

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:

- 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

Para garantir cobertura adequada, confiabilidade dos resultados e eficiência na validação da aplicação ServeRest, a execução dos testes seguirá uma estratégia organizada em quatro fases complementares:

1. Testes Manuais Exploratórios (Postman):

- Serão realizados testes exploratórios diretamente na API, utilizando o Postman, com foco em identificar comportamentos inesperados, verificar fluxos alternativos e validar limites. Essa etapa possibilita descobrir falhas além do que está documentado.

2. Registro Estruturado (QAlity/Jira):

- Os casos de teste planejados serão formalmente documentados no QAlity, permitindo execução controlada, acompanhamento da cobertura e rastreabilidade das execuções. Dessa forma, é possível visualizar quais testes passaram, falharam e quais bugs foram reportados.

3. Automação de Cenários Críticos (Robot Framework):

- Após a validação manual, os cenários de maior prioridade (login, cadastro de usuários, cadastro/exclusão de produtos e operações de carrinho) serão automatizados utilizando o Robot Framework e a biblioteca RequestsLibrary. Isso acelera a execução, facilita regressões e garante maior repetibilidade dos resultados.

4. Análise e Acompanhamento Contínuo (GitHub):

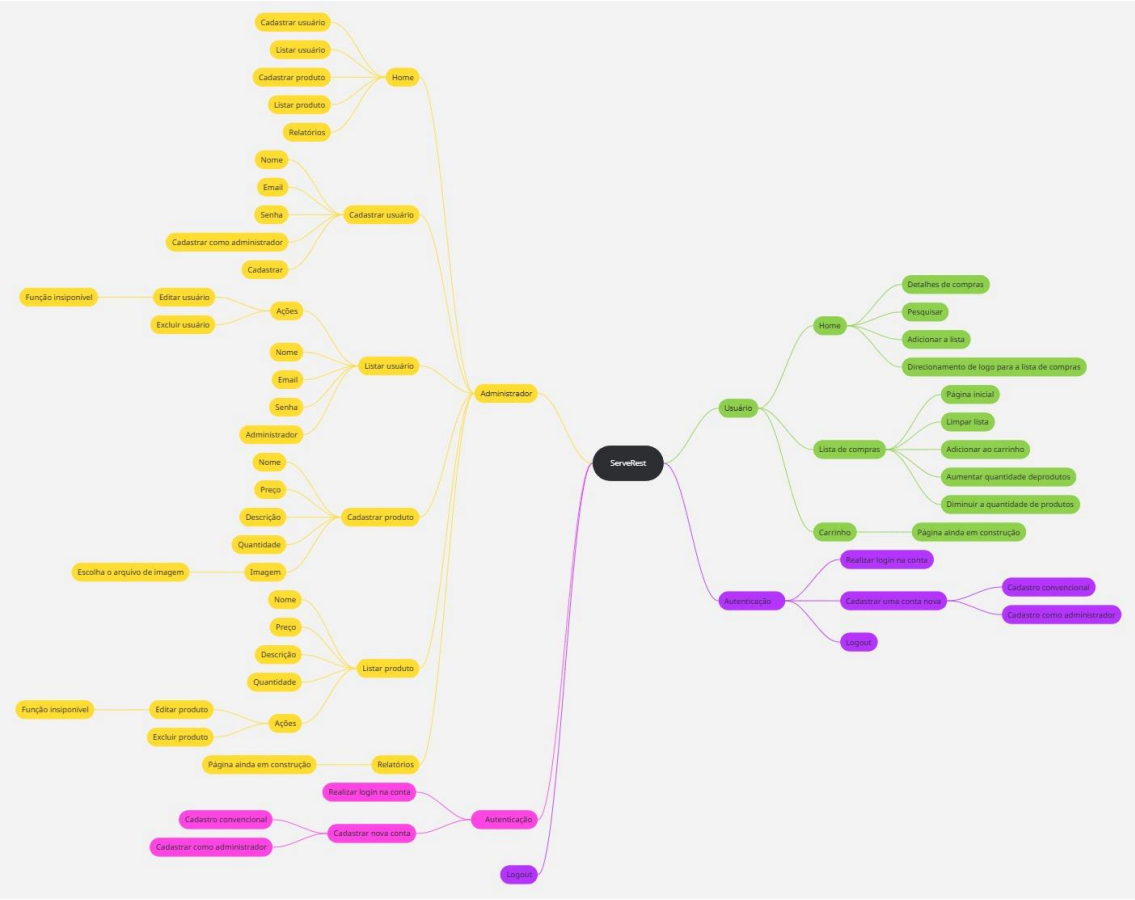
- Toda documentação e código automatizado serão versionados diariamente no GitHub. Os commits serão organizados e realizados em branches específicas, garantindo histórico de evolução, integração entre equipe e fácil auditoria das alterações.

