Plano de Teste

Teste de Interface da API Serverest

**Resumo**

Este documento apresenta o planejamento das atividades de testes do sistema e descreve o plano de ação para implementar e testar o CRUD de usuários no MarketPlace do ServeRest, deseja-se verificar se a usabilidade da interface da API, se cumpre todos os requisitos funcionais, garantir a qualidade geral do uso da API e identificar possíveis defeitos, para isso será utilizada a ferramenta Postman para executar e avaliar os testes.

**Envolvidos**

Gabriel Oliveira

CompassUOL PerfMerch 3

**Funcionalidade ou Módulos a serem testados**

Teste da funcionalidade da interface da API Serverest com base na documentação Swagger para a rota de login para autenticação do usuário, a rota de usuário para gerenciar os usuários, consultar dados para login e cadastrar administrador e a rota de produtos para poder cadastrar, editar, atualizar e excluir produtos.

CRITÉRIOS DE ACEITAÇÃO:

**RF01** - Os vendedores (usuários) deverão possuir os campos NOME, E-MAIL, PASSWORD e ADMINISTRADOR;

**RF02** -Não deverá ser possível fazer ações e chamadas para usuários inexistentes;

**RF03** - Não deve ser possível criar um usuário com e-mail já utilizado;

**RF04** - Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado;

**RF05** - Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT;

**RF06** - Não deverá ser possível cadastrar usuários com e-mails de provedor gmail e hotmail;

**RF07** - Os e-mails devem seguir um padrão válido de e-mail para o cadastro;

**RF08** - As senhas devem possuir no mínimo 5 caracteres e no máximo 10 caracteres;

**NF01** - Os testes executados deverão conter evidências;

**NF02** - A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos;

**Local dos testes**

Será desenvolvido no computador próprio do testador, utilizando o software Postman com o URL base da API

**Recursos necessários**

É necessário internet com conexão estável e máquina pessoal com o software Postman.

**Critérios usados**

Serão realizados todos os testes funcionais com base aos critérios de aceitação da user stories e serão avaliados como “passou” e “falhou” com base no resultado esperado e o resultado obtido, com cerca de 10 cenários de testes levantados de mais importância e valor para a aplicação.

**Riscos**

Os testes deverão ser executados o quanto antes para evitar imprevistos, como ficar sem fornecimento de energia elétrica perto da deadline. Caso a conexão com a internet WiFi seja interrompida, deve-se ativar os dados móveis do celular para a execução da tarefa.

**Como os resultados do teste serão divulgados**

Os resultados finais serão fornecidos após o término dos ciclos, será apresentado um relatório dos resultados de testes e os problemas (bugs) encontrados serão reportados no Issues do projeto do GitLab.

**Cronograma**

Realização do planejamento de teste: 02/07/24 – 09/07/24

Tabulação de dados: 09/07/24 -10/07/24

Entrega dos resultados: 11/07/2024

**Plano de implementação e testes de CRUD de usuários no ServeRest**

**Endpoints Necessários**

* **POST** /login – Realiza login
* **GET** /usuarios – Lista usuários cadastrados
* **POST** /usuarios – Cadastra usuários
* **GET** /usuarios/{\_id} – Busca usuários por ID
* **PUT** /usuarios/{\_id} – Edita usuário
* **DELETE** /usuarios/{\_id} – Exclui usuário

·         **GET** /produtos – Lista usuários cadastrados

·         **POST** /produtos – Cadastra produto

·         **GET** /produtos/{\_id} – Busca produto por ID

·         **DELETE** /produtos/{\_id} – Exclui produto

·         **PUT** /produtos/{\_id} – Edita produto

**História de usuário**

[API] Usuários

Sendo um vendedor de uma loja gostaria de poder me cadastrar no Marketplace do ServeRest para poder realizar as vendas dos meus produtos.

**DoR (Definição de pronto)**

* Banco de dados e infraestrutura para desenvolvimento disponibilizados;
* Ambiente de testes disponibilizado.

