


Plano de Teste

1. Apresentação

Este plano de testes descreve a abordagem e execução para validar a release 2.29.7 da API  **ServeRest**, garantindo conformidade com os requisitos funcionais e reduzindo riscos de falhas em produção.

2. Objetivo

- Definir a estratégia de testes da release.
 - Garantir que os requisitos funcionais sejam atendidos.
 - Reforçar a confiabilidade do sistema por meio de cenários de borda.
 - Suportar testes automatizados de APIs REST utilizando **Robot Framework**.
-

3. Escopo

• Incluído:

- Login
- Módulo de usuários(Listagem usuários, Cadastro de usuário, Listagem de usuário por id, Exclusão de usuário e edição de usuário)
- Módulo de produtos(Listagem de produtos, Cadastro de produto, Busca de produto por id, Exclusão de produto e Edição de produto)
- Módulo de carrinho(Listagem de carrinhos cadastrado, Cadastro de carrinho, Busca de carrinho por id, Exclusão de carrinho e Exclusão de carrinho e retorno de produtos ao estoque)

• Excluído:

- Testes de performance, carga e estresse
- Testes de segurança avançados
- Integrações externas

• Testes a serem realizados (User Story):

- **US001** – Sendo um vendedor de uma loja, gostaria de poder me cadastrar no Marketplace do ServeRest, para poder realizar as vendas dos meus produtos.

- **US002** – Sendo um vendedor de uma loja com cadastro já realizado, gostaria de poder me autenticar no Marketplace da ServeRest, para poder cadastrar, editar, atualizar e excluir meus produtos.
- **US003** – Sendo um vendedor de uma loja com cadastro já realizado, gostaria de poder me autenticar e cadastrar produtos no Marketplace do ServeRest, para poder cadastrar, editar, atualizar e excluir meus produtos.

4. Análise

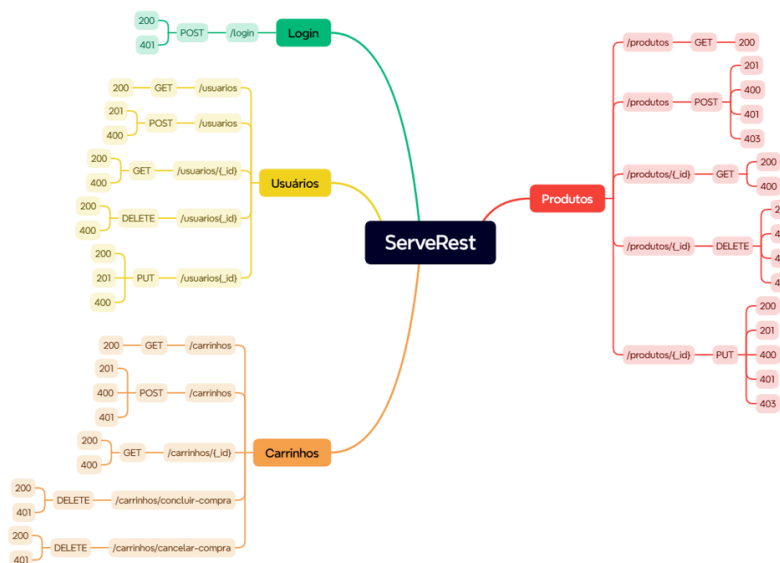
Resumo da análise dos requisitos e riscos técnicos:

- Requisitos funcionais críticos: autenticação, cadastro de produto e carrinho.

5. Técnicas Aplicadas

- Testes Funcionais
- Teste Positivo
- Teste Negativo
- Teste de Limite
- Teste de Robustez

6. Mapa Mental da Aplicação



7. Cenários de Teste Planejados

ID	Módulo	Cenário	Entrada / Payload	Resultado Esperado	Tipo de Teste
CT-01	Login	Login com credenciais válidas	<pre>1 { 2 "email" 3 "password" 4 }</pre>	Retorno 200, token Bearer válido. Mensagem: "Login realizado com sucesso"	Funcional / Positivo
CT-02	Login	Login com senha incorreta	<pre>1 { 2 "email" 3 "password" 4 }</pre>	Retorno 401, Mensagem: "Email e/ou senha inválidos"	Funcional / Negativo
CT-03	Login	Login com campos vazios	<pre>1 { 2 "email" 3 "password" 4 }</pre>	Retorno 400, Mensagem: "email não pode ficar em branco", "password não pode ficar"	Funcional / Negativo

