Planejamento de Testes

Introdução

Este documento apresenta a estratégia completa de testes para a API ServeRest, alinhada com as User Stories US001 (Usuários), US002 (Login), US003 (Produtos) e US004 (Carrinhos).

Objetivos

- Validar o comportamento da API conforme especificado no Swagger e User Stories
- Garantir a cobertura completa dos critérios de aceite
- Identificar inconsistências entre documentação e implementação
- Fornecer métricas de qualidade para decisões estratégicas

Escopo

- Cadastro e login.
- Cadastro, gerenciamento e exclusão de produtos.
- Cadastro, gerenciamento e exclusão de carrinho.
- Rotas de administrador.

Funcionalidades Abrangidas

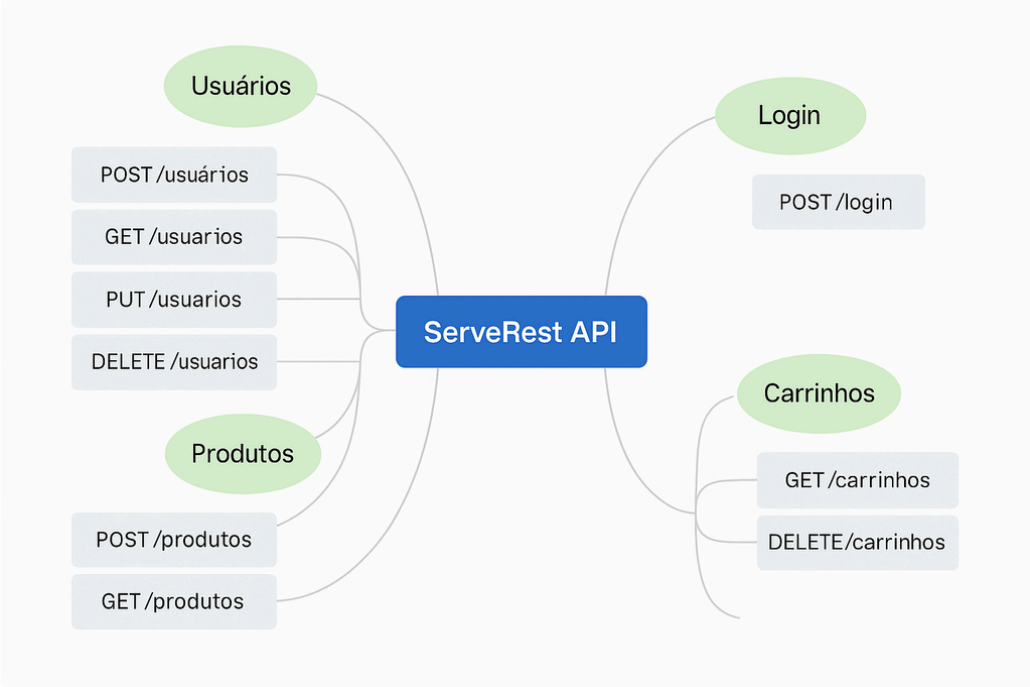
Módulo	Endpoints	Métodos HTTP
Usuários	`/usuarios`	POST, GET, PUT, DELETE
Login	`/login`	POST
Produtos	`/produtos`	POST, GET, PUT, DELETE
Carrinhos	`/carrinhos`	POST, GET, DELETE

Estratégia de Testes

Tipos de Testes

Tipo	Técnica	Ferramenta
Funcional	Validação de endpoints	Postman
Dados	Valores limite/partição	Postman Tests
Integração	Fluxos entre módulos	Postman

Mapa Mental



Cenários de Teste

US001 - Usuários

ID	Cenário	Método	Dados de Entrada	Resultado Esperado
U01	Cadastro válido	POST	Dados completos	201 Created + ID
U02	E-mail duplicado	POST	E-mail existente	400 Bad Request
U03	E-mail inválido (Gmail)	POST	usuario@gmail.com	400 Bad Request
U04	Senha curta (4 chars)	POST	"1234"	400 Bad Request
U05	Senha longa (11 chars)	POST	"12345678901"	400 Bad Request
U06	Listar usuários	GET	-	200 OK + Array
U07	Atualizar usuário	PUT	ID válido	200 OK
U08	Criar via PUT (ID novo)	PUT	ID inexistente	201 Created
U09	Excluir usuário	DELETE	ID válido	200 OK

US002 - Login

ID	Cenário	Método	Dados de Entrada	Resultado Esperado

L01	Login válido	POST	Credenciais corretas	200 OK + Token
L02	Usuário não cadastrado	POST	E-mail inválido	401 Unauthorized
L03	Senha incorreta	POST	Senha errada	401 Unauthorized
L04	Campo e-mail inválido	POST	E-mail: ""	400 Bad Request
L05	Campo senha vazio	POST	Senha: ""	400 Bad Request
L06	Token expirado	POST	Token >10min	401 Unauthorized
L07	Formato e-mail inválido	POST	"email_invalido"	400 Bad Request
L08	Cabeçalho ausente	POST	Sem Content-Type	400 Bad Request

