


## Planejamento de testes - João Gabriel Oliveira Magalhães

### Planejamento de testes - Serverest

#### APRESENTAÇÃO

Este documento apresenta o planejamento de testes da API ServeRest (  [ServeRest](#) ), com o objetivo de garantir a qualidade, confiabilidade e aderência às regras de negócio definidas nas User Stories US001, US002, US003. O plano inclui análise detalhada de endpoints, priorização de cenários, riscos, cobertura de testes e possíveis automações

O planejamento foca nos endpoints principais da API: Usuários, Login e Produtos, validando operações de criação, atualização, listagem e exclusão de registros, bem como autenticação e restrições específicas, como validação de e-mail, senha e dependências com carrinhos.

Serão abordados cenários positivos, onde todas as operações são realizadas corretamente, e cenários negativos, simulando falhas e entradas inválidas, para assegurar que a API se comporta conforme esperado mesmo em situações fora do padrão.

#### OBJETIVO

O objetivo deste planejamento de testes é garantir que a API Sereverest funcione corretamente, cumprindo todas as regras de negócio definidas nas User Stories US001, US002 e US003.

Este plano servirá como base para a execução de testes manuais e automatizados, permitindo identificar falhas, inconsistências e comportamento inesperados nos endpoints de Usuários, Login e Produtos.

Além disso, o planejamento orienta a documentação de issues encontradas durante os testes, garantindo maior confiabilidade da API para desenvolvimento e manutenção futura.

#### ESCOPO

O escopo deste planejamento de testes abrange os endpoints principais da API ServeRest, incluindo Usuários, Login e Produtos.

- Usuários: Testes de criação, atualização, listagem e exclusão, garantindo validação de campos obrigatórios, E-mails válidos e senhas dentro dos limites estabelecidos.
- Login: Testes de autenticação de usuários válidos e inválidos, geração de token Bearer e validade de 10 minutos.
- Produtos: Testes de criação, atualização, listagem e exclusão, incluindo restrições como nomes duplicados e bloqueio de testes necessários para validar dependências de produtos (ex: não excluir produto dentro de um carrinho).

Fora do escopo estão testes de performance, segurança avançada e demais endpoints de carrinhos que não impactam diretamente os cenários das User Stories US001, US002, US003.

Este escopo garante que os testes sejam focados, completos e eficientes, cobrindo as regras de negócio essenciais da API ServRest.

## **ANÁLISE**

A análise da API ServeRest considera todos os endpoints principais e suas funcionalidades:

Usuários (US001):

- POST /usuarios: Criar novo usuário com validação de campos obrigatórios (nome, e-mail, senha, administrador), impedir e-mails duplicados ou de provedores proibidos (gmail, hotmail), validar tamanho da senha (5-10 caracteres).
- GET /usuarios: Listar usuários cadastrados, retornar erro para usuários inexistentes.
- PUT /usuarios/{\_id}: Atualizar usuário existente ou criar novo se ID não existir, mantendo validação de duplicidade de E-mail.
- DELETE /usuarios/{\_id}: Excluir usuário existente, retornar erro se não existir.

LOGIN (US002):

- POST /login: Autenticar usuário cadastrado, gerar token Bearer válido por 10 minutos.
- Cenários negativos: Autenticação com usuário inexistente ou senha inválida retorna 401 Unauthorized.

### PRODUTOS (US003):

- POST /produtos: Criar produto com autenticação válida, impedir nomes duplicados.
  - PUT /produtos/{\_id}: Atualizar produto existente ou criar novo se ID não existir, mantendo regras de nomes duplicados.
- DELETE /produtos/{\_id}: Excluir produto, verificando restrição caso esteja em carrinho.
- GET /produtos: Listar produtos cadastrados.

### CARRINHOS (DEPENDÊNCIA):

- Apenas criar cenário suficiente para testar restrição de exclusão de produto presente em carrinho, sem CRUD completo de carrinhos.

