

Planejamento de teste API challenger final

Plano de Testes: API Cinema App

Objetivo do Documento

Este documento detalha a estratégia, o escopo e os casos de teste para a validação da API RESTful do sistema "Cinema App". O objetivo é garantir que todas as funcionalidades implementadas atendam aos requisitos de negócio (descritos nas Histórias de Usuário), mantenham a integridade dos dados, tratem erros de forma adequada e apliquem as regras de segurança (autenticação e autorização) corretamente.

Escopo dos Testes

Em Escopo

- Testes de Funcionalidade:** Validação de todos os endpoints da API para as entidades:
 - Autenticação (Auth)
 - Usuários (Users)
 - Filmes (Movies)
 - Salas de Cinema (Theaters)
 - Sessões (Sessions)
 - Reservas (Reservations)
- Testes de Integração:** Verificação da interação entre os diferentes módulos (ex: criar uma reserva deve atualizar o status dos assentos em uma sessão).
- Testes de Validação e Tratamento de Erros:** Garantir que a API retorne códigos de status HTTP e mensagens de erro apropriadas para entradas inválidas, recursos não encontrados, etc.
- Testes de Segurança (Autorização):** Assegurar que os endpoints protegidos só possam ser acessados por usuários autenticados e com a permissão (role) correta (`user` vs. `admin`).

Fora de Escopo

- Testes de Interface de Usuário (Frontend):** O foco é exclusivamente na API backend.
- Testes de Performance e Carga:** Testes de estresse para verificar o comportamento da API sob um grande volume de requisições (poderia ser uma fase futura).
- Testes de Usabilidade:** Relacionado à experiência do usuário no frontend.

Níveis de Teste

- Testes de Unidade:** Foco em funções isoladas (ex: `generateToken`, métodos do Mongoose nos Models como `matchPassword`, ou um controller específico com suas dependências mockadas).
- Testes de Integração:** Foco na interação entre controllers, serviços e banco de dados.
- Testes de API (Ponta a Ponta - E2E):** Foco em simular o uso real da API através de requisições HTTP, validando o fluxo completo. *Este plano focará principalmente neste nível.*

Estratégia de Dados de Teste

- Dados Iniciais (Seeding):** O banco de dados de teste deve ser populado antes da execução dos testes usando os scripts fornecidos (`npm run seed`). Isso garante um estado inicial conhecido.
- Usuários de Teste:** Serão utilizados dois perfis principais:
 - Usuário Admin:** `admin@example.com` | `password123`

- **Usuário Comum:** user@example.com | password123

- **Dados Dinâmicos:** Para testes destrutivos (como `DELETE`) ou de criação, os dados serão criados e, se possível, removidos no próprio escopo do teste para garantir a independência e repetibilidade dos testes.

i Ferramentas Recomendadas

- **Cliente de API (Testes Manuais/Exploratórios):** Postman ou Insomnia.
- **Framework de Testes Automatizados: Jest com Supertest** (uma combinação popular para testar APIs em Node.js).
- **CI/CD:** GitHub Actions para rodar os testes automaticamente a cada push ou pull request.

i Casos de Teste Detalhados

Módulo 1: Autenticação (`/api/v1/auth`)

