Emanuel - Challenge ServeRest

Plano de Testes – ServeRest API

1) Apresentação

Este plano descreve a estratégia completa para garantir a qualidade das rotas da API ServeRest, cobrindo planejamento, análise baseada no Swagger e nas User Stories, técnicas de teste, cenários priorizados, matriz de risco, cobertura, automação (Postman) e critérios de reporte.

2) Pessoas Envolvidas:

- Emanuel Felipe Avelino Silva

3) Objetivo

Assegurar que as regras de negócio e comportamentos descritos nas US sejam atendidos e que as rotas de **Usuários**, **Login** e **Produtos** funcionem com confiabilidade, segurança e consistência, incluindo fluxos positivos, negativos e alternativos.

4) Escopo

4.1 Dentro do escopo

Neste plano, vamos testar apenas as partes principais da API que garantem o funcionamento correto do sistema. Inclui:

- Rotas REST principais: /usuarios, /login, /produtos (e dependência consultiva de /carrinhos).
- Verbos: **POST, GET, PUT, DELETE** (CRUD completo onde aplicável).
  - **Usuários:** criar, consultar, atualizar e excluir vendedores.
  - **Login:** autenticação de usuários com geração de token.
  - **Produtos:** cadastrar, consultar, atualizar e excluir produtos.
- **Regras de negócio importantes:**
  - Não permitir e-mails repetidos ou de provedores proibidos (gmail, hotmail).
  - Senhas com tamanho mínimo e máximo.
  - Garantir que produtos tenham nomes únicos.
  - Apenas usuários autenticados podem mexer em produtos.
- **Mensagens e respostas da API:** verificar se a API responde corretamente, mostrando status de sucesso ou erro.

5) Fora do escopo:

- Integrações externas, testes de performance, testes de carga, testes de segurança avançados.

6) Análise pelas user storys

US 001 – Usuários

Critérios principais:

- Campos obrigatórios: nome, email, password, administrador.
- E-mail não pode repetir; **provedores gmail/hotmail proibidos**; e-mail deve seguir padrão válido.
- PUT com ID inexistente deve **criar** (upsert), **mas** não pode violar unicidade de e-mail.
- Ações com usuários inexistentes **não** devem ser permitidas.
- Senha: min 5 e max 10 caracteres.

US 002 – Login

Critérios principais:

- Não autenticar usuário inexistente ou com senha inválida → **401**.
- Autenticar usuário válido → retorna **token Bearer**.
- Token válido por **10 minutos**.

US 003 – Produtos

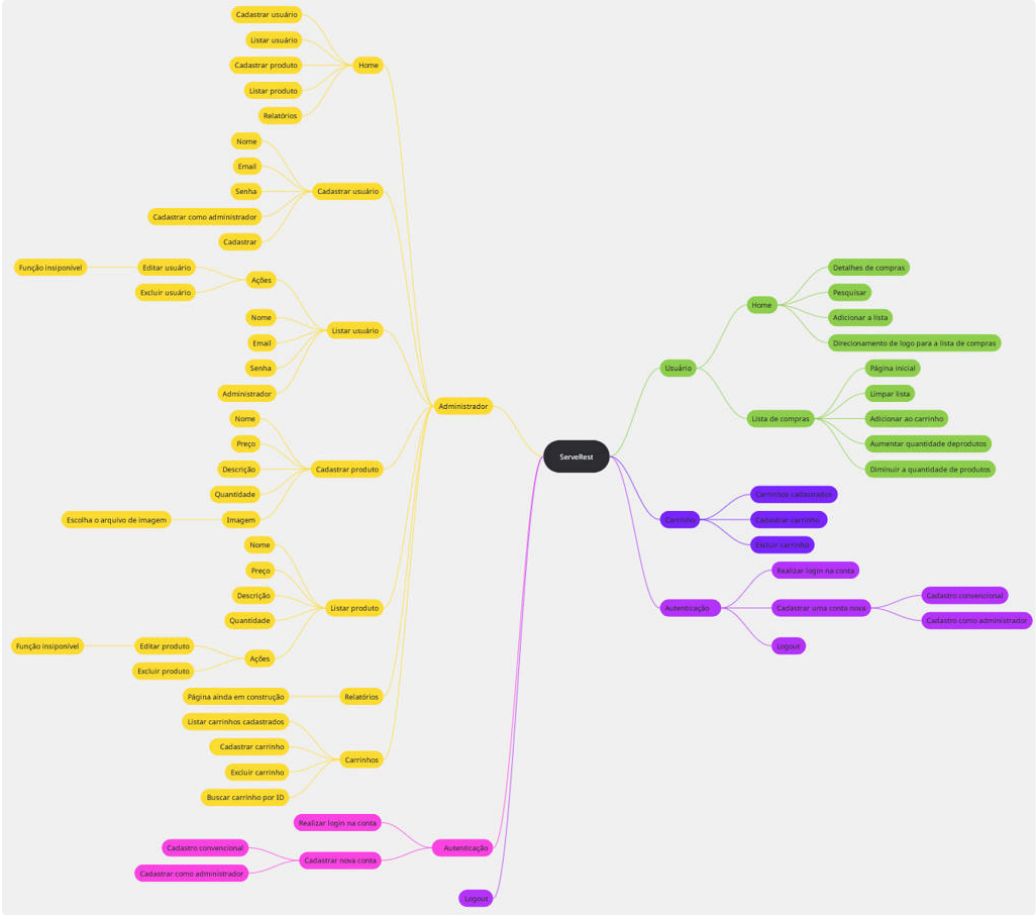
Critérios principais:

