# Challenge 01 - Plano de Testes (ServeRest)

### Plano de Testes 🖉

Software: API Serverest

QA Responsável: Karen Késsia Herrmann

Data de Início: 05/05

Data Estimada de Término: 09/05

### 1. Apresentação 🖉

"Este plano de testes visa validar as funcionalidades da API ServeRest — cadastro de usuários, login e produtos — conforme as regras de negócio definidas nas User Stories (US001 a US003) e documentação do Swagger."

### 2. Objetivo 🖉

Garantir que a API ServeRest esteja conforme os critérios definidos, segura, resistente a dados inválidos e com comportamento adequado em todos os cenários.

### 3. Escopo @

### Escopo incluso: @

- Módulo de Usuários (US001).
- Módulo de Login (US002).
- Módulo de Produtos (US003).
- Verificação de regras de negócio, como e-mails válidos, senhas, duplicidade, token, etc.

### Escopo excluído: @

- Testes de Front-End
- Integração com API's Externas
- Testes de carga, performance e segurança avançada.
- Testes Seção Carrinhos

### 4. Análise de Testes 🖉

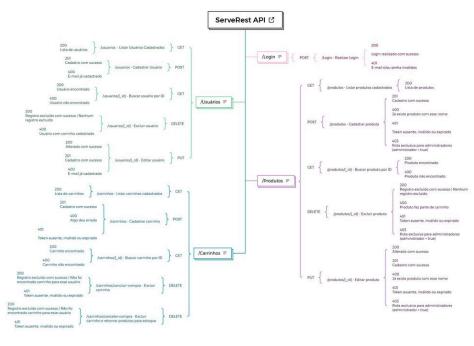
Análise foi feita a partir de:

- Swagger oficial da API: ServeRest
- User Stories fornecidas (US001 a US003).
- Heurísticas como análise do Valor Limite, Teste Baseado em Risco, Exploratórios Orientados a Regras e análise por partições de equivalência.

### 5. Técnicas Aplicadas ${\mathscr O}$

- Análise de Partições de Equivalência
- Valor Limite
- CRUD
- Teste Baseado em Riscos
- Testes Exploratórios
- Teste Baseado em Regras de Negócio

### 6. Mapa Mental da Aplicação 🖉



Presented with xmind A

## Users Storys e Critérios de Aceitação 🔗

### **User Stories** *⊘*

## US 001 - [API] Usuários 🖉

Sendo um vendedor de uma loja

Gostaria de poder me cadastrar no Marketplace do ServeRest

Para poder realizar as vendas dos meus produtos

### DoR

- Banco de dados e infraestrutura para desenvolvimento disponibilizados;
- Ambiente de testes disponibilizado.

### DoD

- CRUD de cadastro de vendedores (usuários) implementado (CRIAR, ATUALIZAR, LISTAR E DELETAR);
- Análise de testes cobrindo todos verbos;
- Matriz de rastreabilidade atualizada;
- Automação de testes baseado na análise realizada;

### **Acceptance Criteria**

- Os vendedores (usuários) deverão possuir os campos NOME, E-MAIL, PASSWORD e ADMINISTRADOR;
- Não deverá ser possível fazer ações e chamadas para usuários inexistentes;
- Não deve ser possível criar um usuário com e-mail já utilizado;
- Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado;
- Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT;
- Os testes executados deverão conter evidências;
- Não deverá ser possível cadastrar usuários com e-mails de provedor gmail e hotmail;
- Os e-mails devem seguir um padrão válido de e-mail para o cadastro;
- As senhas devem possuír no mínimo 5 caracteres e no máximo 10 caracteres;
- A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos.

## US 002: [API] Login *⊘*

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

#### DoR

- Banco de dados e infraestrutura para desenvolvimento disponibilizados;
- API de cadastro de usuários implementada;
- Ambiente de testes disponibilizado.

