Planejamento de Testes – ServeRest

> **Aplicação**: ServeRest Web (v2.29.7)
>
> **Ferramentas**: Postman, Robot Framework, AmazonQ, Jira + QALity
>
> **Autora:** Sabrina Freitas

---

1. Apresentação

A aplicação ServeRest Web (v2.29.7) será submetida a um ciclo de testes para validar a conformidade com as regras de negócio descritas nas User Stories US-001 a US-003, garantindo a coerência da API com o Swagger e contemplando também cenários alternativos (negativos, limítrofes e de erro).

2. Objetivo

- Assegurar que os requisitos funcionais das User Stories US-001 a US-003 sejam cumpridos.
- Validar que os endpoints da API estão consistentes com o contrato definido no Swagger.
- Garantir cobertura de cenários positivos, negativos e limítrofes.

3. Escopo

Funcionalidades incluídas

- CRUD completo de usuários (`/usuarios`)
- Autenticação de usuários (`/login`)
- CRUD completo de produtos (`/produtos`), incluindo regras de autenticação
- Ações com carrinhos e manipulação de produtos em carrinhos

Tipos de teste incluídos

- Testes Funcionais (fluxos principais e alternativos)

- Testes de Regras de Negócio (conformidade com US e Acceptance Criteria)
- Testes de Validação de Schema (contrato da API x Swagger)

Fora do escopo

- Testes de Performance, Carga e Estresse
- Testes de Segurança avançados
- Testes de Usabilidade da interface front-end

---

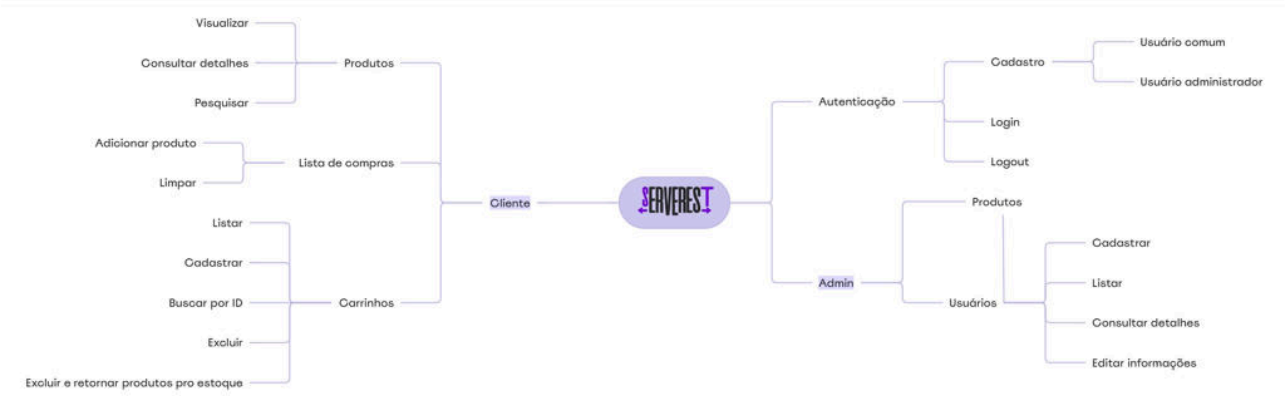
4. Análise

Técnicas aplicadas

- **Baseado em Requisitos:** User Stories + Swagger
- **Partição de Equivalência:** e.g., validar senhas dentro/fora do range permitido
- **Valor-Limite:** senhas com exatamente 5 e 10 caracteres; IDs com 16 caracteres
- **Testes de Valores Inválidos:** `NULL`, vazio, zero, negativos, strings inválidas

---

5. Mapa mental



---

6. Cenários de teste

→ Planejados (Acceptance Criteria)

US 001 – Usuários

- Cadastro válido com todos os campos obrigatórios.
- Ações em usuários inexistentes → devem falhar.
- Cadastro com e-mail duplicado (POST) → não permitido.
- PUT em usuário inexistente → deve criar novo usuário.

- PUT com e-mail duplicado → não permitido.
- Cadastro com e-mail bloqueado (gmail/hotmail) → não permitido.
- Cadastro com e-mail inválido → não permitido.
- Cadastro com senha fora do limite (<5 ou >10) → não permitido.

US 002 – Login

- Login com usuário inexistente → deve falhar (401).
- Login com senha inválida → deve falhar (401).
- Login válido → autenticação bem-sucedida.
- Geração de token Bearer no login válido.
- Validade do token → expira em 10 minutos.

