

# Plano de testes - API ServeRest

## Apresentação

Este documento descreve o planejamento de testes da API ServeRest, utilizando o Swagger oficial, User Stories (US001 – Usuários, US002 – Login, US003 – Produtos), o mapa mental construído e a collection Postman automatizada. O objetivo é garantir que as regras de negócio estejam corretas, cobrindo cenários positivos e negativos, bem como cenários alternativos.

## Objetivo

- Garantir a qualidade funcional da API ServeRest.
- Validar se as regras de negócio descritas nas User Stories estão implementadas corretamente.
- Identificar divergências entre especificação e comportamento real.
- Definir os testes candidatos à automação para execução contínua.

## Escopo

- **Incluído:**
  - Testes em **Usuários, Login, Produtos e Carrinhos** (conforme mapa mental).
  - Cenários de sucesso (200/201) e falha (400/401/403).
  - Testes automatizados via Postman.
- **Excluído:**
  - Testes de carga/performance.
  - Testes de segurança avançados.
  - Testes em UI.

## Análise

- **US001 (Usuários):** validação de cadastro, e-mail único, restrição de domínios (gmail/hotmail), senhas 5–10 caracteres.
- **US002 (Login):** autenticação JWT, bloqueio de usuários inexistentes/senha incorreta, expiração de token em 10 min.

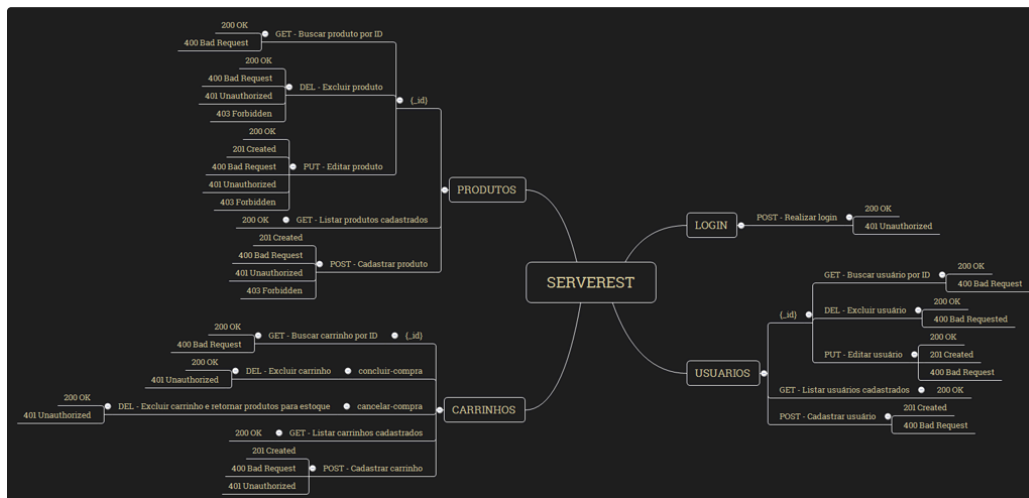
- **US003 (Produtos):** CRUD de produtos restrito a usuários autenticados/admin, nomes únicos, bloqueio de exclusão se em carrinho.
- **Mapa mental:** Define os status code esperados em cada operação.
- **Collection Postman:** Cobre parte dos cenários planejados, validando via scripts os retornos da API.

## Técnicas aplicadas

- Particionamento de equivalência (e-mails válidos/inválidos, senhas curtas/longas, autenticação válida/inválida).
- Análise de valor limite (senha com 4/5/10/11 caracteres).
- Baseado em requisitos (User Stories).
- Testes exploratórios (validação de mensagens, comportamento inesperado).
- Automação de API com Postman (Tests + dados externos).

## Mapa mental da aplicação

Utilizando o que foi testado no Postman, podemos criar o mapa mental da aplicação que abrange toda a aplicação do ServeRest



Mapa mental da API ServeRest

## Cenários de teste planejados

(Cobrem User Stories + Mapa mental)

### Usuários (US001)

- Criar usuário válido (200/201).
- Criar usuário com e-mail duplicado (400).
- Criar usuário com domínio bloqueado (gmail/hotmail) (400).
- Criar usuário com senha fora dos limites (400).
- Buscar usuário por ID válido (200).
- Buscar usuário por ID inválido (400).
- Atualizar usuário existente (200).
- Atualizar usuário inexistente (201 → cria novo).
- Excluir usuário válido (200).
- Excluir usuário inexistente (400).

### Login (US002)

- Login válido retorna token (200).
- Login com senha incorreta retorna (401).
- Login com usuário inexistente retorna (401).
- Validar expiração do token em 10 minutos.

### Produtos (US003)

- Criar produto válido com admin (201).
- Criar produto sem autenticação (401).
- Criar produto com cliente (403).
- Criar produto com nome duplicado (400).
- Listar produtos (200).
- Buscar produto por ID válido (200).
- Buscar produto por ID inválido (400).
- Editar produto válido (200).
- Editar produto com ID inexistente → cria novo (201).
- Excluir produto válido (200).
- Excluir produto sem autorização (401/403).
- Excluir produto em carrinho (400).

### Priorização da execução dos cenários de teste

- **Alta:** login, cadastro de usuário, cadastro de produto, criação de carrinho.

- **Média:** atualização e exclusão de recursos.
- **Baixa:** mensagens de erro detalhadas, validações de borda.

## Matriz de risco

Risco	Impacto	Probabilidade	Nível
Usuário duplicado cadastrado	Alto	Alto	Crítico
Produto cadastrado sem autenticação	Alto	Médio	Alto
Token JWT não expira corretamente	Médio	Baixo	Médio
Exclusão de produto em carrinho permitida	Alto	Médio	Alto
Cadastro com e-mails inválidos aceito	Médio	Alto	Alto

## Cobertura de testes

- **Swagger:** Coberto em CRUD completo.
- **User Stories:** Mapeadas integralmente.
- **Mapa mental:** Usado como guia de endpoints e status code.
- **Collection Postman:** Cobre cenários principais, mas alguns cenários de borda ainda não foram automatizados (ex: senha limite, domínios bloqueados).

## Testes candidatos a automação

- CRUD de Usuários e Produtos.

- Login válido e inválido.
- Fluxo de carrinho (criação, conclusão, cancelamento).
- Validações de duplicidade (usuário/produto).
- Expiração do token JWT (com delay).