PROGRAMA DE BOLSAS

QUALITY & TEST AUTOMATION

Bruna Baldissera

Plano de Testes - API ServeRest

Setembro 2023

Introdução

O ServeRest é uma API REST gratuita que simula uma loja virtual, projetada para servir como um recurso valioso no estudo e teste de APIs. Este ambiente oferece uma variedade de funcionalidades essenciais, incluindo autenticação de usuários, registro de contas, gerenciamento de produtos e carrinhos de compras.

Neste contexto, vamos explorar de forma abrangente os principais aspectos dessa ferramenta, abordando tanto as funcionalidades críticas quanto as melhores práticas em qualidade de software. Ao fazê-lo, destacaremos a importância do teste de APIs como um componente vital para garantir a qualidade e confiabilidade da experiência do usuário final. Além disso, estaremos prontos para identificar e relatar quaisquer inconsistências que possam surgir durante o uso do ServeRest, contribuindo assim para seu aprimoramento contínuo.

Índice

1. **Objetivo.....3**
2. **Escopo.....3**
   2.1 Funcionalidades.....3
3. **Mapa Mental.....4**
4. **Abordagem de Testes.....5**
   4.1 Login.....5
   4.2 Usuários.....5
   4.3 Produtos.....6
   4.4 Carrinho de compras.....6
5. **Prioridades.....7**
6. **Estratégia de Testes 7**
7. **Candidatos para Automação.....7**
8. **Recursos.....7**
9. **Cronograma.....8**

1. Objetivo

O objetivo deste plano de testes é assegurar a funcionalidade e confiabilidade das operações essenciais oferecidas pela API ServeRest, que incluem login, cadastro e gerenciamento de usuários, produtos e carrinhos. Ao mesmo tempo, este plano visa criar um ambiente propício para aprendizado, onde os testadores possam ganhar experiência e compreensão prática sobre testes de API.

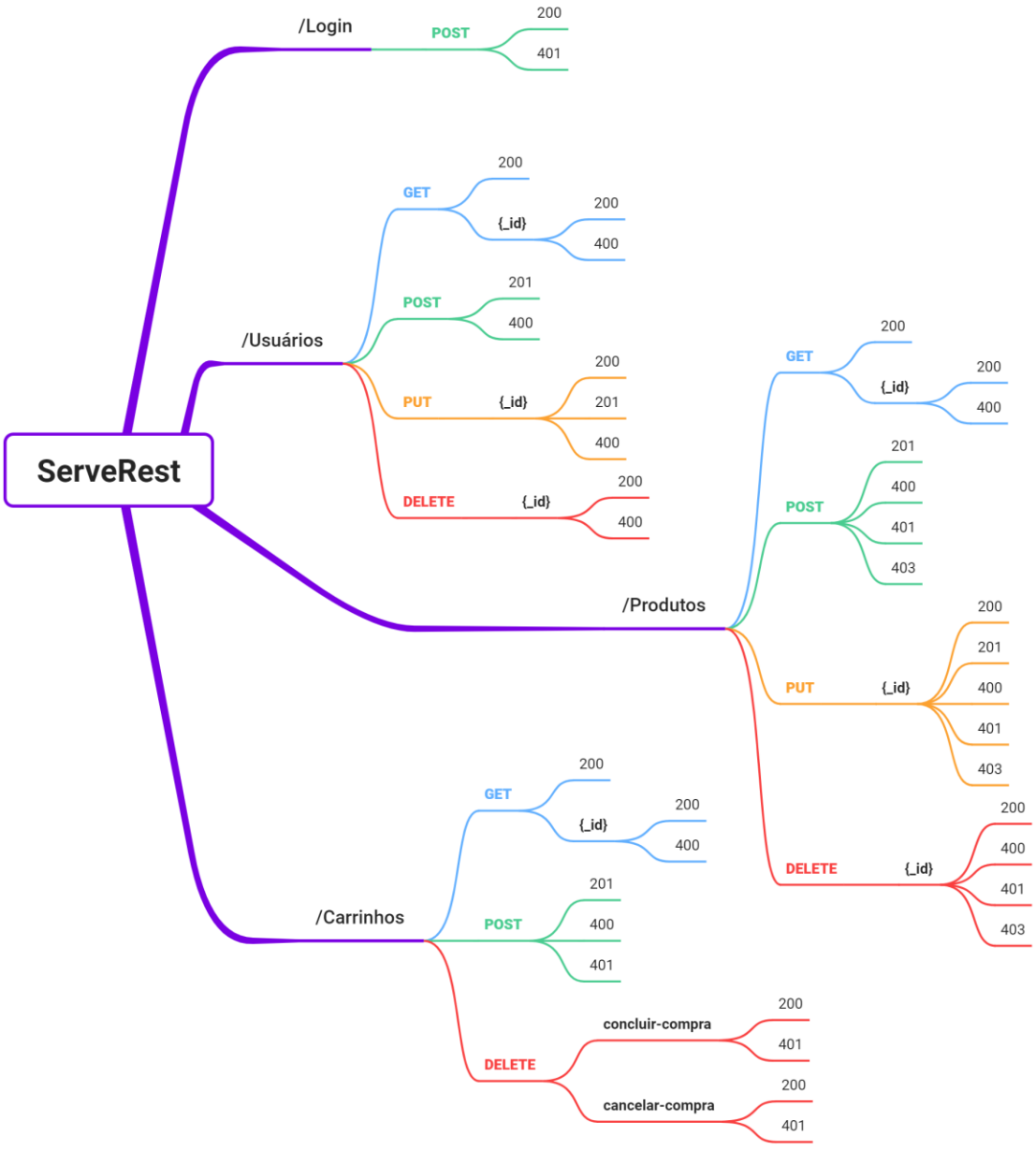
2. Escopo

A. Funcionalidades incluídas no escopo de teste:
   1. Registro de novos usuários.
   2. Login e autenticação de usuários.
   3. Gerenciamento de usuários.
   4. Gerenciamento de produtos.
   5. Gerenciamento de carrinhos.