Essa estratégia garante que os testes mantenham qualidade, abrangência e eficiência, equilibrando execução manual exploratória com automação de fluxos críticos.

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).

9. Cenários de Teste Planejados

Usuários

ID	Cenário de Teste	Resultado Esperado
US-01	Buscar usuário por ID válido	Retornar os dados do usuário
US-02	Buscar usuário por ID inexistente	Retornar erro informando que não existe

US-03	Cadastrar usuário válido	Usuário deve ser cadastrado com sucesso
US-04	Cadastrar usuário com e-mail Gmail ou Hotmail	Sistema deveria bloquear o cadastro
US-05	Cadastrar usuário com senha menor que 5 caracteres	Sistema deveria bloquear o cadastro
US-06	Cadastrar usuário com senha maior que 10 caracteres	Sistema deveria bloquear o cadastro
US-07	Cadastrar usuário duplicado	Bloquear cadastro e avisar que já existe
US-08	Cadastrar usuário com e-mail inválido	Retornar erro indicando formato incorreto
US-09	Atualizar usuário existente com body correto	Usuário deve ser atualizado com sucesso
US-10	Atualizar usuário inexistente	Retornar erro informando que não existe
US-11	Atualizar campo administrador com valor inválido	Retornar erro informando os valores aceitos
US-12	Atualizar usuário sem enviar body	Sistema deve informar campos obrigatórios
US-13	Deletar usuário existente	Usuário deve ser excluído com sucesso
US-14	Deletar usuário já excluído	Informar que não há nada para excluir

Login

ID	Cenário	Entrada/Descrição
CT-L01	Login com dados corretos	POST /login com email e senha válidos
CT-L02	Login sem body	POST /login sem email/senha no body
CT-L03	Login com senha incorreta	POST /login com email válido e senha inválida
CT-L04	Login com email inválido	POST /login com formato inválido de email

Produtos

ID	Cenário de Teste	Resultado Esperado
PR-01	Cadastrar produto com autenticação expirada	Retornar erro de autenticação
PR-02	Cadastrar produto já existente	Retornar erro informando que o produto já existe
PR-03	Cadastrar produto válido com token válido	Produto cadastrado com sucesso (201)
PR-04	Buscar produto por ID válida	Retornar informações do produto (200)

PR-05	Buscar produto com ID inexistente	Retornar erro informando que produto não existe
PR-06	Atualizar produto válido	Produto atualizado com sucesso (200)
PR-07	Atualizar produto com ID inválido	Retornar erro informando que ID não existe
PR-08	Atualizar produto com token de usuário não-admin	Retornar erro de permissão
PR-09	Cadastrar produto com token de usuário não-admin	Retornar erro de permissão
PR-10	Deletar produto válido	Produto excluído com sucesso (200)
PR-11	Deletar produto com ID inexistente	Retornar erro ou aviso claro que não foi excluído
PR-12	Buscar todos os produtos cadastrados	Retornar lista de produtos (200)