A análise garante que todos os critérios das User Stories US001, US002 e US003 sejam contemplados, incluindo fluxos positivos e negativos, regras de validação e dependências entre endpoints.

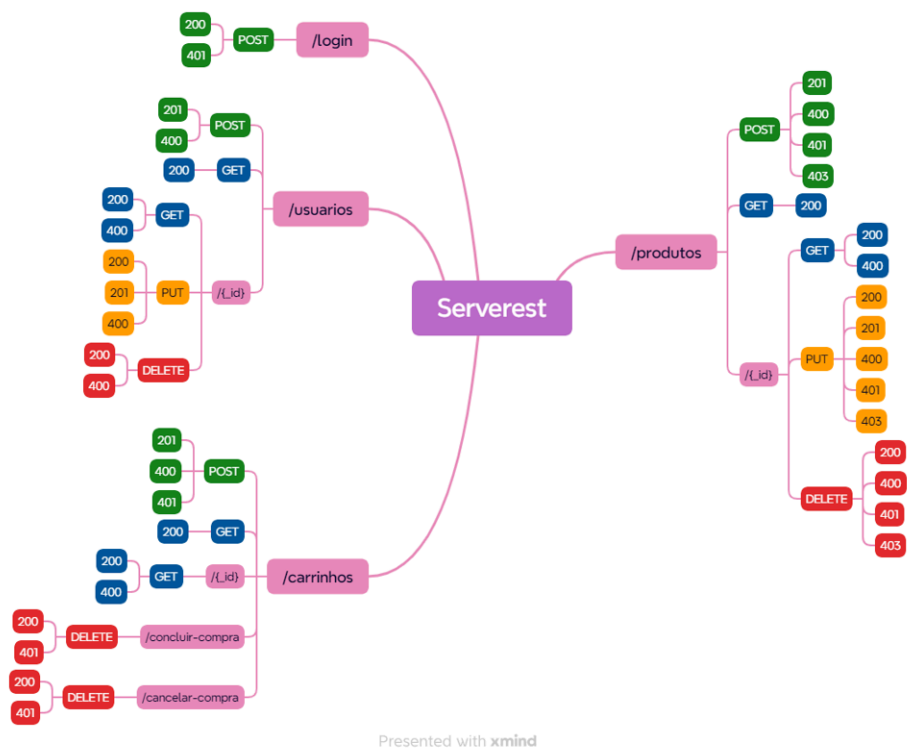
### **TÉCNICAS APLICADAS**

Para testar a API ServeRest, serão aplicadas as seguintes técnicas:

- Teste funcional: Validar todas as operações CRUD dos endpoints de Usuários e Produtos, além da autenticação do Login.
- Teste de validação de dados: Verificar obrigatoriedade de campos, formatos corretos de E-mail, tamanho de senha e nomes de produtos.
- Teste negativo: Simular falhas de autenticação, tentar criar registros duplicados cadastrar usuários com E-mails proibidos e excluir produtos presentes em carrinhos.
- Teste de integração mínima: Validar dependências entre endpoints, como produtos que não podem ser excluídos se estiverem em carrinhos.
- Teste exploratório: Investigar comportamentos alternativos não previstos diretamente nas User Stories.
- Automação parcial: Scripts no Postman serão utilizados para validar respostas, status codes, geração de token e consistência dos dados.

Essas técnicas garantem que os testes sejam abrangentes, cobrindo tanto fluxos esperados quanto situações de erro, aumentando a confiabilidade e a qualidade da API.

