



SERVEREST

12
24

RELATÓRIO TÉCNICO

SERVEREST

DA EMPRESA

SERVEREST

DO AUTOR

Alexandre Monteiro

Sumário

| Tópico | Página |
|-------------------------------------|--------|
| 1. Visão Geral do Relatório..... | 3 |
| a. Introdução | |
| b. Sobre o relatório | |
| c. Equipe | |
| 2. Planejamento de Qualidade..... | 4 |
| a. Escopo e requisitos de qualidade | |
| b. Métricas de Avaliação | |
| 3. Estratégia de Testes..... | 5 |
| a. Técnicas e Abordagem | |
| 2. Casos de Teste..... | 6 |
| a. Validar Rota de login | |
| b. Validar Rota de Usuários | |
| 5. Análise dos Resultados..... | 7 |
| a. Requisitos de Qualidade | |
| 6. Melhorias..... | 10 |

Visão Geral do Relatório

INTRODUÇÃO

Serverest é uma API de e-commerce que serve a clientes finais para compras de produtos e também a clientes intermediários que cadastram seus produtos para venda, o que a torna uma espécie de hub de vendas. Este relatório técnico busca fazer uma análise qualitativa das funcionalidades da aplicação.

OBJETIVOS

O presente documento visa descrever todo o plano para testar a API de e-commerce Serverest, com os seguintes objetivos:

- Identificar as informações sobre o projeto presentes na documentação oficial
- Listar os requisitos de teste
- Desenvolver a estratégia de teste
- realizar os testes
- Avaliar o nível de qualidade do sistema
- sugerir melhorias com base nos resultados

Equipe

Alexandre Monteiro

1522462

Heitor Victor

1329511

Victor Giovanni

1522475

Planejamento de Qualidade

Escopo e Requisitos de Qualidade

O projeto em questão visa cobrir as funcionalidades de Cadastro e Login de usuários da aplicação, utilizando os requisitos de qualidade a seguir:

- Adequação
- Acurácia
- Segurança de acesso
- Maturidade
- Tratamento de erros
- Tempo de resposta
- Apreensibilidade
- Portabilidade
- Inteligibilidade
- Testabilidade

Métricas de Avaliação

As funcionalidades foram avaliadas de acordo com os requisitos citados e os testes desenvolvidos giram em torno desses critérios. Ao final do processo teremos os pontos a serem observados sobre cada aspecto da avaliação da qualidade do sistema.



Estratégia de Testes

Técnicas e Abordagem

Em todos os casos de testes desenvolvidos e realizados foram implementadas validações de respostas para verificar se as requisições estavam de acordo com as melhores práticas de desenvolvimento.

Foram utilizadas as técnicas de análise de valor limite para verificação de intervalos de valores nos testes, testes de exceção para avaliar o tratamento do sistema para casos negativos dos cenários de testes e testes de integração dos componentes para validar as regras de negócio.

Também foram avaliados os aspectos de segurança e desempenho em tempo de resposta da aplicação das funcionalidades em questão.

Os testes foram estruturados de maneira que sejam totalmente independentes uns dos outros. Todos os dados gerados nos testes, como usuários cadastrados, são excluídos ao fim do processo para não ocupar espaço no ambiente e evitar possíveis "flaky" testes, que são como falsos negativos que podem ocorrer principalmente devido ao método de geração aleatória do Postman, que pode acabar gerando um usuário novo com dados de um já existente por exemplo, pois já existiriam muitos usuários cadastrados utilizando esse recurso.

Casos de Teste

CT 01 - Validar Rota de login

- CT 01.01 - Realizar login com sucesso
- CT 01.02 - Realizar login com email vazio
- CT 01.03 - Realizar login senha inválida
- CT 01.04 - Realizar login com senha vazia
- CT 01.05 - Realizar login sem body
- CT 01.06 - Realizar login com body vazio