Carrinhos

ID	Cenário de Teste	Resultado Esperado
CA-01	Buscar todos os carrinhos (GET)	Retornar lista de carrinhos (200)
CA-02	Buscar carrinho por ID válida (GET)	Retornar lista de produtos do carrinho (200)
CA-03	Buscar carrinho por ID inexistente (GET)	Retornar erro informando que não existe
CA-04	Cadastrar carrinho válido (POST)	Criar carrinho e retornar sucesso (201)
CA-05	Cadastrar carrinho repetido (POST)	Retornar erro indicando que já existe carrinho
CA-06	Adicionar produto além da quantidade disponível (POST)	Retornar erro informando falta de estoque
CA-07	Concluir compra (DELETE)	Excluir carrinho e retornar sucesso (200)
CA-08	Concluir compra novamente no mesmo usuário (DELETE)	Retornar erro ou aviso de que não há carrinho
CA-09	Cancelar compra (DELETE)	Cancelar carrinho e reabastecer estoque (200)
CA-10	Cancelar compra novamente no mesmo usuário (DELETE)	Retornar erro ou aviso de que não há carrinho
CA-11	Cancelar compra com token inválido/expirado (DELETE)	Retornar erro de autenticação (401)

10. Priorização

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

11. 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

12.1 Critérios de Seleção

Os cenários foram selecionados para automação com base nos seguintes critérios:

1. **Criticidade:** Cenários que validam funcionalidades essenciais para o negócio
2. **Repetitividade:** Testes que precisam ser executados frequentemente (regressão)
3. **Estabilidade:** Cenários com comportamento previsível e interface estável (API REST)
4. **Complexidade de dados:** Cenários que requerem múltiplas combinações de dados

12.2 Cenários Selecionados para Automação

Módulo: Login

ID	Cenário	Status Code	Criticidade	Motivo da Automação
CT-L01	Login válido	200 + token	Alta	Pré-requisito para todas operações autenticadas
CT-L03	Login com senha inválida	401	Alta	Validação de segurança crítica

Não automatizados: Login sem body, Login com email inválido Motivo: Baixa complexidade, validação simples de campos obrigatórios

Motivo: Baixa complexidade, validação simples de campos obrigatórios, execução manual rápida.

Módulo: Usuários

ID	Cenário	Status Code	Criticidade	Motivo da Automação
US-03	Cadastrar usuário válido	201	Alta	Fluxo principal da aplicação

US-07	Cadastrar usuário duplicado	400	Alta	Validação de regra de negócio crítica
US-05	Senha < 5 caracteres	201 (bug)	Alta	Valida bug crítico identificado
US-06	Senha > 10 caracteres	201 (bug)	Alta	Valida bug crítico identificado
US-04	Email Gmail/Hotmail	201 (bug)	Alta	Valida bug crítico identificado

Não automatizados: Buscas, atualizações e exclusões Motivo: Validações de mensagens requerem análise crítica humana, ROI baixo

Motivo:

- Operações de consulta/atualização têm menor impacto no negócio
- Testes exploratórios manuais são mais eficientes para validar mensagens de erro
- Baixa repetitividade comparado aos fluxos principais

Módulo: Produtos

ID	Cenário	Status Code	Criticidade	Motivo da Automação
PR-03	Cadastrar produto válido	201	Alta	Essencial para fluxo de carrinhos
PR-01	Cadastrar sem autenticação	401	Alta	Validação de segurança
PR-10	Excluir produto válido	200	Média	Necessário para limpeza de dados nos testes

Não automatizados: Produtos duplicados, buscas, atualizações Motivo: Cenários de validação de mensagens são melhor verificados manualmente

Motivo:

- Cenários de validação de mensagens são melhor verificados manualmente
- Testes exploratórios capturam nuances que automação pode não detectar
- ROI baixo para cenários de erro menos críticos

Módulo: Carrinhos

ID	Cenário	Status Code	Criticidade	Motivo da Automação
CA-04	Criar carrinho válido	201	Alta	Fluxo principal de negócio
CA-07	Concluir compra	200	Alta	Simula jornada completa do usuário

Não automatizados: Carrinho duplicado, cancelar compra, buscas
Motivo: Cenários de exceção têm menor impacto, mensagens requerem validação manual

Motivo:

- Validações de regras de negócio específicas são mais ágeis manualmente
- Mensagens de erro requerem análise crítica humana
- Cenários de exceção têm menor impacto em produção

12.3 Cobertura de Automação

Resumo quantitativo:

Módulo	Total de Cenários	Automatizados	% Cobertura	Prioridade
Login	4	2	50%	Alta
Usuários	15	5	33%	Alta
Produtos	12	3	25%	Media
Carrinhos	11	2	18%	Alta
Total	42	12	29%	-