US 003 – Produtos

- Ações sem autenticação → devem falhar.
- Cadastro de produto com nome duplicado → não permitido.
- Exclusão de produto em carrinho → não permitido.
- UPDATE em produto inexistente → deve criar novo produto.
- PUT com nome duplicado → não permitido.

→ Executados

- **Geral**
  - Todos Acceptance Criteria.
  - Automação de fluxos de cadastro, autenticação e manipulação de IDs.
  - Cenários negativos e positivos mapeados no Swagger.
- **Cenários negativos e alternativos**
  - Campos obrigatórios ausentes ou inválidos: NULL, vazio, valores negativos.
  - IDs inválidos (busca/cadastro): menores ou maiores que 16 caracteres, em branco, inexistentes.
  - Validações de tipos e valores:
    - Campos numéricos recebendo *strings*.
    - *Strings* com caracteres especiais inválidos.
    - Quantidade/preço com valores negativos ou zero.
  - Credenciais inválidas: senha incorreta, e-mails malformados (ex.: *qateste*).
  - Campo administrador: valor inválido (diferente de `true/false`).

Fluxos de autenticação

- Manipulação de módulos com token expirado ou malformado.
- Tentativa de acesso com perfil sem permissão de administrador.

Carrinhos

- Criar carrinho.
- Concluir compra.
- Cancelar compra.
- Adicionar produto ao carrinho.
- Listar carrinhos.
- Buscar carrinho por ID.

- **Produtos:**
  - Tentativa de cadastrar produtos com IDs inexistentes ou inválidos.
  - Adicionar ao carrinho produtos com IDs inexistentes ou inválidos.

---

7. Priorização da execução

- **Alta prioridade:** Fluxos críticos de negócio (CRUD de Usuários, Login, Produtos e Carrinhos).
- **Média prioridade:** Validações importantes de dados (campos obrigatórios, formatos, duplicidades).
- **Baixa prioridade:** Cenários alternativos e mensagens de erro.

8. Matriz de risco

- **Probabilidade (P):** 1 (Baixa), 2 (Média), 3 (Alta)
- **Impacto (I):** 1 (Baixo), 2 (Médio), 3 (Alto)
- **Exposição ao Risco (ER):** P × I

ID	User Story	Acceptance Criteria	P	I	ER	Classificação	Mitigação/Teste
AC1	US-001 Usuários	Cadastro válido com todos os campos obrigatórios	3	3	9	**Crítico**	Teste positivo de fluxo completo de cadastro.
AC2	US-001 Usuários	Ações em usuários inexistentes devem falhar	2	3	6	Alto	Testes negativos com IDs inexistentes.

AC3	US-001 Usuários	Cadastro com e-mail duplicado (POST) não permitido	3	3	9	**Crítico**	Testes de duplicidade de e-mail.
AC4	US-001 Usuários	PUT em usuário inexistente deve criar novo usuário	2	2	4	Médio	Testes de PUT com ID inexistente.
AC5	US-001 Usuários	PUT com e-mail duplicado não permitido	3	3	9	**Crítico**	Testes de duplicidade de e-mail via PUT.
AC6	US-001 Usuários	Cadastro com e-mail bloqueado (gmail/hotmail) não permitido	2	2	4	Médio	Testes de validação de domínios de e-mail.
AC7	US-001 Usuários	Cadastro com e-mail inválido não permitido	3	2	6	Alto	Testes com formatos inválidos (sem @, etc.).
AC8	US-001 Usuários	Cadastro com senha fora do limite (<5 ou >10) não permitido	3	2	6	Alto	Testes de partição de equivalência e BVA (5 e 10).
AC9	US-002 Login	Login com usuário inexistente deve falhar (401)	2	3	6	Alto	Testes com usuário não cadastrado.

AC10	US-002 Login	Login com senha inválida deve falhar (401)	2	3	6	Alto	Testes com senha incorreta.
AC11	US-002 Login	Login válido → autenticação bem-sucedida	3	3	9	**Crítico**	Teste positivo de login.
AC12	US-002 Login	Geração de token Bearer no login válido	3	3	9	**Crítico**	Testar se token é gerado corretamente.
AC13	US-002 Login	Validade do token expira em 10 minutos	2	3	6	Alto	Teste de expiração de sessão/token.
AC14	US-003 Produtos	Ações sem autenticação devem falhar	3	3	9	**Crítico**	Testes de endpoints sem token ou sem token de ADMIN.
AC15	US-003 Produtos	Cadastro de produto com nome duplicado não permitido	3	3	9	**Crítico**	Testes de duplicidade de produto.
AC16	US-003 Produtos	Exclusão de produto em carrinho não permitida	2	3	6	Alto	Testes de regra de negócio (restrição de exclusão).
AC17	US-003 Produtos	PUT em produto inexistente deve criar novo produto	2	2	4	Médio	Testes de PUT em IDs inexistentes.
AC18	US-003 Produtos	PUT com nome duplicado não	3	3	9	**Crítico**	Testes de duplicidade via

		permitido					PUT.