### DoD

- Autenticação com geração de token Bearer implementada;
- Análise de testes cobrindo a rota de login;
- Matriz de rastreabilidade atualizada;
- Automação de testes baseado na análise realizada;

### **Acceptance Criteria**

- Usuários não cadastrados não deverão conseguir autenticar;
- Usuários com senha inválida não deverão conseguir autenticar;
- No caso de não autenticação, deverá ser retornado um status code 401 (Unauthorized);
- Usuários existentes e com a senha correta deverão ser autenticados;
- A autenticação deverá gerar um token Bearer;
- A duração da validade do token deverá ser de 10 minutos;
- Os testes executados deverão conter evidências;
- A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos.

### US 003: [API] Produtos €

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

### DoR

- Banco de dados e infraestrutura para desenvolvimento disponibilizados;
- API de cadastro de usuários implementada;
- API de autenticação implementada;
- Ambiente de testes disponibilizado.

### DoD

- CRUD de cadastro de Produtos implementado (CRIAR, ATUALIZAR, LISTAR E DELETAR);
- Análise de testes cobrindo a rota de produtos;
- Matriz de rastreabilidade atualizada;
- Automação de testes baseado na análise realizada;

### **Acceptance Criteria**

- Usuários não autenticados não devem conseguir realizar ações na rota de Produtos;
- Não deve ser possível realizar o cadastro de produtos com nomes já utilizados;
- Não deve ser possível excluir produtos que estão dentro de carrinhos (dependência API Carrinhos);
- Caso não exista produto com o o ID informado na hora do UPDATE, um novo produto deverá ser criado;
- Produtos criados através do PUT não poderão ter nomes previamente cadastrados;
- Os testes executados deverão conter evidências;
- A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos.

### 7. Cenários de Teste Planejados 🔗

### Planned Scenarios

| ID    | Endpoint       | Cenário                     | Tipo     | Resultado<br>Esperado | Prioridade | Estratégia              |
|-------|----------------|-----------------------------|----------|-----------------------|------------|-------------------------|
| TC001 | POST /usuarios | Criar usuário<br>válido     | Positivo | 201 Created           | Alta       | Fluxo principal         |
| TC002 | POST /usuarios | E-mail<br>duplicado         | Negativo | 400                   | Alta       | Regra de<br>unicidade   |
| TC003 | POST /usuarios | E-mail sem @                | Negativo | 400                   | Média      | Validação de<br>formato |
| TC004 | POST /usuarios | E-mail com<br>domínio Gmail | Negativo | 400                   | Alta       | Regra de<br>negócio     |
| TC005 | POST /usuarios | Senha com 4 caracteres      | Negativo | 400                   | Alta       | Valor-limite inferior   |
| TC006 | POST /usuarios | Senha com 5 caracteres      | Positivo | 201                   | Alta       | Valor-limite<br>válido  |
| TC007 | POST /usuarios | Senha com 10 caracteres     | Positivo | 201                   | Alta       | Valor-limite<br>válido  |
| TC008 | POST /usuarios | Senha com 11<br>caracteres  | Negativo | 400                   | Alta       | Valor-limite superior   |