				em branco "	
CT-04	Usuários	Criar usuário válido	<pre> 1 { 2 "nome": 3 "email": 4 "password": 5 "adminis 6 }</pre>	Retorno 201, Mensagem: " Cadastro o realiza do com sucesso "	Funcional / Positivo
CT-05	Usuários	Criar usuário com email inválido	<pre> 1 { 2 "nome": 3 "email": 4 "password": 5 "adminis 6 }</pre>	Retorno 400, Mensagem: " email deve ser um email válido "	Funcional / Negativo
CT-06	Usuários	Criar usuário com payload grande	<pre> 1 { 2 "nome": 3 "email": 4 "password": 5 "adminis 6 }</pre>	Retorno 400	Limite / Robustez
CT-07	Produtos	Listar produtos existentes	GET /produt os	Retorno 200, array de produtos	Funcional / Positivo
CT-08	Produtos	Buscar produto inexistente	GET /produc ts/999	Retorno 404, produto não encontrado	Funcional / Negativo
CT-09	Produtos	Cadastrar Produto	<pre> 1 { 2 "nome" 3 "preco" 4 "descr" 5 "quant 6 }</pre>	Retorno 201, cadastro realizado	Funcional/Po sitivo

				com sucesso	
CT-10	Carrinho	Adicionar produto válido ao carrinho	{ "productId":1, "quantity":2 }	Retorno 200, produto adicionado	Funcional / Positivo
CT-11	Carrinho	Adicionar produto com quantidade inválida	{ "productId":1, "quantity":-1 }	Retorno 400, mensagem de erro	Funcional / Negativo
CT-12	Carrinho	Adicionar produto com payload grande	{ "productId":999999999999, "quantity":999999 }	Retorno 400 ou tratamento de limite	Limite / Robustez
CT-13	Carrinho	Remover produto do carrinho	DELETE /cart/1	Retorno 200, produto removido	Funcional / Positivo
CT-14	Login	Login com formato de e-mail inválido	{ "email":"ana@hotmail.com",	Retorno 400, mensagem: "email deve ser um email válido".	Funcional / Negativo

			<pre>"password": "teste" }</pre>		
CT-15	Produtos	Cadastrar produto com preço negativo	<pre>{ "nome": "teste" , "preco" : -10, "descricao": "inválido", "quantidade": 5 }</pre>	Retorno 400, mensagem: <i>"preço deve ser maior que zero".</i>	Limite / Negativo
CT-16	Carrinho	Tentar adicionar produto inexistente	<pre>{ "productId":99 99, "quantity":1 }</pre>	Retorno 404, mensagem: <i>"Produto não encontrado".</i>	Funcional / Negativo
CT-17	Produtos	Excluir produto presente em carrinho	<pre>DELETE /products/1 (produto já em carrinho)</pre>	Retorno 400, mensagem de erro → estoque deve ser ajustado corretamente.	Integração / Negativo

8. Priorização da Execução dos Cenários de Teste

Prioridade	Cenários	Justificativa
Alta	CT-01, CT-04, CT-09, CT-10, CT-14, CT-15	Fluxos críticos para uso do sistema
Média	CT-07, CT-13, CT-17	Funcionalidade importante, mas não bloqueadora
Baixa	CT-05, CT-06, CT-11, CT-12, CT-16	Cenários alternativos

9. Matriz de Risco

Risco	Impacto	Probabilidade	Severidade
Falha no login	Alto	Baixa	Alto
Erro no cadastro de produto	Alto	Baixa	Alto
Erro ao cadastrar carrinho	Alto	Baixa	Alto
Erro ao cadastrar produto	Alto	Baixa	Alto
Erro ao cadastrar usuário	Alto	Baixa	Alto
Erro ao deletar produto	Médio	Baixa	Baixa
Erro ao deletar usuário	Médio	Baixa	Baixa
Erro ao deletar carrinho	Alto	Baixa	Alto

10. Cobertura de Testes

- Cobertura de requisitos funcionais: **90%**.
 - Automação planejada: **40% dos cenários críticos**
 - Cobertura manual: **60%** para cenários complementares.
-

11. Testes Candidatos à Automação

- **Login:** positivo/negativo (CT-01, CT-02, CT-03, CT-14)
- **Usuários:** criação válida/inválida (CT-04, CT-05, CT-06)
- **Produtos:** cadastro e edição (CT-09, CT-15, CT-17)
- **Carrinho:** adicionar/remover produtos (CT-10, CT-11, CT-12, CT-16, CT-13)

Organizados com **tags do Robot Framework:**

- `critical` → fluxos principais
- `smoke` → validação rápida
- `regression` → suíte completa