

Challenge PB AWS & AI for QE

Parte 1: Planejamento Funcional [🔗](#)

Este documento detalha a fase inicial de planejamento para o desafio final "PB AWS & AI for QE", focado na garantia da qualidade funcional do sistema Cinema App. Ele abrange a análise de requisitos, a definição do escopo de teste, a priorização de funcionalidades e a seleção de técnicas de teste, estabelecendo uma base sólida para a automação e a execução de testes exploratórios. Meu objetivo é assegurar que o sistema atenda às expectativas do usuário e às regras de negócio, utilizando abordagens eficazes de QA.

Objetivo Principal [🔗](#)

Garantir a qualidade funcional do sistema Cinema App (Back-end e Front-end), validando a aderência aos requisitos especificados nas User Stories e na documentação da API. Além disso, visa identificar e registrar defeitos, bem como oportunidades de melhoria, antes da entrega final do produto, aumentando a confiança no software.

Escopo dos Testes Funcionais [🔗](#)

O escopo dos testes funcionais para este desafio cobre as seguintes áreas do sistema Cinema App, em suas camadas de **API (Back-end)** e **UI (Front-end)**:

- **Autenticação e Autorização:** Registro, Login e Logout de usuários (comum para API e UI).
- **Gestão de Filmes:** Visualização da listagem, detalhes do filme (UI e API), e criação de filmes (apenas API para usuários administradores).
- **Gestão de Sessões:** Visualização de horários de sessões por filme (UI e API).
- **Reservas (Criação):** Criação de reservas de assentos (principalmente via API, com validação de regras de negócio de assentos).

Fora do Escopo Inicial para Automação: [🔗](#)

Testes de Performance e Carga (apenas validação de tempo de resposta nas requisições).

Testes de Segurança (avaliação mais aprofundada de vulnerabilidades).

Testes completos de usabilidade e acessibilidade (abordagem inicial via testes exploratórios).

Testes de gerenciamento de usuários, teatros e outras operações administrativas completas via UI.

Análise e Priorização das Funcionalidades [🔗](#)

A priorização das funcionalidades para o teste e automação foi baseada em critérios como a criticidade do negócio, o risco de falhas, a frequência de uso e a abrangência tecnológica (API vs. UI).

Prioridade ALTA (Must Have - Foco principal da Automação): [🔗](#)

- **US-AUTH-002:** Login de Usuário (API e UI): Essencial para o acesso a recursos personalizados. Testar credenciais válidas e inválidas.
- **US-AUTH-001:** Registro de Usuário (API e UI): Porta de entrada para novos usuários. Validação de formato (e-mail, senha) e duplicidade.

- **US-MOVIE-001:** Navegar na Lista de Filmes (API e UI): Fluxo principal de descoberta de conteúdo. Testar listagem e filtros.
- **US-MOVIE-002:** Visualizar Detalhes do Filme (API e UI): Essencial para a decisão do usuário. Testar exibição de informações completas e sessões.
- **POST /api/v1/movies (API):** Criação de filmes por administrador. Essencial para a massa de dados e validação de permissões.

Prioridade MÉDIA (A ser automatizado se o tempo permitir, caso contrário, será tratado como Melhoria): [🔗](#)

- **US-SESSION-001:** Visualizar Horários de Sessões (API e UI): Conecta filmes à reserva. Testar exibição de horários, teatro e disponibilidade.
- **US-RESERVE-001:** Selecionar Assentos para Reserva (API e UI - API primeiro): O coração do processo de negócio. Foco na validação de assentos disponíveis/ocupados.
- **US-RESERVE-002:** Processo de Checkout (API - se US-RESERVE-001 for automatizado): Finalização da compra. Testar sucesso e falhas de pagamento (simulado).

Técnicas de Teste Aplicadas e Justificativa [🔗](#)

Para cada fase e tipo de teste, serão aplicadas técnicas de design de testes que otimizam a cobertura e a eficiência:

- **Análise de Requisitos:** Leitura e interpretação aprofundada das documentações fornecidas (API-DOCUMENTATION.md, DOCUMENTATION.md, README.md dos repositórios) e USER-STORIES.md.

Por que: Garante uma compreensão clara do comportamento esperado e das regras de negócio antes de qualquer teste, evitando falsos positivos e negativos.

- **Modelagem Visual com Mapa Mental:** Utilização do XMind para criar uma representação visual hierárquica das funcionalidades e responses.

Por que: Facilita a visualização do escopo, ajuda a identificar lacunas na cobertura e serve como guia durante a criação e execução dos testes.

- **Teste Baseado em Risco:** Priorização dos cenários de teste com maior impacto para o negócio e maior probabilidade de falha, conforme a Matriz de Risco.

Por que: Otimiza o esforço de teste em um tempo limitado, focando nas áreas que trazem maior retorno sobre o investimento (ROI) em termos de qualidade e estabilidade.

- **Técnica de Partição de Equivalência (Classes de Equivalência):** Divisão dos dados de entrada em grupos (classes) que se espera que se comportem da mesma forma. Um valor representativo de cada classe é testado.

Por que: Reduz drasticamente o número de casos de teste necessários, mantendo uma alta probabilidade de encontrar defeitos, especialmente em campos com grandes domínios de valores (ex: senhas, duração de filmes, campos de busca).

- **Análise de Valor Limite (Boundary Value Analysis):** Teste de valores nas "bordas" das classes de equivalência (mínimo, máximo, um abaixo do mínimo, um acima do máximo).

Por que: Erros são frequentemente encontrados nos limites de faixas de valores, tornando esta técnica crucial para validar restrições e evitar falhas "off-by-one".

- **Tabelas de Decisão:** Para funcionalidades com múltiplas condições e regras de negócio complexas (ex: login, criação de filme com múltiplas validações).

Por que: Garante que todas as combinações relevantes de condições sejam testadas, documentando claramente o comportamento esperado para cada cenário.

- **Técnica de Fluxo Principal e Alternativo:** Mapeamento do "caminho feliz" (fluxo principal) e dos caminhos alternativos (ex: login com credenciais válidas vs. login com usuário inexistente).

Por que: Garante que os fluxos mais usados funcionem corretamente e que o sistema lide bem com variações esperadas.

- **Heurísticas para Testes Web (Alter Face, CRUD, Campos Obrigatórios, Navegação, Consistência de UI, Goldilocks, etc.):** Durante a fase de Teste Manual Exploratório, serão usadas heurísticas como guias para a exploração do Front-end.

Por que: Permite uma abordagem mais livre e criativa para encontrar bugs inesperados, inconsistências visuais, problemas de usabilidade e falhas em comportamentos interativos que roteiros formais poderiam ignorar.

Complementa a automação ao focar na "qualidade" percebida pelo usuário.

- **Validação de Código de Status HTTP, Corpo da Resposta, Campos Obrigatórios e Tipos de Dados (para API):** Verificação se a API retorna os status codes adequados (200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 404 Not Found, 500 Internal Server Error), se o JSON da resposta está corretamente estruturado, se todos os campos esperados estão presentes e com o tipo de dado correto.

Por que: Garante que a comunicação entre sistemas seja robusta e que a API se comporte conforme seu contrato, sendo a base para a automação de testes de API.

Mapa Mental [🔗](#)