US003 - Produtos

ID	Cenário	Método	Dados de Entrada	Resultado Esperado
P01	Cadastro autenticado	POST	Dados válidos + token	201 Created
P02	Sem autenticação	POST	Sem token	401 Unauthorized
P03	Nome duplicado	POST	Nome existente	400 Bad Request
P06	Listar produtos	GET	-	200 OK + Array
P07	Atualizar produto	PUT	ID válido + token	200 OK
P09	Excluir produto	DELETE	ID válido + token	200 OK

US004 - Carrinho

ID	Cenário	Método	Dados de Entrada	Resultado Esperado
C05	Listar carrinhos	GET	-	200 OK + Array
C06	Concluir compra	POST	ID carrinho + token	200 OK
C07	Cadastrar Carrinho	POST	ID carrinho + token	201 OK
C07	Buscar carrinho por ID	GET	ID carrinho válido	200 OK + Dados completos

C08	Concluir compra	POST	ID carrinho válido	200 OK + Mensagem sucesso

Priorização

Critérios de Priorização

1. **Criticidade:** Impacto no funcionamento.
2. **Frequência:** Uso comum na aplicação
3. **Complexidade:** Risco de falhas

Matriz de Risco

ID	Cenário	Probabilidade	Impacto	Severidade	Ações de Mitigação
U02	Cadastro válido	Baixa	Alto	Médio	Validar dados antes do POST.
U03	E-mail duplicado	Alta	Médio	Alto	Implementar verificação única no BD.
U04	E-mail inválido (Gmail)	Média	Baixo	Médio	Usar regex para validação de formato.
U05	Senha curta (4 chars)	Alta	Alto	Crítico	Exigir senha com mínimo de 6 caracteres.
U06	Senha longa (11 chars)	Baixa	Baixo	Baixo	Limitar campo a 10 caracteres no frontend.
U07	Listar usuários	Baixa	Baixo	Baixo	Garantir autenticação para GET.
U08	Atualizar usuário	Média	Alto	Alto	Validar ID e permissões antes do PUT.
U09	Criar via PUT (ID novo)	Baixa	Alto	Médio	Restringir criação apenas ao POST.

U10	Excluir usuário	Média	Alto	Alto	Confirmar ação com modal de confirmação.
L01	Login válido	Baixa	Alto	Médio	Monitorar tentativas de login.
L02	Usuário não cadastrado	Alta	Médio	Alto	Mensagem clara: "Credenciais inválidas".
L03	Senha incorreta	Alta	Médio	Alto	Bloquear após 3 tentativas falhas.
L04	Campo e-mail vazio	Média	Baixo	Médio	Validação required no frontend.
L05	Campo senha vazio	Média	Baixo	Médio	Validação required no frontend.
L06	Token expirado	Média	Alto	Alto	Implementar refresh token automático.
L07	Formato e-mail inválido	Alta	Baixo	Médio	Validar com regex no submit.
L08	Cabeçalho ausente	Baixa	Alto	Médio	Middleware para verificar headers.
P01	Cadastro autenticado (produto)	Baixa	Alto	Alto	Validar token JWT e permissões.
P02	Sem autenticação	Alta	Alto	Crítico	Exigir token em rotas protegidas.
P03	Nome duplicado (produto)	Média	Médio	Alto	Índice único no BD para nomes.
P06	Listar produtos	Baixa	Baixo	Baixo	Cachear resposta para performance.
P07	Atualizar produto	Média	Alto	Alto	Validar ID e estoque antes do PUT.

P09	Excluir produto	Média	Alto	Alto	Verificar se produto está em carrinhos ativos.
C05	[Cenário não especificado]	-	-	-	Definir critérios claros para o cenário.

Recomendações Gerais:

1. **Autenticação/Token (L06, P02):**
   - Implementar renovação automática de token.
   - Usar middleware para verificar headers em todas as rotas.
2. **Validações de Dados (U02, U03, P03):**
   - Adicionar validações no frontend e backend (ex.: regex para e-mail).
   - Criar constraints no BD para evitar duplicidades.
3. **Segurança (L03, U04):**
   - Limitar tentativas de login e exigir senhas complexas.
4. **Padronização de Métodos (U08):**
   - Restringir criação de recursos apenas ao método POST.

Cobertura de Testes

Módulo	Cobertura Funcional	Cobertura Negativa
Usuários	100%	100%
Login	100%	100%
Produtos	100%	80%
Carrinhos	90%	70%

Anexos

Elaborado por: Alícia Kathleen
Data: 12/05/2025
Versão do Documento: 1.0

Postman: https://martian-meadow-2883510.postman.co/workspace/ALÍCIA's-Workspace~c2bb0329-05a8-423a-8cf3-6097eec88926/collection/44693461-d9366d9a-d654-4e91-bfd8-b2d70d0e8799?action=share&creator=44693461&active-environment=44693461-e109f3ca-b4ea-42b1-9aac-52d089c8d5e6

Report: Relatório de Execução de Testes DRAFT