ID do Teste	Descrição	Passos	Resultado Esperado
AUTH-001	Registrar um novo usuário com sucesso	1. Enviar <code>POST</code> para <code>/register</code> com <code>name</code> , <code>email</code> (único) e <code>password</code> válidos.	<ul style="list-style-type: none"> • Status <code>201 Created</code>. -&lt;br&gt;- Resposta contém <code>success: true</code> e os dados do usuário com um token JWT.
AUTH-002	Tentar registrar um usuário com e-mail já existente	1. Enviar <code>POST</code> para <code>/register</code> com um e-mail que já existe no banco.	<ul style="list-style-type: none"> • Status <code>400 Bad Request</code>. -&lt;br&gt;- Mensagem de erro "User already exists".
AUTH-003	Fazer login com credenciais válidas	1. Enviar <code>POST</code> para <code>/login</code> com <code>email</code> e <code>password</code> corretos.	<ul style="list-style-type: none"> • Status <code>200 OK</code>. -&lt;br&gt;- Resposta contém <code>success: true</code> e os dados do usuário com um novo token JWT.
AUTH-004	Tentar fazer login com senha incorreta	1. Enviar <code>POST</code> para <code>/login</code> com <code>email</code> correto e <code>password</code> incorreto.	<ul style="list-style-type: none"> • Status <code>401 Unauthorized</code>. -&lt;br&gt;- Mensagem de erro "Invalid email or password".
AUTH-005	Acessar perfil (<code>/me</code>) com token válido	1. Fazer login para obter um token. 2. Enviar <code>GET</code> para <code>/me</code> com o header <code>Authorization: Bearer <token></code> .	<ul style="list-style-type: none"> • Status <code>200 OK</code>. -&lt;br&gt;- Resposta contém os dados do usuário (sem a senha).
AUTH-006	Tentar acessar perfil (<code>/me</code>) sem token	1. Enviar <code>GET</code> para <code>/me</code> sem o header de autorização.	<ul style="list-style-type: none"> • Status <code>401 Unauthorized</code>. -&lt;br&gt;- Mensagem de erro indicando falta de autorização.
AUTH-007	Atualizar o perfil do usuário	1. Fazer login para obter um token. 2. Enviar <code>PUT</code> para <code>/profile</code> com um novo <code>name</code> , com uma nova senha.	<ul style="list-style-type: none"> • Status <code>200 OK</code>. -&lt;br&gt;- Resposta contém os dados do usuário atualizados.

AUTH-008	Atualizar o perfil do usuário	1. Fazer login para obter um token. 2. Enviar <code>PUT</code> para <code>/profile</code> apenas com um novo <code>name</code> .	<ul style="list-style-type: none"> • Status <code>200 OK</code>. - Resposta contém os dados do usuário atualizados.
-----------------	-------------------------------	---	--

Módulo 2: Gerenciamento de Usuários (/api/v1/users) (Casos de Teste Sugeridos)

Esta seção foca nas ações que um administrador pode realizar sobre todos os usuários do sistema.

ID do Teste	Descrição	Passos	Resultado Esperado
USR-001	(Admin) Listar todos os usuários	1. Fazer login como Admin. 2. Enviar GET para <code>/users</code> .	Status 200 OK. - A resposta contém uma lista de objetos de usuário. VERIFICAÇÃO CRÍTICA: Nenhum dos objetos de usuário deve conter o campo <code>password</code> .
USR-002	(Segurança) Tentar listar usuários como usuário comum	1. Fazer login como Usuário Comum. 2. Enviar GET para <code>/users</code> .	Status 403 Forbidden. - Mensagem de erro indicando falta de permissão.
USR-003	(Admin) Obter um usuário específico por ID	1. Obter um <code>userId</code> da listagem de USR-001. 2. Fazer login como Admin. 3. Enviar GET para <code>/users/{id}</code> .	Status 200 OK. - A resposta contém os dados do usuário solicitado (sem a senha).
USR-004	(Admin) Atualizar os dados de um usuário	1. Obter um <code>userId</code> . 2. Fazer login como Admin. 3. Enviar PUT para <code>/users/{id}</code> com um <code>name</code> ou <code>role</code> atualizado.	Status 200 OK. - A resposta contém os dados do usuário com as informações atualizadas.
USR-005	(Segurança) Tentar atualizar outro usuário como usuário comum	1. Obter o ID do usuário Admin. 2. Fazer login como Usuário Comum. 3. Enviar PUT para <code>/users/{adminId}</code> .	Status 403 Forbidden.
USR-006	(Admin) Deletar um usuário	1. Criar um novo usuário para o teste. 2. Fazer login como Admin. 3. Enviar	Status 200 OK. - Mensagem "User removed".

			DELETE para /users/{id_do_novo_usuario} .	
USR-007	(Integridade) Verificar se o usuário deletado não consegue mais logar	1. Executar os passos do USR-006.
2. Tentar fazer login com as credenciais do usuário que foi deletado.	Status 401 Unauthorized.
- Mensagem de erro "Invalid email or password".	

