Planejamento de teste API - ServeRest

Plano de Testes - API ServeRest @

Apresentação 🖉

Este plano de testes tem como objetivo estruturar e documentar as atividades de qualidade aplicadas à API **ServeRest**. Os testes foram baseados em três principais funcionalidades: usuários, login e produtos. O foco está nas **regras de negócio**, nos **requisitos definidos pelas User Stories** e na **garantia da integridade e segurança das rotas** REST disponíveis.

Informações Gerais do Plano de Testes ∅

Item	Descrição
Nome do Projeto	API ServeRest
Data de Início dos Testes	06/05/2025
Data Estimada de Término	09/05/2025
Ambiente de Teste	ServeRest (Ambiente de Desenvolvimento e Homologação)
Ferramentas Utilizadas	Postman, Confluence, Jira, Swagger,xmind
QA Responsável	Marcos Paulo Alves de Freitas
Instrutor	Sms
Modalidade da Atividade	Individual
Fontes de Requisitos	Swagger da API, User Stories (US-User, LG-LOGIN, PR-PRODUTOS, CR-CARRINHO), Critérios de Aceitação

Objetivo 🖉

Validar, através de testes manuais e automatizados, se a API atende às especificações do negócio, garantindo que os endpoints implementados estejam funcionando conforme os critérios de aceitação definidos nas User Stories, cobrindo cenários positivos, negativos e alternativos.

Escopo €

Escopo incluído: 🔗

- Testes funcionais das rotas:
 - /usuarios
 - ∘ /login
 - ∘ /produtos
 - o /carrinho
- Validação de regras de negócio

- Testes negativos e de segurança básica (autenticação, acesso não autorizado)
- Automação via Postman (aba *Tests*)

Fora de escopo: ${\mathscr O}$

- Integrações com APIs externas
- Testes de performance, carga ou estresse
- · Testes de front-end

Análise 🖉

A análise baseou-se em:

- Swagger da API ServeRest
- User Stories fornecidas (US001, US002, US003)
- Regras de negócio destacadas em cada DoD e critérios de aceitação
- Comportamentos observados durante testes exploratórios na API

Técnicas Aplicadas 🖉

- Caixa preta (Black Box Testing)
- Técnica de particionamento de equivalência
- Análise de valores-limite
- Teste baseado em regras de negócio
- Exploratory Testing
- xmind para mapa mental

Mapa Mental da Aplicação 🔗

Cenários de Teste Planejados – Usuários 🔗

Critério de Aceite	Descrição	Link para o Cenário de Teste
01	Os vendedores devem possuir NOME, E-MAIL, PASSWORD e ADMINISTRADOR	Acessar
02	Não deverá ser possível fazer ações para usuários inexistentes	Acessar
03	Não deve ser possível criar usuário com e-mail já utilizado	Acessar
04	PUT com ID inexistente deve criar novo usuário	Acessar
05	PUT não deve permitir e-mail duplicado	Acessar
06	Testes executados devem conter evidências	Acessar

07	Bloquear cadastro com e-mails gmail e hotmail	Acessar
08	E-mails devem seguir formato válido	Acessar
09	Senhas devem ter entre 5 e 10 caracteres	Acessar
10	Cobertura baseada no Swagger e cenários alternativos	Acessar

Login: 🖉

Critério de Aceite	Descrição	Link para o Cenário de Teste
1	Usuários não cadastrados não deverão conseguir autenticar	Acessar
2	Usuários com senha inválida não deverão conseguir autenticar	Acessar
3	No caso de não autenticação, deverá ser retornado um status code 401 (Unauthorized)	Acessar
4	Usuários existentes e com a senha correta deverão ser autenticados	Acessar
5	A autenticação deverá gerar um token Bearer	Acessar
6	A duração da validade do token deverá ser de 10 minutos	Acessar
7	Os testes executados deverão conter evidências	Acessar
8	A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos	Acessar

Produtos: @

Critério de Aceite	Descrição	Link para o Cenário de Teste
1	Usuários não autenticados não devem conseguir realizar ações na rota de Produtos	Acessar
2	Não deve ser possível realizar o cadastro de produtos com nomes já utilizados	Acessar

3	Não deve ser possível excluir produtos que estão dentro de carrinhos (dependência API Carrinhos)	Acessar
4	Caso não exista produto com o ID informado na hora do UPDATE, um novo produto deverá ser criado	Acessar
5	Produtos criados através do PUT não poderão ter nomes previamente cadastrados	Acessar
6	Os testes executados deverão conter evidências	Acessar
7	A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos	Acessar

Priorização da Execução dos Cenários ${\mathscr O}$

Prioridade	Módulo	Cenário
Alta	Login	Autenticação válida / inválida
Alta	Usuários	Criar usuário com dados válidos
Alta	Produtos	Criar produto autenticado
Média	Usuários	Validação de campos obrigatórios
Média	Produtos	Nome duplicado
Baixa	Produtos	Excluir produto em carrinho (não automatizado ainda)

Matriz de Risco 🔗

Módulo	Links para as Matrizes de Risco
Usuários	Matriz de Risco - Usuários
Login	Matriz de Risco - Login
Produtos	Matriz de Risco - Produtos
Carrinho	Matriz de Risco - Carrinho

Cobertura de Testes ${\mathscr O}$

Rotas /usuarios:

Cobertura funcional completa (100%) dos critérios definidos nas User Stories.

Rotas /login:

Cobertura total (100%) dos critérios relacionados à autenticação.

Rotas /produtos:

Cobertura de 90% dos critérios planejados.

Pendência: Teste da rota DELETE /produtos quando o produto estiver vinculado a um carrinho.

Rotas /carrinhos:

Cobertura funcional parcial (atualmente 70%) devido à pendência de alguns testes.

A cobertura já inclui:

Adição, remoção e atualização de produtos no carrinho

Visualização e esvaziamento do carrinho

Validações de autenticação e quantidade inválida

Pendência:

Testes adicionais para os cenários que envolvem produtos vinculados a um carrinho, como a remoção de um produto de um carrinho já utilizado.

Teste da rota DELETE /produtos quando o produto estiver vinculado a um carrinho.

Testes candidatos a automação 🖉

[LINK PARA TESTES SUGERIDOS PARA AUTOMAÇÃO → 📵 Testes candidatos a automação]

📁 Documentos Gerados 🖉

- V Plano de Testes (este documento)
- Collection Postman com scripts em aba *Tests* (em progresso, posso te ajudar a montar)
- 🔽 Documento de Issues & Melhorias (posso estruturar em Markdown se quiser usar no Confluence ou copiar para Jira)