- Necessita autenticação para qualquer ação.
- Nome de produto **deve ser único** (POST e PUT-upsert).
- Não permitir excluir produtos dentro de carrinhos.
- PUT com ID inexistente deve **criar** novo produto, respeitando unicidade do nome.

7) Técnicas aplicadas

- Testes de API.
- Testes de automação.

8) Mapa mental da aplicação



9. Cenários de Teste Planejados

Usuários

- CT-01: Cadastrar usuário válido → Esperado: 201
- CT-02: Cadastrar usuário duplicado → Esperado: 400
- CT-03: Consultar usuário inexistente → Esperado: 404
- CT-04: Editar usuário inexistente → Esperado: 201

Login

- CT-05: Login com sucesso → Esperado: 200 + token
- CT-06: Login com senha incorreta → Esperado: 401
- CT-07: Login com usuário inexistente → Esperado: 401

Produtos

- CT-08: Cadastrar produto válido (com autenticação) → Esperado: 201
- CT-09: Cadastrar produto sem autenticação → Esperado: 401
- CT-10: Cadastrar produto duplicado → Esperado: 400

- CT-11: Editar produto inexistente → Esperado: 201

10) Priorização da Execução

- **Alta prioridade:** Testes de Login e cadastro de usuários/produtos.
- **Média prioridade:** Edição de dados.
- **Baixa prioridade:** Consultas a registros inexistentes.

11) Matriz de Risco

A matriz de risco tem como objetivo identificar, avaliar e priorizar possíveis falhas que podem ocorrer durante a utilização da API. Os riscos foram classificados de acordo com:

- **Impacto:** efeito que o risco causa no sistema/usuário (Baixo, Médio, Alto).
- **Probabilidade:** chance de o risco ocorrer (Baixa, Média, Alta).
- **Nível de Risco:** combinação entre impacto e probabilidade (Baixo, Moderado, Crítico).

Tabela de Riscos

Risco	Impacto	Probabilidade	Nível	Justificativa
Falha no login (usuário não conseguir acessar)	Alto	Alto	**Crítico**	Sem autenticação, o usuário não acessa os recursos da API, comprometendo todo o uso.
Cadastro de usuário duplicado	Médio	Médio	**Moderado**	Pode gerar inconsistências no banco de dados e confusão na gestão de contas.
Cadastro de produto	Médio	Médio	**Moderado**	Pode afetar relatórios de

duplicado				estoque e integridade dos dados.
Produto/usuário inexistente sendo editado	Baixo	Médio	**Baixo**	Ocorre em situações pontuais, não afeta a aplicação globalmente.
Acesso a endpoint protegido sem autenticação	Alto	Médio	**Alto**	Se não tratado corretamente, pode comprometer a segurança dos dados.
Erro em mensagens de validação (campos obrigatórios ausentes, limites inválidos, etc.)	Médio	Alto	**Alto**	Impacta diretamente a usabilidade, já que o usuário não entende porque a requisição falhou.

12) Cobertura de Testes

A cobertura definida contempla os principais fluxos da API, abrangendo:

- **Operações CRUD** de usuários e produtos;
- **Autenticação** via login;
- **Validação de códigos de resposta (200, 201, 400, 401, 404)**.

13) Testes Candidatos à Automação

Foram identificados alguns cenários que poderão ser automatizados no futuro para garantir maior velocidade, confiabilidade e repetibilidade nas execuções.

Os principais candidatos à automação são:

- **Login**
  - Validar login com credenciais corretas, garantindo que o token seja gerado corretamente.
  - Testar tentativas de login com senha inválida ou usuário inexistente, verificando o retorno esperado (401 Unauthorized).
- **Usuários**
  - Automatizar o cadastro de usuário válido, validando o retorno 201 e a persistência dos dados.
  - Testar automaticamente cadastros duplicados, garantindo que a API retorne 400.
  - Verificar senhas dentro e fora do limite permitido.
- **Produtos**
  - Automatizar o cadastro de produto válido e checar o ID retornado.
  - Testar cadastros duplicados de produto, assegurando rejeição correta.
  - Automatizar exclusão de produto válido e tentativa de exclusão de produto inexistente.
- **Carrinhos**
  - Automatizar a criação de carrinho com produto válido e validar o controle de estoque.
  - Testar automaticamente as ações de concluir e cancelar compras, verificando as mensagens esperadas.