluxo importante para compra e UX
CT-Checkout-Reserva	Finalizar compra e confirmar reserva	Processo crítico e sensível, garante fluxo completo

CT-Listar-Reservas-	Visualizar histórico de	Função usada
Usuario	reservas do usuário	frequentemente, importante
		para confiança