Módulo 3: Filmes (/api/v1/movies) 🔍

ID do Teste	Descrição	Passos	Resultado Esperado
MOV-001	Listar todos os filmes (acesso público)	1. Enviar GET para /movies .	<ul style="list-style-type: none"> • Status 200 OK . &lt;br> - A resposta contém uma lista de filmes.
MOV-002	Obter detalhes de um filme por ID (acesso público)	1. Enviar GET para /movies/{id} com um ID válido.	<ul style="list-style-type: none"> • Status 200 OK . &lt;br> - A resposta contém os dados completos do filme solicitado.
MOV-003	Tentar obter um filme com ID inválido/inexistente	1. Enviar GET para /movies/{id} com um ID que não existe.	<ul style="list-style-type: none"> • Status 404 Not Found . &lt;br> - Mensagem "Movie not found".
MOV-004	(Admin) Criar um novo filme	1. Fazer login como Admin.
 2. Enviar POST para /movies com dados de um novo filme.	<ul style="list-style-type: none"> • Status 201 Created . &lt;br> - A resposta contém os dados do filme criado.
MOV-005	(Segurança) Tentar criar um novo filme como usuário comum	1. Fazer login como Usuário Comum.
 2. Enviar POST para /movies .	<ul style="list-style-type: none"> • Status 403 Forbidden . &lt;br> - Mensagem de erro indicando que o usuário não tem permissão.
MOV-006	(Admin) Deletar um filme	1. Fazer login como Admin.
 2. Enviar DELETE para /movies/{id} com um ID válido.	<ul style="list-style-type: none"> • Status 200 OK . &lt;br> - Mensagem "Movie removed".
MOV-007	(Segurança) Tentar deletar um filme como usuário comum	1. Fazer login como Usuário Comum.
 2. Enviar DELETE para /movies/{id} .	<ul style="list-style-type: none"> • Status 403 Forbidden .

módulo 4: Sessões (/api/v1/sessions)

ID do Teste	Descrição	Passos	Resultado Esperado
-------------	-----------	--------	--------------------

SES-001	Listar todas as sessões (acesso público)	1. Enviar <code>GET</code> para <code>/sessions</code> .	<ul style="list-style-type: none"> Status <code>200 OK</code>. &lt;br&gt; - A resposta contém uma lista de sessões.
SES-002	Filtrar sessões por filme	1. Enviar <code>GET</code> para <code>/sessions?movie={movieId}</code> .	<ul style="list-style-type: none"> Status <code>200 OK</code>. &lt;br&gt; - Retorna apenas sessões para o filme especificado.
SES-003	Obter detalhes de uma sessão (público)	1. Enviar <code>GET</code> para <code>/sessions/{id}</code> com um ID válido.	<ul style="list-style-type: none"> Status <code>200 OK</code>. &lt;br&gt; - Contém dados da sessão, incluindo o mapa de assentos (<code>seats</code>).
SES-004	(Admin) Criar uma nova sessão	1. Fazer login como Admin .
 2. Enviar <code>POST</code> para <code>/sessions</code> com dados válidos.	<ul style="list-style-type: none"> Status <code>201 Created</code>. &lt;br&gt; - A resposta contém os dados da sessão criada, com o mapa de assentos gerado.
SES-005	(Admin) Resetar os assentos de uma sessão	1. Fazer login como Admin .
 2. Enviar <code>PUT</code> para <code>/sessions/{id}/reset-seats</code> .	<ul style="list-style-type: none"> Status <code>200 OK</code>. &lt;br&gt; - Todos os assentos na sessão retornam ao status "available".
SES-006	(Segurança) Tentar criar uma sessão como usuário comum	1. Fazer login como Usuário Comum .
 2. Enviar <code>POST</code> para <code>/sessions</code> .	<ul style="list-style-type: none"> Status <code>403 Forbidden</code>.