| TC009 | PUT /usuarios/{id}                    | Atualizar<br>usuário<br>inexistente<br>(upsert) | Positivo | 201 Created    | Alta  | Upsert                   |
|-------|---------------------------------------|---|----------|----------------|-------|--------------------------|
| TC010 | PUT<br>/usuarios/{id}                 | Atualizar com<br>e-mail de outro<br>usuário     | Negativo | 400            | Alta  | Regra de<br>negócio      |
| TC011 | PUT<br>/usuarios/{id}                 | Atualizar senha<br>com +10<br>caracteres        | Negativo | 400            | Média | Valor-limite             |
| TC012 | DELETE /usuarios/{id}                 | Deletar usuário<br>válido                       | Positivo | 200 OK         | Média | Cobertura<br>CRUD        |
| TC013 | DELETE /usuarios/{id}                 | Deletar usuário inexistente                     | Negativo | 404            | Baixa | Erro esperado            |
| TC014 | POST /login                           | Login com<br>usuário válido                     | Positivo | 200 OK + token | Alta  | Fluxo principal          |
| TC015 | POST /login                           | Login com<br>usuário<br>inexistente             | Negativo | 400            | Alta  | Autenticação<br>negativa |
| TC016 | POST /login                           | Login com<br>senha incorreta                    | Negativo | 401            | Alta  | Autenticação negativa    |
| TC017 | POST /login                           | Login com<br>senha correta e<br>token expirando | Positivo | 200 + token    | Alta  | Segurança                |
| TC018 | GET /produtos<br>(token<br>expirado)  | Token vencido<br>após 10 minutos                | Negativo | 401            | Alta  | Teste de sessão          |
| TC019 | POST /produtos<br>(token<br>expirado) | Criar com token<br>vencido                      | Negativo | 401            | Alta  | Segurança/toke<br>n      |
| TC020 | POST /produtos                        | Criar com token<br>válido                       | Positivo | 201 Created    | Alta  | Fluxo principal          |
| TC021 | POST /produtos                        | Criar com token inválido                        | Negativo | 401            | Alta  | Autenticação             |
| TC022 | POST /produtos                        | Criar com nome<br>duplicado                     | Negativo | 400            | Alta  | Regra de<br>unicidade    |
| TC023 | POST /produtos                        | Criar sem token                                 | Negativo | 401            | Alta  | Segurança                |
| TC024 | GET /produtos                         | Listar todos<br>produtos                        | Positivo | 200            | Média | Fluxo comum              |
| TC025 | DELETE /produtos/{id}                 | Deletar produto<br>válido                       | Positivo | 200            | Média | CRUD                     |

| TC026 | DELETE /produtos/{id} | Produto em carrinho             | Negativo | 400 | Alta  | Regra de<br>negócio   |
|-------|-----------------------|---------------------------------|----------|-----|-------|-----------------------|
| TC027 | DELETE /produtos/{id} | Produto inexistente             | Negativo | 404 | Baixa | Erro esperado         |
| TC028 | PUT /produtos/{id}    | Atualizar<br>produto válido     | Positivo | 200 | Alta  | Fluxo comum           |
| TC029 | PUT<br>/produtos/{id} | Atualizar inexistente (upsert)  | Positivo | 201 | Alta  | Upsert                |
| TC030 | PUT<br>/produtos/{id} | Atualizar com<br>nome duplicado | Negativo | 400 | Alta  | Regra de<br>unicidade |
| TC031 | PUT<br>/produtos/{id} | Atualizar sem<br>token          | Negativo | 401 | Alta  | Segurança             |
| TC032 | GET /usuarios         | Listar todos<br>usuários        | Positivo | 200 | Baixa | Complementar          |

Total de testes planejados: 32 cenários

Testes Executados  ${\mathscr O}$ 

(/usuarios) 🖉

| Story ID | Cenário   | Endpoint          | Pré-<br>Condição  | Esperado           | Obtido<br>(Resposta) | Pós-<br>condição                        |
|----------|---|-------------------|---|--------------------|----------------------|---|
| US001    | Criar<br>usuário com<br>dados<br>válidos  | POST<br>/usuarios | Nenhum<br>usuário com<br>o mesmo e-<br>mail<br>cadastrado | 201 Created        | 201 Created          | Usuário<br>registrado<br>com<br>sucesso |
| US001    | Criar<br>usuário com<br>e-mail<br>duplicado                                     | POST<br>/usuarios | E-mail já<br>cadastrado<br>previamente                    | 400 Bad<br>Request | 400 Bad<br>Request   | Usuário não<br>é criado                 |
| US001    | Criar<br>usuário com<br>domínio de<br>e-mail<br>inválido<br>(Gmail/Hot<br>mail) | POST<br>/usuarios | E-mail com<br>domínio<br>@gmail.co<br>m ou<br>@hotmail.c  | 400 Bad<br>Request | 400 Bad<br>Request   | Usuário não<br>é criado                 |

