


Plano de Testes - API ServeRest (Versão Melhorada)

1. Apresentação

Este documento apresenta o planejamento de testes aprimorado para a API ServeRest ( ServeRest), incorporando as melhorias solicitadas no feedback: ampliação da cobertura para usuários e carrinhos, reforço em mensagens de erro e inclusão de cenários de borda.

Testador: Kethelem da Cruz Socoowski

Período: 01/09/2025 à 08/09/2025

Ambiente:  ServeRest

Ferramentas: Robot Framework, Postman

2. Melhorias Implementadas

2.1 Ampliação para Usuários e Carrinhos

Usuários - Novos cenários adicionados:

- Criação de usuário com nome contendo caracteres especiais
- Email com formatos incomuns (domínio curto, caracteres especiais)
- Exclusão de usuário com dependências
- Validação de limites de campos (nome mínimo/máximo)

Carrinhos - Expansão completa:

- Carrinho vazio
- Carrinho com produto inexistente
- Exclusão de carrinho e retorno de estoque
- Validação de múltiplos produtos em um mesmo carrinho
- Concluir compra e cancelar compra

2.2 Reforço em Mensagens de Erro

Cenários específicos para validar mensagens de erro além dos status codes:

Cenário	Mensagem Esperada
Email duplicado	"Este email já está sendo usado"
Login senha incorreta	"Email e/ou senha inválidos"
Produto em carrinho	"Não é permitido excluir produto que está em um carrinho"
Token expirado	"Token de acesso ausente, inválido, expirado"
Campo obrigatório	"nome é um campo obrigatório"

2.3 Cenários de Borda

Testes de limites:

- Senha com exatamente 5 e 10 caracteres
- Preço de produto = 0 e valores negativos
- Nomes com comprimento mínimo e máximo permitido

Testes de entrada inválida:

- Campos nulos
- Strings com caracteres especiais e emojis
- Requisições com payload muito grande

2.4 Automação com Robot Framework

Keywords planejadas:

```
1  *** Keywords ***
2  Realizar Login Valido
3      [Arguments]    ${email}    ${password}
4      ${login_data}=    Create Dictionary    email=${email}
5      password=${password}
6      ${response}=    POST    ${BASE_URL}/login    json=${login_data}
7      Should Be Equal As Numbers    ${response.status_code}    200
8      [Return]    ${response.json()[authorization]}
9
10 Validar Mensagem Erro
11     [Arguments]    ${response}    ${mensagem_esperada}
12     Should Contain    ${response.json()[message]}
13     ${mensagem_esperada}
14
15 Criar Usuario E Retornar ID
16     [Arguments]    ${nome}    ${email}    ${password}
17     ${admin}=false
18     ${user_data}=    Create Dictionary    nome=${nome}
19     email=${email}    password=${password}    administrador=${admin}
20     ${response}=    POST    ${BASE_URL}/usuarios    json=${user_data}
21     Should Be Equal As Numbers    ${response.status_code}    201
22     [Return]    ${response.json()[_id]}
```

3. Cenários de Teste - Resumo Expandido

3.1 Usuários (30 cenários - ampliado de 25)

Novos cenários críticos:

- TC011: Nome com acentos e caracteres especiais
- TC022: Email com domínio inválido (.c, .toolongdomain)
- TC033: Senha exatamente no limite (5 e 10 chars)
- TC034: Nome com emojis
- TC035: Validar mensagem específica email duplicado

3.2 Login (15 cenários - ampliado de 12)

Novos cenários:

- TC026: Múltiplas tentativas login inválido
- TC027: Token malformatado

- TC028: Validar claims JWT

3.3 Produtos (20 cenários - mantido com melhorias)

Cenários de borda adicionados:

- TC052: Preço exatamente R\$ 0,00
- TC053: Preço negativo (-10)
- TC054: Nome produto com 500+ caracteres

3.4 Carrinhos (15 cenários - NOVO módulo completo)

Cenários principais:

- TC070: Listar carrinhos
 - TC071: Criar carrinho com produtos válidos
 - TC072: Carrinho vazio (sem produtos)
 - TC073: Produto inexistente no carrinho
 - TC074: Múltiplos produtos mesmo carrinho
 - TC075: Excluir carrinho - retornar estoque
 - TC076: Concluir compra
 - TC077: Cancelar compra
 - TC078: Validar estrutura carrinho
 - TC079: Carrinho usuário não logado
 - TC080: Produto sem estoque suficiente
 - TC081: Quantidade negativa no carrinho
 - TC082: Validar total carrinho
 - TC083: Carrinho com produto deletado
 - TC084: Operações simultâneas carrinho
-

