

Kauã Raffaello – Plano de testes do e-commerce ServeRest.

Apresentação

- Este documento descreve a estratégia de testes para a API ServeRest, no qual simula um e-commerce para fins de aprendizado.
-

Objetivo

- Garantir a qualidade do e-commerce ServeRest por meio de testes que garantem que os endpoints estejam funcionais garantindo a segurança da aplicação.
-

Resumo

- O ServeRest é um E-commerce genérico com os usuário de administrador, administrador do sistema, e usuário comum, usuário-alvo da plataforma. Esse planejamento de teste será realizado por conta de problemas de ferramentas do sistema. O planejamento de testes terão certas pessoas envolvidas, além do escopo explicado dos testes, estratégias, critérios, cronograma, recursos, ambiente de testes e possíveis riscos.
 - A hipótese deste teste é que os endpoints estejam todos corretos e funcionais.
-

Pessoas Envolvidas:

Neste planejamento teve apenas um total de 1 QA envolvido.

- Kauã Raffaello – QA
-

Escopo

- Endpoints de teste:

 login

 usuarios

 produtos

 carrinhos

Fora do escopo:

- Integrações externas, conexões.

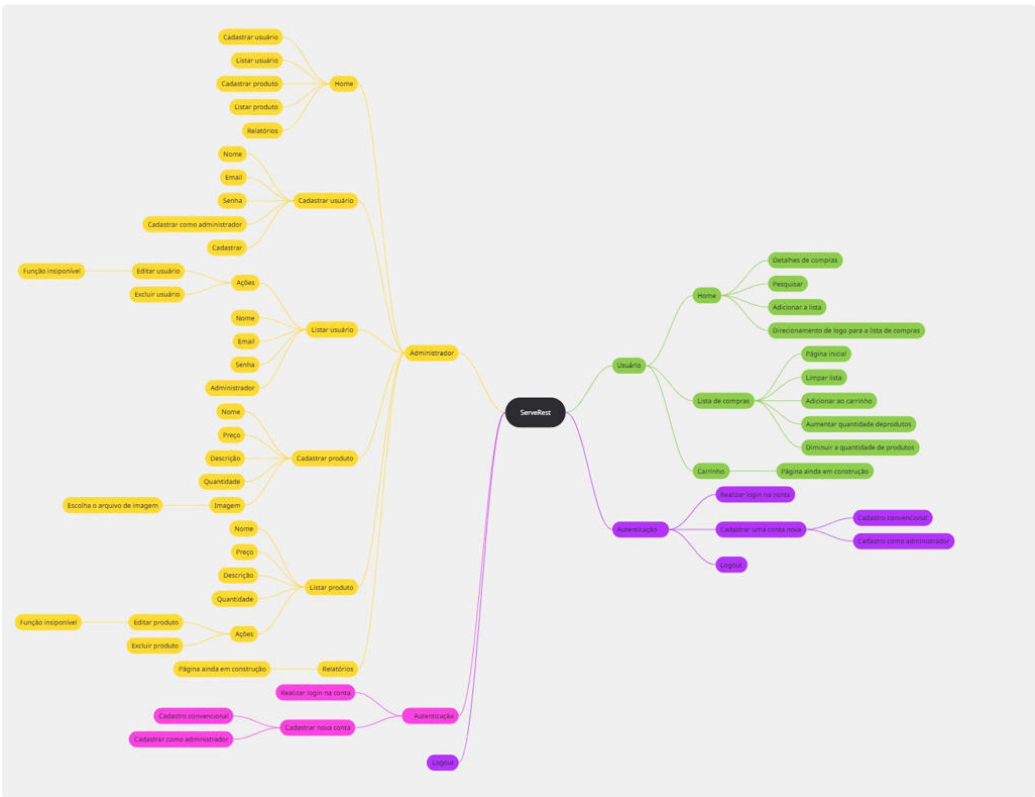
Análise

- Endpoints críticos: login, cadastro de usuários, cadastro de vendedores (admin), criação do produto e finalização do produto.
- Riscos: falhas de autenticação, problemas de autorização.

Técnicas Aplicadas

- Teste de caixa-branca.
- Testes de API.
- Testes de automação.

Mapa mental da aplicação



Cenários de teste planejado

1. Dado que eu esteja autenticado como vendedor

Quando envio uma requisição POST para a rota /produtos com dados válidos

Então o produto deve ser cadastrado com sucesso

E deve retornar o status 201.

2. Dado que eu esteja autenticado como vendedor

Quando envio uma requisição POST para a rota /produtos com **dados inválidos** (ex: preço negativo ou nome vazio)

Então o produto não deve ser cadastrado

E deve retornar o status 400 com a mensagem de erro correspondente

3. Dado que não exista produto com o ID informado

Quando envio uma requisição PUT para a rota /produtos/{id}

Então um novo produto deve ser criado

E deve retornar o status 201

Priorização da execução dos cenários de teste

Cenário	Prioridade	Justificativa
Criar produto com autenticação válida (POST)	Alta	É o fluxo principal de cadastro de produto; falha aqui compromete todo o uso da API de produtos.
Criar produto com autenticação inválida (POST)	Alta	Garante que a API valida corretamente os dados de entrada; falha aqui pode permitir produtos inválidos no sistema.
Atualizar produto inexistente (PUT cria novo)	Média	Caso de exceção; importante, mas não impede o funcionamento básico da rota de produtos.

Matriz de risco

Cenário	Impacto	Probabilidade	Classificação
Criar produto com autenticação válida (POST)	Alto	Alta	Crítico
Criar produto com dados inválidos (POST)	Alto	Alta	Crítico
Atualizar produto inexistente (PUT cria novo)	Médio	Média	Alto

Cobertura de testes

Cenário	Endpoint	Cobertura API
Criar produto com autenticação válida (POST)	<code>/produtos</code>	Cadastro de produto com dados válidos
Criar produto com dados inválidos (POST)	<code>/produtos</code>	Validação de dados obrigatórios e regras - Através de uma leve automação no “pre-request” do postman, com números aleatórios.
Atualizar produto inexistente (PUT cria novo)	<code>/produtos/{id}</code>	Criação de produto quando ID não existe. uma leve automatização no pre-request para gerar IDs

Testes candidatos a automação

Cenário	Endpoint	Justificativa para automação
Criar produto com autenticação válida (POST)	<code>/produtos</code>	Fluxo principal, precisa ser verificado sempre
Criar produto com dados inválidos (POST)	<code>/produtos</code>	Teste repetitivo e previsível, garante consistência
Atualizar produto inexistente (PUT cria novo)	<code>/produtos/{id}</code> <code>}</code>	Pode ser automatizado para validar exceções recorrentes