

## Plano de Testes – ServeRest API

### 1. Apresentação

Este plano revisado descreve a estratégia de testes da API **ServeRest**, consolidando o planejamento inicial e as melhorias encontradas durante a execução do Challenge. O documento foi ajustado após feedback dos instrutores, incorporando:

- Testes de limite (senhas e provedores de e-mail).
- Análise crítica das mensagens de erro.
- Detalhamento das evidências.
- Alinhamento entre o planejamento inicial e os testes efetivamente executados.

### 2. Pessoas Envolvidas

- **Responsável:** Emanuel Felipe Avelino Silva

### 3. Objetivo

Garantir que as regras de negócio da API sejam cumpridas, validando fluxos principais, alternativos e cenários de erro, com foco em qualidade, clareza das mensagens retornadas e robustez.

### 4. Escopo

#### 4.1 Dentro do escopo

- Rotas: /usuarios, /login, /produtos, /carrinhos.
- Operações: POST, GET, PUT, DELETE (CRUD completo quando aplicável).
- Regras de negócio:
  - Unicidade de e-mails.
  - Restrição de provedores (gmail/hotmail).
  - Validação de senhas (mínimo 5, máximo 10 caracteres).
  - Apenas usuários autenticados e administradores podem gerenciar produtos.
  - Um usuário pode ter apenas um carrinho ativo.

#### 4.2 Fora do escopo

- Integrações externas.
- Testes de carga, stress e performance.
- Testes de segurança avançados.

### 5. Estratégia de Testes

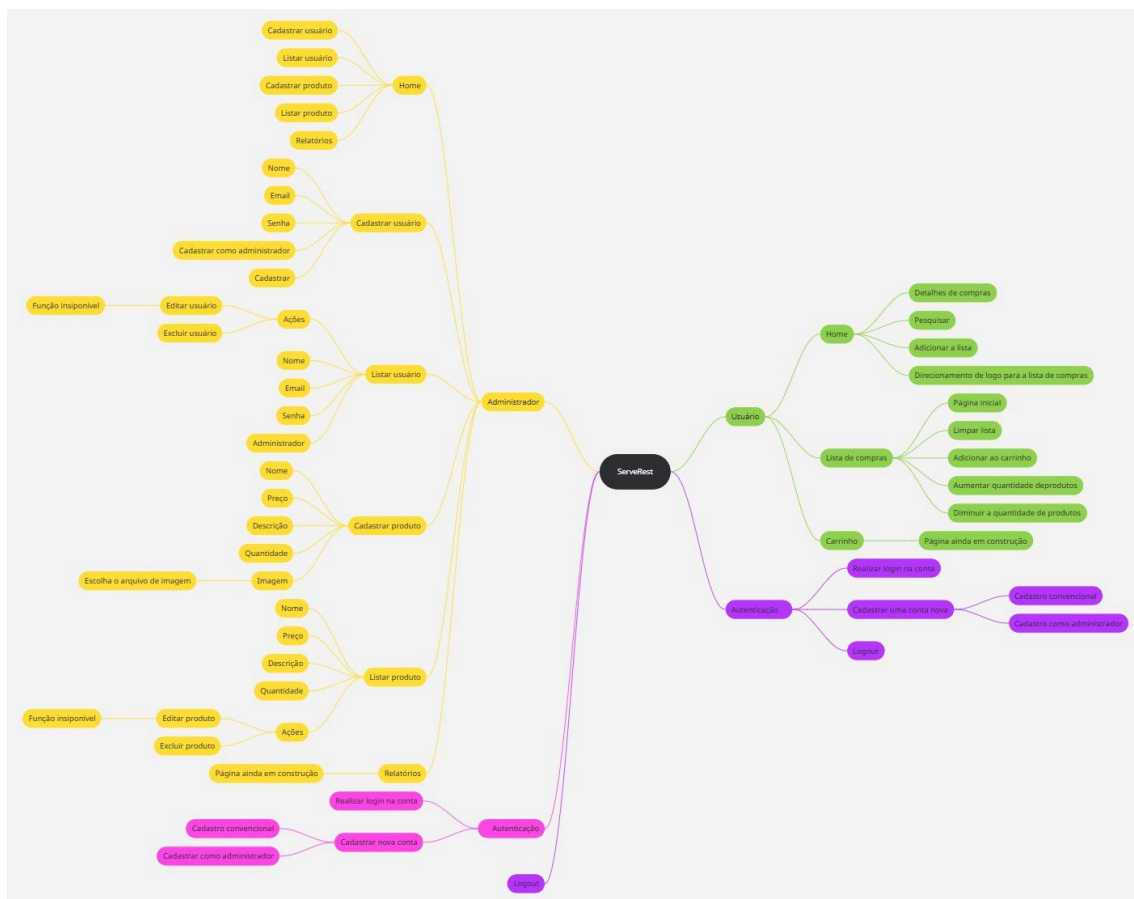
1. **Execução manual exploratória** com Postman para validar regras de negócio.