B. Ambiente de Teste:
   1. Sistema Operacional: Windows 10 Pro – 64 bits.
   2. Plataforma: Postman

3. Mapa Mental

Segue abaixo o mapa mental utilizado como base para definir o plano de testes, bem como para automatizar os fluxos e seus respectivos resultados:



4. Abordagem de testes

Suíte de testes 1 – Login

CT_101: Realizar login válido;

CT_102: Realizar login inválido (usuário não cadastrado);

CT_103: Realizar login inválido (email inválido);

CT_104: Realizar login inválido (campos vazios);

Suíte de testes 2 – Usuários

CT_201: Listar todos os usuários;

CT_202: Criar usuário válido;

CT_203: Criar usuário inválido (email inválido);

CT_204: Criar usuário inválido (campos vazios);

CT_205: Criar usuário inválido (duplicidade);

CT_206: Criar usuário inválido (senha não segura);

CT_207: Criar usuário inválido (senha sem limite máximo);

CT_208: Criar usuário inválido (senha sem limite mínimo);

CT_209: Listar usuário por ID válido;

CT_210: Listar usuário por ID inválido;

CT_211: Deletar usuário por ID válido;

CT_212: Deletar usuário por ID inválido;

CT_213: Deletar usuário inválido (campos vazios);

CT_214: Editar usuário por ID válido;

CT_215: Editar usuário inválido (caminho vazio);

CT_216: Editar usuário inválido (campos vazios);

CT_217: Editar usuário (novo ID);

CT_218: Editar usuário (duplicidade);

Suíte de testes 3 – Produtos

CT_301: Listar todos os produtos;

CT_302: Criar produto válido (com token);

CT_303: Criar produto válido (sem token);

CT_304: Criar produto inválido (valores incorretos);

CT_305: Criar produto inválido (campos vazios);

CT_306: Criar produto inválido (duplicidade);

CT_307: Criar produto válido (sem admin);

CT_308: Listar produto por ID válido;

CT_309: Listar produto por ID inválido;

CT_310: Editar produto por ID válido (com token);

CT_311: Editar produto por ID válido (sem token);

CT_312: Editar produto inválido (caminho vazio);

CT_313: Editar produto inválido (campos vazios);

CT_314: Editar produto (novo ID);

CT_315: Editar produto (duplicidade);

Suíte de testes 4 - Carrinhos

CT_401: Listar todos os carrinhos;

CT_402: Criar carrinho válido (com token);

CT_403: Criar carrinho válido (sem token);

CT_404: Criar carrinho inválido (campos vazios);

CT_405: Listar carrinho por ID válido;

CT_406: Listar carrinho por ID inválido;

CT_407: Deletar carrinho válido (com token);

CT_408: Deletar carrinho válido (sem token);

CT_409: Deletar carrinho inválido (não existente);

CT_410: Cancelar compra válida (com token);

CT_411: Cancelar compra válida (sem token);

CT_412: Cancelar compra inválida (não existente).

5. Prioridades

Prioridades são definidas com base na importância das funcionalidades, riscos potenciais e critérios de negócios. Os principais testes críticos nesse plano de testes incluem o processo de autenticação do usuário, o gerenciamento de usuários, produtos e carrinhos. Dessa forma, é possível evitar grandes erro e aprimorar a ferramenta de forma a visar uma boa experiência ao usuário.

6. Estratégia de Testes

A estratégia de testes adotada envolve uma abordagem abrangente e bem planejada para garantir a qualidade e o desempenho da API ServeRest. Para alcançar isso, uma variedade de tipos de testes foi empregada, cada um com seu foco específico.

Primeiramente, os testes de login foram efetuados, pois são de alta prioridade, já que a autenticação é uma funcionalidade central da aplicação. Também foram priorizados os testes de validação de campos, que verificam se o aplicativo lida adequadamente com entradas inválidas, prevenindo erros e garantindo a integridade dos dados. Além disso, as funcionalidades disponíveis de gerenciamento de usuários, produtos e carrinhos foram devidamente testadas, como a criação, edição e exclusão de cada um desses elementos.

Esta estratégia de testes é iterativa e se adapta às mudanças nos requisitos e no progresso do projeto. Priorizamos testes com base na importância das funcionalidades e nos riscos potenciais, buscando garantir que a API ServeRest ofereça uma excelente experiência aos usuários finais.

7. Candidatos para Automação

Tendo em vista as questões abordadas anteriormente, são fortes candidatos para automação os fluxos de login, cadastro e gerenciamento de usuários, produtos e carrinhos.

8. Recursos

1. Linguagem de Programação: Ruby.
2. Frameworks: Capybara, Cucumber, RSpec e Selenium.
3. IDE para desenvolvimento e depuração: Visual Studio Code.

9. Cronograma

9.1. Etapa 01 – Preparação e planejamento (de 25/09/23 a 29/09/23)

- Definição dos casos de testes;
- Preparação do ambiente de testes;
- Configuração e instalação das ferramentas necessárias.

9.2. Etapa 02 – Execução do plano de testes (de 02/10/23 a 04/10/23)

- Automação dos testes;
- Execução do plano de testes;

9.3. Etapa 03 – Relatórios e correções (de 04/10/23 a 06/10/23)

- Documentação dos resultados obtidos;
- Identificação e correção de possíveis problemas encontrados

9.4. Etapa 04 – Apresentação do relatório de testes (09/10/2023)