Módulo 5: Reservas (/api/v1/reservations) - FLUXO MAIS CRÍTICO ☀

ID do Teste	Descrição	Passos	Resultado Esperado
RES-001	Criar uma reserva com sucesso	1. Fazer login como Usuário Comum .
 2. Escolher uma sessão e assentos com status "available".
 3. Enviar <code>POST</code> para <code>/reservations</code> com os dados da sessão e dos assentos.	<ul style="list-style-type: none"> Status <code>201 Created</code>. &lt;br&gt; - Resposta contém os dados da reserva. &lt;br&gt; - Os assentos selecionados na sessão correspondente devem ter seu status alterado para "occupied".
RES-002	Tentar criar uma reserva para um assento já ocupado	1. Fazer login.
 2. Escolher uma sessão e um assento com status "occupied".
 3. Enviar <code>POST</code> para <code>/reservations</code> .	<ul style="list-style-type: none"> Status <code>400 Bad Request</code>. &lt;br&gt; - Mensagem de erro informando que os assentos não estão disponíveis.
RES-003	Listar "minhas reservas"	1. Fazer login como Usuário Comum que já possui reservas.
 2. Enviar <code>GET</code> para <code>/reservations</code> .	<ul style="list-style-type: none"> Status <code>200 OK</code>. &lt;br&gt; - A resposta contém uma lista de reservas pertencentes apenas ao usuário logado.

		<code>GET</code> para <code>/reservations/me</code> .	
RES-004	Obter detalhes de uma reserva própria	<p>1. Fazer login como Usuário Comum. &lt;br></p> <p>2. Enviar <code>GET</code> para <code>/reservations/{id}</code> onde <code>{id}</code> é de uma reserva própria.</p>	<ul style="list-style-type: none"> • Status <code>200 OK</code>. &lt;br> - Retorna os detalhes da reserva.
RES-005	(Segurança) Tentar obter detalhes de uma reserva de outro usuário	<p>1. Fazer login como Usuário A. &lt;br></p> <p>2. Enviar <code>GET</code> para <code>/reservations/{id}</code> onde <code>{id}</code> é de uma reserva do Usuário B.</p>	<ul style="list-style-type: none"> • Status <code>403 Forbidden</code>. &lt;br> - Mensagem de erro "Not authorized to access this reservation".
RES-006	(Admin) Listar todas as reservas do sistema	<p>1. Fazer login como Admin. &lt;br></p> <p>2. Enviar <code>GET</code> para <code>/reservations</code> .</p>	<ul style="list-style-type: none"> • Status <code>200 OK</code>. &lt;br> - Retorna uma lista com todas as reservas de todos os usuários.
RES-007	(Admin) Deletar uma reserva (simula um cancelamento com reembolso)	<p>1. Fazer login como Admin. &lt;br></p> <p>2. Enviar <code>DELETE</code> para <code>/reservations/{id}</code> .</p>	<ul style="list-style-type: none"> • Status <code>200 OK</code>. &lt;br> - A reserva é removida. &lt;br> - VERIFICAÇÃO DE INTEGRAÇÃO: Os assentos correspondentes na sessão devem voltar para o status "available".
RES-008	Tentar criar uma reserva sem autenticação	<p>1. Fazer um post reservation sem o token de autenti</p>	

Módulo 6: Salas de Cinema (`/api/v1/theaters`) (Casos de Teste Sugeridos) ☕

ID do Teste	Descrição	Passos	Resultado Esperado
THE-001	Listar todas as salas (acesso público)	Enviar <code>GET</code> para <code>/theaters</code> .	Status 200 OK.
- A resposta contém uma lista de todas as salas de cinema cadastradas.
THE-002	Obter detalhes de uma sala por ID (acesso público)	Enviar <code>GET</code> para <code>/theaters/{id}</code> com um ID válido.	Status 200 OK.
- A resposta contém os dados completos da sala (nome, capacidade, etc.).
THE-003	(Admin) Criar uma nova sala	<p>1. Fazer login como Admin. &lt;br></p> <p>2. Enviar <code>POST</code> para <code>/theaters</code> com dados válidos (ex: <code>name</code> , <code>capacity</code>).</p>	Status 201 Created.
- A resposta contém os dados da sala criada.