| US001 | Criar<br>usuário com<br>senha<br>menor que<br>5<br>caracteres | POST<br>/usuarios | Senha com<br>menos de 5<br>caracteres     | 400 Bad<br>Request | 400 Bad<br>Request | Usuário não<br>é criado                      |
|-------|---|-------------------|---|--------------------|--------------------|--|
| US001 | Criar<br>usuário com<br>senha maior<br>que 10<br>caracteres   | POST<br>/usuarios | Senha com<br>mais de 10<br>caracteres     | 400 Bad<br>Request | 400 Bad<br>Request | Usuário não<br>é criado                      |
| US001 | Atualizar<br>usuário com<br>ID<br>inexistente                 | PUT<br>/usuarios  | ID inexistente                            | 201 Created        | 201 Created        | Novo usuário criado com base no ID informado |
| US001 | Atualizar<br>usuário com<br>e-mail já<br>utilizado            | PUT<br>/usuarios  | Outro usuário com mesmo e- mail existente | 400 Bad<br>Request | 400 Bad<br>Request | Atualização<br>rejeitada                     |

# (/login)

Ø

| Story ID | Cenário  | Endpoint               | Pré-<br>Condição                              | Esperado                | Obtido<br>(Resposta)              | Pós-<br>condição                                     |
|----------|--|------------------------|---|-------------------------|-----------------------------------|--|
| US002    | Login com<br>usuário<br>inexistente                    | POST /login            | Usuário não<br>cadastrado                     | 401<br>Unauthorize<br>d | 401<br>Unauthorize<br>d           | Acesso não permitido                                 |
| US002    | Login com<br>senha<br>inválida                         | POST /login            | Usuário<br>existente,<br>senha<br>incorreta   | 401<br>Unauthorize<br>d | 401<br>Unauthorize<br>d           | Acesso não permitido                                 |
| US002    | Login com<br>usuário<br>válido                         | POST /login            | Usuário<br>cadastrado<br>com senha<br>correta | 200 OK                  | 200 OK                            | Token<br>Bearer<br>gerado                            |
| US002    | Login com<br>token<br>expirado em<br>rota<br>protegida | GET /rota<br>protegida | Token<br>expirado                             | 401<br>Unauthorize<br>d | 200 OK ( ! Validaçã o necessária) | Acesso<br>indevido<br>permitido<br>(possível<br>bug) |

| Story ID | Cenário  | Endpoint            | Pré-<br>Condição                                | Esperado            | Obtido<br>(Resposta) | Pós-<br>condição  |
|----------|--|---------------------|---|---------------------|----------------------|---|
| US003    | Criar produto<br>autenticado                       | POST<br>/produtos   | Usuário<br>autenticado<br>com token<br>válido   | 201 Created         | 401<br>Unauthorized  | Produto não é criado (Possível falha na autenticação (token inválido ou mal formatado). |
| US003    | Criar produto<br>com nome<br>duplicado             | POST<br>/produtos   | Produto com<br>mesmo nome<br>já cadastrado      | 400 Bad<br>Request  | 401<br>Unauthorized  | Produto não<br>criado<br>(validação de<br>autenticação<br>falhou)                       |
| US003    | Deletar<br>produto<br>vinculado a<br>carrinho      | DELETE<br>/produtos | Produto<br>associado a<br>um carrinho           | 400 Bad<br>Request  | 401<br>Unauthorized  | Produto não<br>deletado<br>(validação de<br>autenticação<br>falhou)                     |
| US003    | Criar produto<br>sem token                         | POST<br>/produtos   | Nenhum<br>token<br>fornecido                    | 401<br>Unauthorized | 401<br>Unauthorized  | Produto não é<br>criado   |
| US003    | Atualizar<br>produto<br>inexistente                | PUT<br>/produtos    | Produto com ID inexistente                      | 201 Created         | 401<br>Unauthorized  | Produto não<br>atualizado<br>(validação de<br>autenticação<br>falhou)                   |
| US003    | Atualizar<br>produto com<br>quantidade<br>negativa | PUT<br>/produtos    | Produto<br>existente,<br>quantidade<br>negativa | 400 Bad<br>Request  | 401<br>Unauthorized  | Produto não<br>atualizado<br>(autenticação<br>falhou antes<br>da validação)             |