4. Priorização Revisada

4.1 Alta Prioridade (35 cenários)

Fluxos críticos:

- CRUD usuários básico
- Login válido e token
- Criação produto autenticado
- Carrinho: criar, listar, excluir

4.2 Média Prioridade (25 cenários)

Regras de negócio:

- Validações unicidade (email, nome produto)
- Mensagens de erro específicas
- Validações token expirado/inválido

4.3 Baixa Prioridade (20 cenários)

Borda e edge cases:

- Caracteres especiais
- Limites de campos
- Payload muito grande

5. Automação Robot Framework

5.1 Cenários Seleccionados (40 cenários)

Distribuição por módulo:

- Usuários: 15 cenários
- Login: 8 cenários
- Produtos: 10 cenários
- Carrinhos: 7 cenários

5.2 Estrutura de Testes

```
1  *** Test Cases ***
2  TC001 - Criar Usuario Valido
3      [Tags]    usuarios    positivo    alta_prioridade
4      ${user_id}=    Criar Usuario E Retornar ID    João Silva
5      joao@teste.com    senha123    true
6      Should Not Be Empty    ${user_id}
7
8  TC011 - Usuario Nome Caracteres Especiais
9      [Tags]    usuarios    borda    media_prioridade
10     ${user_data}=    Create Dictionary    nome=José Maria
11     email=jose@teste.com    password=senha123    administrador=false
12     ${response}=    POST    ${BASE_URL}/usuarios    json=${user_data}
13     Should Be Equal As Numbers    ${response.status_code}    201
14
15  TC022 - Validar Erro Email Duplicado
16     [Tags]    usuarios    negativo    mensagem_erro
17     ${user_data}=    Create Dictionary    nome=Teste
18     email=duplicado@teste.com    password=senha123    administrador=false
19     # Primeiro cadastro
20     POST    ${BASE_URL}/usuarios    json=${user_data}
21     # Segundo cadastro - deve falhar
22     ${response}=    POST    ${BASE_URL}/usuarios    json=${user_data}
23     Should Be Equal As Numbers    ${response.status_code}    400
24     Validar Mensagem Erro    ${response}    Este email já está sendo
25     usado
26
27  TC071 - Criar Carrinho Produtos Validos
28     [Tags]    carrinhos    positivo    alta_prioridade
29     ${token}=    Realizar Login Valido    admin@teste.com    senha123
30     ${headers}=    Create Dictionary    Authorization=${token}
31     ${produto_id}=    Criar Produto E Retornar ID    Notebook    2500
32     10    Notebook Dell
33     ${carrinho_data}=    Create Dictionary    produtos=[{"idProduto":
34     "${produto_id}", "quantidade": 2}]
35     ${response}=    POST    ${BASE_URL}/carrinhos
36     json=${carrinho_data}    headers=${headers}
37     Should Be Equal As Numbers    ${response.status_code}    201
38
39  31
```

6. Cobertura Final

6.1 Comparativo de Cenários

Módulo	Original	Melhorado	Aumento
Usuários	25	30	+5 cenários
Login	12	15	+3 cenários
Produtos	20	20	melhorias qualitativas
Carrinhos	5	15	+10 cenários
TOTAL	62	80	+29% cobertura

6.2 Cobertura por Tipo

- **Cenários Positivos:** 30 cenários
- **Cenários Negativos:** 35 cenários
- **Cenários de Borda:** 15 cenários
- **Automação RF:** 40 cenários

7. Cronograma

Fase	Atividade	Duração
1	Setup + Cenários Alta Prioridade	2 dias
2	Cenários Média Prioridade + Carrinhos	2 dias
3	Cenários de Borda + Automação	2 dias
4	Validação + Documentação	2 dias
Total		8 dias

8. Entregáveis

1. **Plano de Testes Melhorado** (este documento)
2. **Suite Robot Framework** com 40 cenários automatizados
3. **Collection Postman** com 80 cenários
4. **Relatório de Execução** com evidências
5. **Lista de Bugs** por severidade
6. **Apresentação** (4 min) com resultados

Resumo das Melhorias Aplicadas:

- ✓ **Ampliação Usuários e Carrinhos** - Usuários +5 cenários, Carrinhos +10 cenários
- ✓ **Reforço Mensagens de Erro** - 15 validações específicas mapeadas
- ✓ **Cenários de Borda** - 15 cenários cobrindo limites e edge cases
- ✓ **Automação Robot Framework** - 40 cenários com keywords reutilizáveis
- ✓ **Priorização Revisada** - Foco em impacto no negócio

Documento preparado por: Kethelem da Cruz Socoowski

Data: 26/09/2025

Versão: 2.0