2. **Execução estruturada** com QALity (Jira), registrando cenários planejados, resultados e bugs.
3. **Automação com Robot Framework** nos cenários críticos, para acelerar execuções repetitivas.
4. **Feedback contínuo**: evolução do plano com base nos resultados.

## 6. Técnicas Aplicadas

- Testes funcionais manuais (Postman).
- Testes exploratórios.
- Testes de limite (senhas, e-mails, mensagens de erro).
- Automação de API (Robot Framework + RequestsLibrary).

## 7. Mapa mental



## 8. Diferenças entre o Plano Inicial e a Execução Real

- O **plano inicial** previa 11 cenários básicos, cobrindo apenas fluxos positivos/negativos de usuários, login e produtos.
- A **execução real** expandiu para **42 cenários** (15 usuários, 4 login, 12 produtos, 11 carrinhos), cobrindo também:
  - Casos de **limite** (senhas curtas/longas, e-mails proibidos).

- **Mensagens de erro pouco claras** (cadastro duplicado, exclusão de inexistentes, carrinho concluído/cancelado em duplicidade).
- Cenários exploratórios não previstos (atualização criando registros automaticamente).

Essa evolução mostra que o plano amadureceu e passou a refletir melhor a realidade da API.

## 9. Cenários de Teste Planejados (Revisado)

### Usuários

- Cadastrar usuário válido (201).
- Cadastrar usuário duplicado (400).
- Cadastrar usuário com senha < 5 caracteres (400 – limite).
- Cadastrar usuário com senha > 10 caracteres (400 – limite).
- Cadastrar usuário com e-mail de provedores proibidos (400 – limite).
- Cadastrar usuário com e-mail inválido (400).
- Atualizar usuário inexistente → deve retornar erro (melhoria).
- Deletar usuário inexistente → mensagem clara esperada (melhoria).
- Deletar usuário já excluído → mensagem clara esperada (melhoria).

### Login

- Login com credenciais válidas → 200 + token.
- Login com senha inválida → 401.
- Login com usuário inexistente → 401.
- Login sem body → 400.
- Login com e-mail inválido → 400.

### Produtos

- Cadastro de produto válido (201).
- Cadastro de produto duplicado (400 – mensagem deve ser clara).
- Cadastro sem autenticação → 401.
- Atualizar produto válido (200).
- Atualizar produto com ID inexistente → mensagem incorreta (bug).
- Excluir produto válido (200).
- Excluir produto inexistente → mensagem pouco clara (melhoria).
- Acesso com usuário não-admin → 403.

## Carrinhos

- Criar carrinho válido (201).
- Criar carrinho duplicado → 400.
- Buscar carrinho inexistente → 400.
- Concluir compra (200).
- Concluir compra duplicada → mensagem genérica (melhoria).
- Cancelar compra (200).
- Cancelar compra duplicada → mensagem genérica (melhoria).
- Cancelar compra com token inválido → 401.

## 10. Priorização

- **Alta prioridade:** Login, cadastro de usuários, cadastro de produtos.
- **Média prioridade:** Atualizações e exclusões.
- **Baixa prioridade:** Consultas de registros inexistentes.

## 12. Matriz de Risco

- **Crítico:** Falhas em login e cadastro (usuário/produto).
- **Alto:** Mensagens de erro genéricas → impactam a usabilidade.
- **Moderado:** Problemas em carrinhos.
- **Baixo:** Consultas a registros inexistentes.

## 12. Candidatos à Automação

- Login válido e inválido.
- Cadastro de usuário válido, duplicado e senhas fora do limite.
- Cadastro de produto válido, duplicado e sem autenticação.
- Exclusão de produto válido e inexistente.
- Criação e cancelamento/conclusão de carrinhos.

Esses cenários trazem maior ganho em tempo, confiabilidade e cobertura.