THE-004	(Segurança) Tentar criar uma sala como usuário comum	1. Fazer login como Usuário Comum.
2. Enviar POST para /theaters .	Status 403 Forbidden.
- Mensagem de erro indicando que o usuário não tem permissão.
THE-005	(Admin) Atualizar os dados de uma sala	1. Fazer login como Admin.
2. Enviar PUT para /theaters/{id} com um novo name ou capacity .	Status 200 OK.
- A resposta contém os dados da sala atualizados.
THE-006	(Admin) Deletar uma sala vazia	1. Fazer login como Admin.
2. Enviar DELETE para /theaters/{id} de uma sala que não tenha sessões futuras.	Status 200 OK.
- Mensagem "Theater removed".
THE-007	(Integridade) Tentar deletar uma sala com sessões ativas	1. Fazer login como Admin.
2. Enviar DELETE para /theaters/{id} de uma sala que tenha sessões associadas.	Status 400 Bad Request.
- Mensagem de erro indicando que a sala não pode ser removida pois possui sessões ativas.

Matriz de Risco Final e Completa ☀

ID do Risco	Módulo	Descrição do Risco	Impacto	Probabilidade	Nível de Risco	Estratégia de Mitigação (Caso de Teste)
RISK-RES-01	Reservas	Double booking em assentos.	Crítico	Média	Crítico	RES-001 , RES-002
RISK-RES-02	Reservas	Reserva não atualiza o status do assento.	Crítico	Média	Crítico	RES-001
RISK-SEC-01	Reservas	Acesso à reserva de outro usuário.	Crítico	Média	Crítico	RES-004 , RES-005
RISK-SEC-02	Filmes / Sessões	Não-admin gerenciando filmes/sessões.	Crítico	Média	Crítico	MOV-005 , MOV-007 , SES-006
RISK-INT-02	Salas / Sessões	Excluir sala com sessões ativas.	Crítico	Média	Crítico	THE-007
RISK-SEC-04	Salas de Cinema	Não-admin gerenciando salas.	Crítico	Média	Crítico	THE-004

RISK-DATA-02	Usuários	Vazamento de senhas na listagem de usuários.	Crítico	Média	Crítico	USR-001
RISK-SEC-05	Usuários	Não-admin acessando/modificando outros usuários.	Crítico	Média	Crítico	USR-002 , USR-005
RISK-SEC-03	Autenticação	Acesso a rotas protegidas sem token.	Crítico	Baixa	Alto	AUTH-006 , RES-008
RISK-INT-01	Reservas	Cancelamento não libera os assentos.	Alto	Média	Alto	RES-007
RISK-AUTH-02	Autenticação	Login com credenciais inválidas.	Alto	Média	Alto	AUTH-004
RISK-FUNC-01	Reservas	Usuário não conseguir criar reserva.	Alto	Média	Alto	RES-001
RISK-FUNC-03	Sessões	Mapa de assentos com erro ou ausente.	Alto	Média	Alto	SES-003 , SES-004
RISK-FUNC-05	Salas de Cinema	Admin não conseguir gerenciar salas.	Alto	Média	Alto	THE-003 , THE-005
RISK-FUNC-07	Usuários	Admin não conseguir gerenciar usuários.	Alto	Média	Alto	USR-001 , USR-004 , USR-006
RISK-INT-03	Usuários	Deletar usuário não invalida seu acesso.	Alto	Média	Alto	USR-007
RISK-AUTH-01	Autenticação	Falha no login válido.	Alto	Baixa	Médio	AUTH-003
RISK-DATA-01	Autenticação	Registro com e-mail duplicado.	Médio	Média	Médio	AUTH-002
RISK-ERR-01	Todos	API falhar (crash) com dados inválidos.	Médio	Alta	Médio	MOV-003 , AUTH-002 , RES-002
RISK-FUNC-02	Reservas	Usuário não conseguir listar suas reservas.	Alto	Baixa	Médio	RES-003

RISK-FUNC-04	Filmes / Sessões	Falha na listagem pública de filmes/sessões.	Alto	Baixa	Médio	MOV-001 , SES-001
RISK-FUNC-06	Salas de Cinema	Falha na listagem pública de salas.	Médio	Baixa	Médio	THE-001 , THE-002