


# Plano de Testes - API ServeRest

## 1. Apresentação

Este documento apresenta o planejamento de testes para a API **ServeRest** (  ), com o objetivo de validar o cumprimento das regras de negócio e assegurar a qualidade das APIs REST.

## 2. Objetivo

Garantir que todos os endpoints da API ServeRest funcionem conforme os requisitos de negócio, assegurando integridade, desempenho e segurança, focado em:

- Validação de funcionalidades principais (CRUD das funcionalidades).
- Conformidade com as regras de negócio.
- Identificação de possíveis falhas e riscos.
- Viabilidade para automação de testes.

## 3. Escopo

- Testes funcionais das APIs REST de:
  - Autenticação → **login**
  - Usuários → **usuarios**
  - Produtos → **produtos**
  - Carrinhos → **carrinhos**
- Validação de regras de negócio (ex.: carrinho vinculado a um usuário, controle de estoque).

## 4. Análise

A API ServeRest é um simulador de e-commerce com operações CRUD para usuários, produtos e carrinhos.

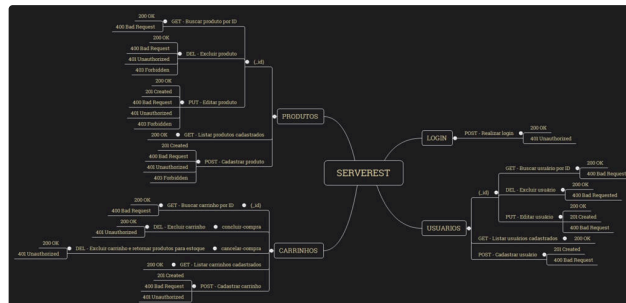
Alguns dos pontos a serem abrangidos:

- Estoque reduzido automaticamente ao inserir produto no carrinho.
- Apenas um carrinho por usuário.
- Exclusão de usuários com carrinho ativo não permitida.
- Exclusão de produtos vinculados a carrinhos não permitida.
- Autenticação obrigatória para operações restritas.

## 5. Técnicas aplicadas

- **Path Coverage** → verifica a cobertura da suíte de testes de acordo com os endpoints da API.

## 6. Mapa mental da aplicação



## 7. Cenários de teste a serem realizados

Alguns dos cenários a serem cobertos:

1. Criar usuário com dados válidos (201).
2. Criar usuário já existente (400).
3. Login com credenciais corretas (200).
4. Login com credenciais inválidas (401).
5. Criar produto com token válido (201).
6. Criar produto com nome já existente (400).
7. Criar produto sem token (401).
8. Criar produto sem administrador (403).
9. Criar carrinho com produto em estoque (201).
10. Criar carrinho com produto sem estoque (400).
11. Tentar criar dois carrinhos para o mesmo usuário (400).
12. Excluir produto vinculado a carrinho (400).
13. Concluir compra e validar redução de estoque.
14. Cancelar compra e validar reposição de estoque.

## 8. Priorização da execução dos cenários de teste

- **Alta:** Fluxos principais (CRUD, login, carrinho) e regras críticas (estoque, carrinho único).
- **Média:** Cenários de erros não críticos (campos inválidos, limites de tamanho).
- **Baixa:** Cenários de dados incomuns e campos opcionais.

## 9. Matriz de risco

Risco	Impacto	Probabilidade	Nível
Estoque incorreto após compra	Alto	Alta	Crítico
Usuário com mais de um carrinho	Alto	Média	Alto
Autenticação falha permite acesso restrito	Alto	Baixa	Alto
Exclusão de produto vinculado a carrinho permitida	Médio	Alta	Alto

#### 10. Cobertura de testes

- 100% dos endpoints principais.
- 100% das regras de negócio documentadas.
- Casos de sucesso e falha para cada operação CRUD.
- Cenários com dados no limite e inválidos.

#### 11. Testes candidatos à automação

- Login com credenciais válidas/inválidas.
- CRUD completo de usuários e produtos.
- Criação e manipulação de carrinhos.
- Cenários de estoque (redução e reposição).
- Cenários de erro comuns (status code e mensagens).