Plano de teste - API ServeRest

1. Apresentação

Este documento apresenta o planejamento de testes para a API ServeRest, cujo objetivo é validar as funcionalidades de usuários, login e produtos, garantindo a qualidade da aplicação conforme os critérios estabelecidos nas User Stories e nas especificações do Swagger.

2. Objetivo

Garantir que as funcionalidades da API estejam de acordo com os requisitos funcionais e regras de negócio, com cobertura completa dos principais fluxos e validações, por meio de testes manuais e automatizados.

3. Escopo

Inclui as rotas e funcionalidades:

- CRUD Usuários (Cadastro de vendedores)
- Autenticação (Login)
- CRUD Produtos

4. Análise

A análise foi baseada na documentação Swagger da API ServeRest e nas User Stories US001, US002 e US003, contemplando cenários positivos, negativos e alternativos, como restrições de e-mail, tokens expirados e criação via PUT.

5. Técnicas Aplicadas

- Testes funcionais baseados em requisitos e User Stories
- Análise de valores limite e validação de dados
- Testes baseados em riscos para priorização
- Testes exploratórios para casos alternativos

6. Mapa Mental da Aplicação

7. Cenários de Teste Planejados

ID	US Ref.	Cenário	Entrada / Pré-	Resultado	Status Code	Observações

			condiçã o	Esperad o		
CT001	US001	Criar usuário válido	Nome, e-mail válido (não gmail/ho tmail), senha 5-10 caracter es	Usuário criado com ID	201	E-mail único, validaçã o de senha
CT002	US001	Criar usuário com e-mail duplicad o	E-mail já cadastra do	Erro de duplicid ade	400	Bloqueio e-mail duplicad o
CT003	US001	Criar usuário com e-mail gmail/ho tmail	E-mail @gmail. com ou @hotma il.com	Erro de provedo r não permitid o	400	Restriçõ es de provedo res
CT004	US001	Atualizar usuário inexiste nte (PUT cria novo)	ID inexiste nte, dados válidos	Novo usuário criado	201	PUT cria registro se não existir

CT005	US001	Deletar usuário inexistente	ID inválido	Erro usuário não encontrado	404	Bloqueio para usuários inexistentes
CT006	US001	Listar usuários	-	Retorna lista	200	
CT007	US002	Login válido	Usuário cadastrado com credenciais corretas	Token Bearer válido (10 min)	200	
CT008	US002	Login com usuário inexistente	E-mail não cadastrado	Erro autenticação inválida	401	
CT009	US002	Login com senha incorreta	Senha errada	Erro autenticação inválida	401	
CT010	US002	Acesso com token expirado	Token expirado após 10 minutos	Acesso negado	401	
CT011	US003	Criar produto válido	Nome único, dados completos	Produto criado com ID	201	

CT012	US003	Criar produto com nome duplicado	Nome já cadastrado	Erro duplicidade nome	400	
CT013	US003	Criar produto sem autenticação	Sem token de autenticação	Erro autenticação	401	
CT014	US003	Atualizar produto inexistente (PUT cria novo)	ID inexistente, dados válidos	Novo produto criado	201	
CT015	US003	Excluir produto em carrinho	Produto vinculado a carrinho	Erro exclusão proibida	400	
CT016	US003	Listar produtos	-	Retorna lista	200	

8. Priorização da Execução

Prioridade	Casos de Teste	Justificativa
Alta	CT001, CT007, CT011, CT013, CT015	Funcionalidades críticas
Média	CT002, CT004, CT005, CT008,	Validações importantes

	CT009, CT010, CT012, CT014	
Baixa	CT006, CT016	Funcionalidades secundárias

9. Matriz de Risco

Risco	Probabilidade	Impacto	Nível	Mitigação
Usuário com e-mail duplicado	Alta	Alto	Alto	Validação rigorosa de cadastro
Login com credenciais inválidas	Alta	Alto	Alto	Mensagens claras e bloqueios
Produto criado com nome duplicado	Média	Alto	Alto	Validação de nome no backend
Exclusão de produto em carrinho ativo	Baixa	Alto	Médio	Bloqueio de exclusão pelo sistema

10. Cobertura de Testes

O calculo da cobertura deteste foi feito a partir do método path Coverage, que verifica a cobertura da suíte de testes de acordo com os endpoints que a API possui. Onde a análise é realizada pela quantidade de URI(URL + URN (*Resource name*)) que a API possui, e se for a mesma URI para métodos diferentes, considera-se apenas um, onde o ideal é realizar ao menos uma requisição para verificar cada endpoint.

- A API ServeRest Possui 9 Endpoints, sendo eles: **/Login, /usuarios, /usuarios/{_id}, /produtos, /produtos/{_id}, /carrinhos, /carrinhos/{_id}, /carrinhos/concluir-compra, /carrinhos/cancelar-compra.**

- Testes Realizados:
- Criar usuários; criar usuário duplicado; criar usuário com Gmail; deletar usuário inexistente; Listar Usuários; Fazer login com usuário existente; Fazer login com usuário inexistente; Fazer login com senha incorreta; Acesso com token expirado; Criar produto válido; Criar produto com nome duplicado; Criar produto sem autenticação; Atualizar produto inexistente (PUT cria novo); Excluir produto em carrinho;
  Endpoints testados diretamente:
- `/login` → testado (vários testes de login)
- `/usuarios` → testado (criação, listagem)
- `/usuarios/{_id}` → testado (deletar usuário inexistente)
- `/produtos` → testado (criar produtos, autenticação, nome duplicado)
- `/produtos/{_id}` → testado (atualizar produto inexistente, excluir produto em carrinho)
- `/carrinhos` → **não testado diretamente** (apenas usado para verificar vínculo produto)

Endpoints testados vs total

- Total de endpoints: 9
- Endpoints com testes diretos realizados: 5
- Endpoints sem testes diretos: 4

Calculo de cobertura Path Coverage (endpoints)

Cobertura= Numero de endpoints testados/Total de endpoints×100 =5/9×100≈55.56%

Resultado:

Métrica	Valor
Total de endpoints	9
Endpoints testados	5
Endpoints não testados	4
Cobertura Path Coverage (%)	55,56%

11. Testes Candidatos à Automação

- Criação/atualização/exclusão de usuários
- Login com token
- Criação/atualização/exclusão de produtos
- Testes de permissão e autenticação

12. Matriz de Rastreabilidade

US Ref.	Acceptance Criteria Principal	Casos de Teste Relacionados
US001	Campos obrigatórios, e-mail único, restrições	CT001, CT002, CT003, CT004, CT005, CT006
US002	Login correto, bloqueio de usuários inválidos	CT007, CT008, CT009, CT010
US003	CRUD produtos, autenticação obrigatória, restrições	CT011, CT012, CT013, CT014, CT015, CT016