CT 02 - Validar Rota de Usuários

- CT 02.01 - Listar usuários
- CT 02.02 - Cadastrar usuário Administrador
- CT 02.03 - Listar usuário por ID
- CT 02.04 - Excluir usuário por ID
- CT 02.05 - Editar usuário cadastrado
- CT 02.06 - Cadastrar usuário Comum
- CT 02.07 - Editar usuário não cadastrado
- CT 02.08 - Cadastrar usuário com email vazio
- CT 02.09 - Cadastrar usuário com senha vazia
- CT 02.10 - Cadastrar usuário sem body
- CT 02.11 - Cadastrar usuário com body vazio
- CT 02.12 - Cadastrar usuário já cadastrado
- CT 02.13 - Listar usuários por id (usuário não encontrado)
- CT 02.14 - Editar usuário com email já cadastrado
- CT 02.15 - Excluir usuário com id inválido
- CT 02.16 - Excluir usuário com carrinho cadastrado
- CT 02.17 - Cadastrar usuário sem aspas no body
- CT 02.18 - Cadastrar usuário com senha inválida

Análise dos Resultados

Requisitos de Qualidade

Através dos testes foram validados os aspectos acordados do relatório e a seguir os resultados da avaliação.

ADEQUAÇÃO

A aplicação consegue atender ao que é proposto nas funcionalidades de cadastro e login de usuários, tendo dois tipos de usuários: administrador e usuário comum com suas finalidades bem definidas.

ACURÁCIA

Numa visão geral as funcionalidades testadas geram a maioria dos resultados corretamente, porém o caso de cadastro de senha precisa ser revisto (Ver tópico Melhorias).

SEGURANÇA DE ACESSO

O sistema conta com a implementação de tokens de autenticação que garantem um controle de acesso adequado da aplicação.

Análise dos Resultados

MATURIDADE

Em todos os testes realizados nas funcionalidades o sistema não apresentou falhas de encerramento da aplicação ou travamentos.

TRATAMENTO DE ERROS

O sistema trata os casos de erro de forma consistente nas funcionalidades avaliadas, apenas foram recomendadas pequenas adequações de status code de retorno em certos casos de testes (Ver tópico Melhorias).

TEMPO DE RESPOSTA

Nos testes realizados o tempo de respostas nas operações foi satisfatório com menos de 1 segundo de tempo de espera.

PORTABILIDADE

O sistema é de fácil configuração para ser implementado em outros ambientes, bastando apenas inicializar a API com comandos npx (requer nodeJS), mas também disponibiliza opções com docker.

Análise dos Resultados

INTELIGIBILIDADE

O sistema é de fácil compreensão, com a documentação especificando e exemplificando suas funcionalidades de forma clara.

APREENSIBILIDADE

Sistema de fácil aprendizado, suas funções e elementos de utilização são simples e a maneira como operá-los.

TESTABILIDADE

Por ter um Swagger online e com ele uma documentação acessível com suas especificações e de fácil implementação, o sistema tem um bom nível de testabilidade.

Melhorias

Critério de senha para cadastro

CT 02.18 - CADASTRAR USUÁRIO COM SENHA INVÁLIDA

Esse caso específico é um defeito na regra de negócios. Foi observado que a API não possui um critério mínimo para cadastro de senhas e como consequência não existe o caso de teste em questão. Esse ponto é preocupante no quesito segurança de conta por dar a possibilidade de geração de senhas muito simples e fáceis de serem quebradas por crackers, prejudicando toda a aplicação e seus clientes. Então, ao tentar um cadastro de senha considerada inválida nos padrões gerais a API realiza o cadastro com sucesso, retornando um status code 201.

RECOMENDAÇÃO

Criar um padrão mínimo para cadastro de senhas e sua respectiva resposta HTTP que seria status code 422 o mais recomendado com a mensagem "Senha não atende aos requisitos mínimos de cadastro".

Melhorias

Melhorias em respostas HTTP

CT 02.13 - LISTAR USUÁRIOS POR ID (USUÁRIO NÃO ENCONTRADO)

Este caso de teste retorna um status code 400, que é mais genérico para um caso em que o ID buscado não foi encontrado.

RECOMENDAÇÃO

Substituir o status code 400 pelo status code 404 de não encontrado.

CT 02.15 - EXCLUIR USUÁRIO COM ID NÃO ENCONTRADO

Este caso de teste retorna o status code 200 OK para uma requisição que não encontrou o dado requisitado para exclusão e sua mensagem de retorno não especifica o fato.

RECOMENDAÇÃO

substituição do status code 200 pelo 404 de não encontrado que é mais adequado para a situação, e alteração da mensagem atual de "Nenhum registro excluído" por uma mensagem mais completa de "Usuário não encontrado. Nenhum registro excluído".