

Plano de Testes – API ServeRest

1. Apresentação

Este documento descreve o planejamento de testes para a API ServeRest, cujo objetivo é validar a conformidade das funcionalidades implementadas nos módulos **Usuários**, **Login** e **Produtos**.

A API será validada quanto à aderência às regras de negócio, requisitos funcionais e critérios de aceite definidos, com foco na qualidade e confiabilidade das operações REST.

2. Objetivo

Garantir que a API ServeRest atenda aos requisitos funcionais e regras de negócio definidas nas User Stories, validando o correto funcionamento dos endpoints, tratamento de erros, restrições e mensagens de retorno.

3. Escopo

Inclui:

- Testes funcionais dos endpoints:
 - `/usuarios`
 - `/login`
 - `/produtos`
- Testes positivos, negativos e de borda.
- Validação de status codes, estrutura e conteúdo de resposta.

Não inclui:

- Testes de carga, desempenho ou segurança avançada.
- Testes de interface gráfica.

- Testes de APIs não listadas nas User Stories.

4. Análise

A API ServeRest disponibiliza endpoints REST para gestão de usuários, autenticação e produtos.

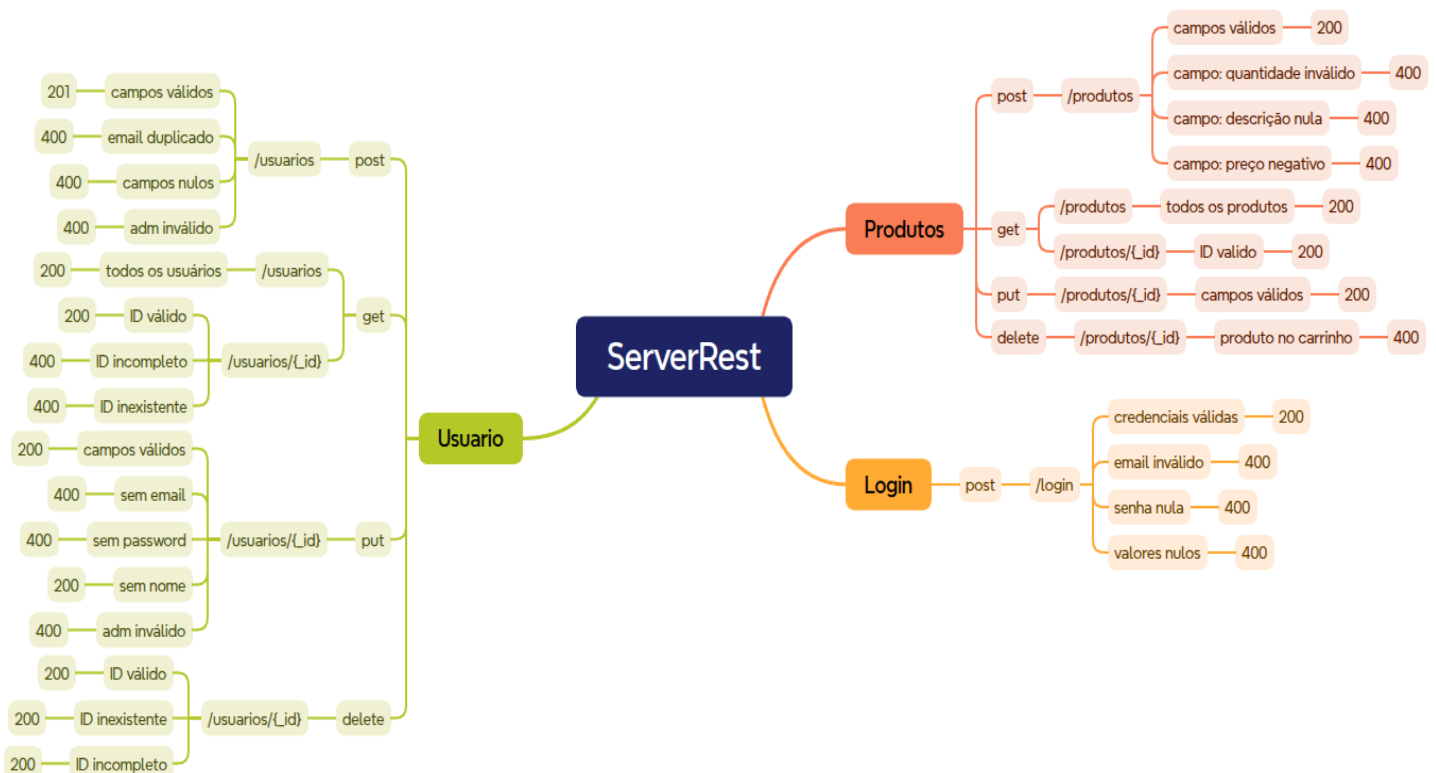
A análise baseou-se no **Swagger da API** e nas **User Stories US001, US002 e US003**, considerando:

- Operações CRUD para usuários e produtos.
- Autenticação via token Bearer.
- Regras específicas de validação para e-mails, senhas, nomes e duplicidades.

5. Técnicas Aplicadas

- **Particionamento de equivalência:** para dividir dados válidos e inválidos.
- **Análise de valor-limite:** para campos com restrições numéricas ou de tamanho.
- **Tabela de decisão:** para combinar diferentes condições e resultados esperados.
- **Teste exploratório:** para identificar cenários não documentados.

6. Mapa Mental da Aplicação



7. Cenários de Teste Planejados

Usuários (/usuarios)

1. Criar usuário com dados válidos.
2. Criar usuário com e-mail duplicado.
3. Criar usuário com e-mail inválido.
4. Criar usuário com senha menor que 5 caracteres.
5. Criar usuário com senha maior que 10 caracteres.
6. PUT com ID inexistente cria novo usuário.
7. PUT com e-mail já cadastrado retorna erro.
8. Consultar lista de usuários.

9. Deletar usuário inexistente retorna erro.

Login (/login)

1. Autenticar com credenciais corretas.
2. Autenticar com usuário inexistente.
3. Autenticar com senha incorreta.
4. Validar retorno de token Bearer.

Produtos (/produtos)

1. Criar produto com token válido.
2. Criar produto sem token.
3. Criar produto com nome duplicado.
4. PUT com ID inexistente cria novo produto.
5. PUT com nome duplicado retorna erro.
6. Deletar produto em carrinho retorna erro.
7. Listar produtos.

9. Matriz de Risco

Funcionalidade	Impacto	Probabilidade	Risco
Atualização de usuário	Alto	Alta	Alto
Cadastro de usuário	Alto	Média	Alto
Validações de senha	Alto	Alta	Alto

11. Testes Candidatos à Automação

- CRUD completo de usuários e produtos.
- Testes repetitivos de status code e mensagens.

12. Planejado x Executado

Cenário	Planejado	Executado	Status
Criar usuário válido	Sim	Sim	Passou
Criar usuário duplicado	Sim	Sim	Falhou
Criar senha negativa	Sim	Sim	Passou
Atualizar usuário sem nome	Sim	Sim	Passou