MAPA MENTAL DA APLICAÇÃO



CENÁRIOS DE TESTES PLANEJADOS

USUÁRIOS (US001)

Prioridade	Cenário	Tipo	Resultado esperado
Alta	Criar usuário com E-mail válido e senha correta	Funcional	Usuário criado com sucesso, status code 201
Alta	Criar usuário com E-mail já existente	Negativo	Retornar erro, não permitir duplicidade status code 400

Média	Atualizar usuário inexistente via PUT	Funcional	Novo usuário criado com dados informados
Média	Criar usuário com senha inválida (menor que 5 ou maior que 10 caracteres)	Validação	Retornar erro, status code 400, não permitir cadastro
Média	Criar usuário com E-mail inválido	Validação	Retornar erro, status code 400, não permitir cadastro
Média	Cadastrar usuário com E- mail de provedor proibido (gmail/hotmail)	Validação	Retornar erro, status code 400
Baixa	Deletar usuário inexistente	Negativo	Retornar erro, status code 404

#### LOGIN (US002)

Prioridade	Cenário	Tipo	Resultado esperado
Alta	Autenticar usuário válido	Funcional	Token Bearer gerado, status code 200
Alta	Autenticar usuário inexistente	Negativo	Retornar status code 401 Unauthorized

Média	Autenticar com senha inválida	Negativo	Retornar status code 401 Unauthorized
Baixa	Validar duração do token (10 minutos)	Funcional	Token expira após 10 minutos.

#### PRODUTOS (US003)

Prioridade	Cenário	Tipo	Resultado esperado
Alta	Criar produto com nome válido e autenticação	Funcional	Produto criado, status code 201
Alta	Criar produto com nome duplicado	Negativo	Retornar erro, status code 400
Alta	Excluir produto presente no carrinho	Integração	Exclusão bloqueada, status code 400
Média	Atualizar produto inexistente via PUT	Funcional	Produto criado com dados informados
Média	Tentar ação de produto sem autenticação	Negativo	Retornar erro, status code 401
Baixa	Listar produtos	Funcional	Retornar lista de produtos cadastrados

#### CARRINHOS (DEPENDÊNCIA MÍNIMA)


Prioridade	Cenário	Tipo	Resultado esperado
Alta	Criar carrinho com produto	Funcional	Carrinho criado com sucesso, status 201

#### PRIORIZAÇÃO DA EXECUÇÃO DOS CENÁRIOS DE TESTE

Ordem	Endpoint	Cenário	Prioridade
1	/usuarios	Criar usuário válido	Alta
2	/login	Autenticar usuário válido	Alta
3	/produtos	Criar produto com nome válido e autenticação	Alta
4	/carrinhos	Criar carrinho com produto	Alta
5	/usuarios	Criar usuário com E-mail duplicado	Alta
6	/produtos	Criar produto com nome duplicado	Alta
7	/usuarios	Cadastrar usuário com provedor proibido	Alta
8	/login	Autenticar usuário inexistente	Alta

9	/produtos	Excluir produto que está em carrinho	Alta
10	/usuários	Atualizar usuário inexistente via PUT	Média
11	/usuarios	Criar usuário com senha inválida	Média
12	/usuarios	Criar usuário com E-mail inválido	Média
13	/produtos	Atualizar produto inexistente via PUT	Média
14	/produtos	Tentar ação de produto sem autenticação	Média
15	/login	Autenticar com senha inválida	Média
16	/login	Validar duração do token	Baixa
17	/usuarios	Deletar usuário inexistente	Baixa
18	/produtos	Listar produtos	Baixa

## MATRIZ DE RISCO

	Sem impacto	Leve	Médio	Grave	Gravíssimo
---	-------------	------	-------	-------	------------



Quase certo					
Alta					Criar usuário com E-mail duplicado / Criar produto com nome duplicado / Excluir produto em carrinho
Média			Atualizar usuário inexistente e via PUT / Criar usuário com senha inválida / Criar usuário com E-mail inválido / Atualizar produto inexistente e via put / Tentar ação de produto		

			sem autenticação / Autenticar usuário inexistente / Autenticar com senha inválida		
Baixa	Deletar usuário inexistente / Listar produtos		Validar duração do token / Criar usuário válido / Autenticar usuário válido / Criar produto com nome válido e autenticação / Criar carrinho com produto		
Raro					