---

9. Testes candidatos à automação

Critérios de seleção

- Exposição ao Risco (ER) ≥ 6
- Frequência de execução (fluxos principais)
- Impacto no negócio (autenticação, integridade de dados)
- Regressão (validações que devem permanecer funcionais)

Testes selecionados

US-001 – Usuários (12 testes)

1. Cadastro válido completo (AC1)
2. E-mail duplicado no POST (AC3)
3. Formato de e-mail inválido (AC7)
4. Senha muito curta (AC8)
5. Senha muito longa (AC8)
6. Senha no limite máximo (10 caracteres)
7. Senha no limite mínimo (5 caracteres)
8. Senha com campos vazios
9. Cadastro com domínio “@gmail.com” bloqueado (AC6)
10. Cadastro com domínio “@hotmail.com” bloqueado (AC6)
11. Deletar usuário
12. PUT com ID inexistente cria um novo usuário
13. Busca de usuário inexistente (AC2)

US-002 – Login (3 testes)

14. Login válido com geração de token (AC11 + AC12)
15. Login com e-mail inválido (AC9 + AC10)
16. Login com senha inválida (AC9 + AC10)
17. Expiração de token após 10 minutos (AC13)

US-003 – Produtos (6 testes)

18. Cadastro válido completo

19. Cadastro com token inválido

20. Cadastro com nome duplicado (AC15)

21. Cadastro sem permissão de ADMIN

22. Exclusão de produto em carrinho (AC16)

23. Listar produtos

Carrinhos (3 testes)

24. Criar carrinho válido

25. Concluir compra

26. Cancelar compra

10. Cobertura de testes automatizados

- **Total: 26 testes automatizados** cobrindo os cenários críticos (ER ≥ 6) e fluxos principais de negócio.
- Cobertura de endpoints: 9/9 = 100%

11. QALity

Os cenários de teste planejados e executados foram gerenciados e documentados na ferramenta QAlity, possibilitando rastreabilidade direta.

Anteriormente, foram realizados 60 testes manuais no Postman. A partir desse conjunto, 29 casos considerados mais relevantes para os fluxos principais da aplicação foram migrados para o QAlity, priorizando cobertura de funcionalidades críticas e regras de negócio.Organização dos testes no QAlity

- **Estrutura:** Os casos de teste foram organizados por módulo funcional — **Usuários**, **Login**, **Produtos** e **Carrinhos** — e vinculados às respectivas User Stories **(US-001 a US-003)** quando aplicável.
- **Nomenclatura:** Foi adotado o padrão `USXXX - [Ação] - [Resultado Esperado]`, garantindo clareza e consistência na identificação dos testes.
- **Detalhamento:** Cada cenário inclui:
  - Passos de execução objetivos e sequenciais
  - Dados de entrada e pré-condições
  - Resultado esperado
  - Evidências de execução (prints e logs), quando aplicável

Execução

- **Total de testes executados anteriormente no Postman:** 60
- **Total de casos mapeados no QAlity:** 29
- **Ciclos de teste executados:** 1 (versão v2.29.7)
- **Taxa de aprovação geral: 88,89%** (27 cenários executados)
- **Falhas identificadas:** 3 falhas relacionadas a regras de negócio das User Stories, detalhadas no documento de rastreabilidade de issues.

Além disso, foram **registrados issues adicionais** não relacionados diretamente ao Swagger ou aos Acceptance Criteria, mas que representariam riscos de segurança caso a API fosse privada. Exemplos:

- Endpoints de manipulação de usuários (`GET` , `POST` , `DELETE`) sem autenticação obrigatória.
- Endpoint de listagem de carrinhos (`GET /carrinhos`) acessível sem autenticação.

Embora esses comportamentos estejam de acordo com as especificações atuais da API (e, portanto, foram marcados como “Passed” nos testes), issues foram abertos com evidências e detalhamento, de forma a documentar os riscos e apoiar decisões futuras.

Benefícios

- **Centralização e versionamento** de todos os cenários e evidências de teste
- **Acompanhamento em tempo real** da cobertura de requisitos e status de execução
- **Priorização facilitada para automação**, com base em criticidade e resultados obtidos