**DoD (Definição de Feito)**

* CRUD de cadastro de vendedores (usuários) implementado (CRIAR, ATUALIZAR, LISTAR E DELETAR);
* Análise de testes cobrindo todos verbos;
* Matriz de rastreabilidade atualizada;
* Automação de testes baseado na análise realizada;

**Cenários de testes**

* Rota de Login

CT001, CT002, CT003

* Rota de usuário

- Criar usuário

CT004, CT006, CT007, CT008, CT009, CT010, CT011, CT014, CT015, CT017

-Listar usuários

CT005

-Atualizar usuários

CT012, CT013

**Casos de teste**

CT001 - Testar login com credenciais válidas;

CT002 - Testar login com credenciais inválidas;

CT003 – Testar fazer login com credenciais inexistentes;

CT004 – Testar os campos obrigatórios do usuário; **/ RF01**

CT005 – Testar pesquisa de usuários inexistente; / RF02?

CT006 – Testar criar user utilizando um e-mail válido já utilizado; **/ RF03**

CT007 - Testar criar user utilizando um e-mail válido não utilizado;

CT008 – Testar utilizando e-mail de provedores Gmail; **/ RF06**

CT009 - Testar utilizando e-mail de provedores Hotmail; **/ RF06**

CT010 – Testar utilizando senhas com menos de 5 caracteres; **/ RF08**

CT011 – Testar utilizando senhas com mais de 10 caracteres; **/ RF08**

CT012 - Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado; **/RF04**

CT013 - Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT; **/RF05**

CT014 - Testar se campo nome permite caracteres além de string;

CT015 - Os e-mails devem seguir um padrão válido de e-mail para o cadastro; **/RF07**

CT016 - Não deverá ser possível fazer ações e chamadas para usuários inexistentes; **/RF02**

CT017 - Testar múltiplos campos "password" no body dos usuários;

**Matriz de Rastreabilidade**

Relaciona os critérios de aceitação com os casos de testes desenvolvidos para garantir que todos os requisitos sejam cobertos.

|  |  |
| --- | --- |
| **REQUISITOS CASOS DE TESTES** | |
| **RF01 -** Os vendedores (usuários) deverão possuir os campos NOME, E-MAIL, PASSWORD e ADMINISTRADOR | **CT004 –** Testar os campos obrigatórios do usuário; |
| **RF02 -** Não deverá ser possível fazer ações e chamadas para usuários inexistentes; | **CT016 -** Não deverá ser possível fazer ações e chamadas para usuários inexistentes; |
| **RF03** - Não deve ser possível criar um usuário com e-mail já utilizado; | **CT006 –** Testar criar user utilizando um e-mail válido já utilizado; |
| **RF04** - Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado; | **CT012 -** Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado; |
| **RF05** - Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT; | **CT013 -** Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT; |
| **RF06** - Não deverá ser possível cadastrar usuários com e-mails de provedor gmail e hotmail; | **CT008 –** Testar utilizando e-mail de provedores Gmail;  **CT009 -** Testar utilizando e-mail de provedores Hotmail; |
| **RF07** - Os e-mails devem seguir um padrão válido de e-mail para o cadastro; | **CT015 -** Os e-mails devem seguir um padrão válido de e-mail para o cadastro; |
| **RF08** - As senhas devem possuir no mínimo 5 caracteres e no máximo 10 caracteres; | **CT010 –** Testar utilizando senhas com menos de 5 caracteres;  **CT011 –** Testar utilizando senhas com mais de 10 caracteres; |

**Testes candidatos a automação**

Foram selecionados os fluxos principais da aplicação, que não podem parar para serem selecionados a automação.

* Login

Descrição: Testar o login do usuário é efetuado

Valor: Reduz o tempo de teste manual.

* Cadastro

Descrição: Testar o cadastro do usuário é realizado

Valor: Reduz o tempo de teste manual

* Validação de Campos

Descrição: Verificar se os campos obrigatórios estão funcionando corretamente.

Valor: Evita erros de entrada de dados.