### Legenda:

Cinza - Baixo risco

Verde - Risco moderado

Amarelo - Risco elevado

Vermelho - Risco extremo

## COBERTURA DE TESTES

Endpoint	Cenário	Status do teste
/usuarios	Criar usuário válido	Passou
/usuarios	Criar usuário duplicado	Passou
/usuarios	Criar usuário email inválido	Passou
/usuarios	Criar usuário gmail	Não passou
/usuarios	Criar usuário hotmail	Não passou
/usuarios	Criar usuário senha menor que 5	Não passou
/usuarios	Criar usuário senha maior que 10	Não passou
/usuarios	Atualizar usuário inexistente	Passou
/usuarios	Atualizar usuário existente	Passou
/usuarios	Deletar usuário existente	Passou
/usuarios	Deletar usuário inexistente	Passou
/login	Autenticar usuário válido	Passou
/login	Autenticar usuário inexistente	Passou
/login	Autenticar senha inválida	Passou

/login	Validar duração do token	Não testado
/produtos	Criar produto válido	Passou
/produtos	Criar produto duplicado	Passou
/produtos	Atualizar produto inexistente	Passou
/produtos	Tentar criar produto sem autenticação	Passou
/produtos	Tentar atualizar produto sem autenticação	Passou
/produtos	Tentar deletar produto sem autenticação	Passou
/produtos	Listar produtos	Passou
/produtos	Excluir produto no carrinho	Passou
/carrinhos	Criar carrinho com produto	Passou

Resumo: Dos 24 possíveis cenários de teste, 4 falharam e 1 não foi testado. Isso mostra que há incoerências com os critérios de aceite descritos nas User Stories, sendo elas:

- É possível criar um usuário com provedor proibido (@gmail, @hotmail);
- É possível criar um usuário com senha inválida (menor que 5 caracteres ou maior que 10 caracteres).

### TESTES CANDIDATOS A POSSÍVEL AUTOMAÇÃO

Endpoint	Cenário	Tipo	Automatizar?	Justificativa
----------	---------	------	--------------	---------------

/usuarios	Criar usuário válido	Funcional	Sim	Fluxo crítico do sistema, repetitivo e previsível
/usuarios	Criar usuário com E-mail duplicado	Negativo	Sim	Valida regra de negócio crítica sobre duplicidade
/usuarios	Cadastrar usuário com provedor proibido	Validação	Sim	Pode ser testado automaticamente com diferentes provedores
/usuarios	Atualizar usuário inexistente via PUT	Funcional	Sim	Verifica criação automática de novo usuário via PUT
/usuarios	Criar usuário com senha inválida	Validação	Sim	Teste de validação repetitivo, fácil de automatizar
/usuarios	Criar usuário com E-mail inválido	Validação	Sim	Pode ser testado com múltiplos formatos de E-mail
/usuarios	Deletar usuário inexistente	Negativo	Opcional	Teste de erro, menos crítico, pode ser manual

/login	Autenticar usuário válido	Funcional	Sim	Fluxo principal de autenticação crítico
/login	Autenticar usuário inexistente	Negativo	Sim	Valida regra de autenticação pode ser repetitivo
/login	Autenticar com senha inválida	Negativo	Sim	Verifica segurança e regras de login repetitivas
/login	Validar duração do token	Funcional	Opcional	Pode ser automatizado, mas envolve tempo, menos crítico
/produtos	Criar produto com nome válido e autenticação	Funcional	Sim	Fluxo crítico do marketplace repetitivo
/produtos	Criar produto com nome duplicado	Negativo	Sim	Regra de negócio importante, pode ser repetitiva
/produtos	Atualizar produto inexistente via PUT	Funcional	Sim	Valida criação automática

				via PUT, útil automatizar
/produtos	Listar produtos	Funcional	Opcional	Teste simples, pode ser manual ou automatizado
/produtos	Tentar ação de produto sem autenticação	Negativo	Sim	Verifica segurança e acesso indevido
/carrinhos	Criar carrinho com produto	Funcional	Sim	Fluxo crítico de vendas, repetitivo
/carrinhos	Excluir produto presente no carrinho	Integração	Sim	Valida regra de negócio e integração, útil automatizar