Total de testes Executados: 17 cenários

# 9. Matriz de Risco ${\mathscr O}$

| Risco ID | TCID  | Descrição do<br>Risco                 | Impacto | Probabilidade | Nível de Risco |
|----------|-------|---------------------------------------|---------|---------------|----------------|
| R01      | TC002 | Permitir<br>criação de<br>usuário com | Alto    | Alta          | Crítico        |

|     |                  | e-mail<br>duplicado  |         |       |         |
|-----|------------------|--|---------|-------|---------|
| R02 | TC003            | Cadastro com<br>domínio de e-<br>mail inválido<br>permitido<br>(gmail/hotmail<br>) | Médio   | Alta  | Alto    |
| R03 | TC004 /<br>TC005 | Cadastro com<br>senha fora do<br>padrão (curta<br>ou longa)                        | Médio   | Alta  | Alto    |
| RO4 | TC006            | PUT cria usuário ao invés de atualizar quando ID não existe                        | Alto    | Média | Alto    |
| R05 | TC007            | PUT aceita e-<br>mail já<br>existente  | Alto    | Alta  | Crítico |
| R06 | TC008            | Login com usuário inexistente não retorna erro adequado                            | Médio   | Alta  | Alto    |
| R07 | TC009            | Login com<br>senha inválida<br>não retorna<br>erro adequado                        | Alto    | Alta  | Crítico |
| R08 | TC010            | Falha ao gerar<br>token de login<br>válido   | Alto    | Média | Alto    |
| R09 | TC011            | Token expirado permite acesso indevido   | Crítico | Média | Crítico |
| R10 | TC012            | Produto não<br>pode ser<br>criado mesmo<br>com token<br>válido                     | Alto    | Média | Alto    |
| R11 | TC013            | Produto com<br>nome  | Alto    | Alta  | Crítico |

|     |       | duplicado é<br>criado                                      |         |       |         |
|-----|-------|--|---------|-------|---------|
| R12 | TC014 | Produto é excluído mesmo estando em carrinho               | Crítico | Média | Crítico |
| R13 | TC015 | Produto é<br>criado sem<br>token                           | Alto    | Alta  | Crítico |
| R14 | TC016 | PUT cria<br>produto com<br>ID inexistente<br>sem validação | Alto    | Média | Alto    |
| R15 | TC017 | Atualizar<br>produto com<br>quantidade<br>negativa         | Médio   | Média | Médio   |

### Cobertura de Testes Manual 🖉

Dos **32 cenários planejados**, **17 foram executados**, representando uma **cobertura de 53**% das funcionalidades mapeadas com base nas user stories, critérios de aceitação e Swagger da API.

A priorização focou em:

- Fluxos principais (happy path)
- Regras de negócio críticas (ex: domínio de e-mail, senhas inválidas, unicidade)
- Testes negativos de segurança (ex: token expirado ou ausente)
- Upserts relevantes (ex: PUT com ID inexistente)

Testes de menor risco ou menor impacto foram documentados, mas não executados por questões de objetividade e escopo inicial. A execução futura poderá complementar a cobertura para atingir 100%.

| Total Planejados | Executados | Cobertura % | Foco da Execução   |
|------------------|------------|-------------|--|
| 32               | 17         | 53%         | Fluxo feliz, segurança,<br>regras críticas,<br>negativos |

# 10. Testes Candidatos à Automação ${\mathscr O}$

- Login com credenciais válidas e inválidas
- Cadastro de usuário com e-mail duplicado
- Exclusão de produto em carrinho
- Verificação de token expirado
- CRUD completo de Produtos com validações

# 11. Entregáveis 🖉

- Compilado (Resultados, Issues e Melhorias)
- Collection (ServeRest API):



Serverest API.... 1).json 11 mai. 2025, 11:44 PM