



COMPASS.UOL

PLANO DE TESTE

Setembro, 2022
Regina Azzi

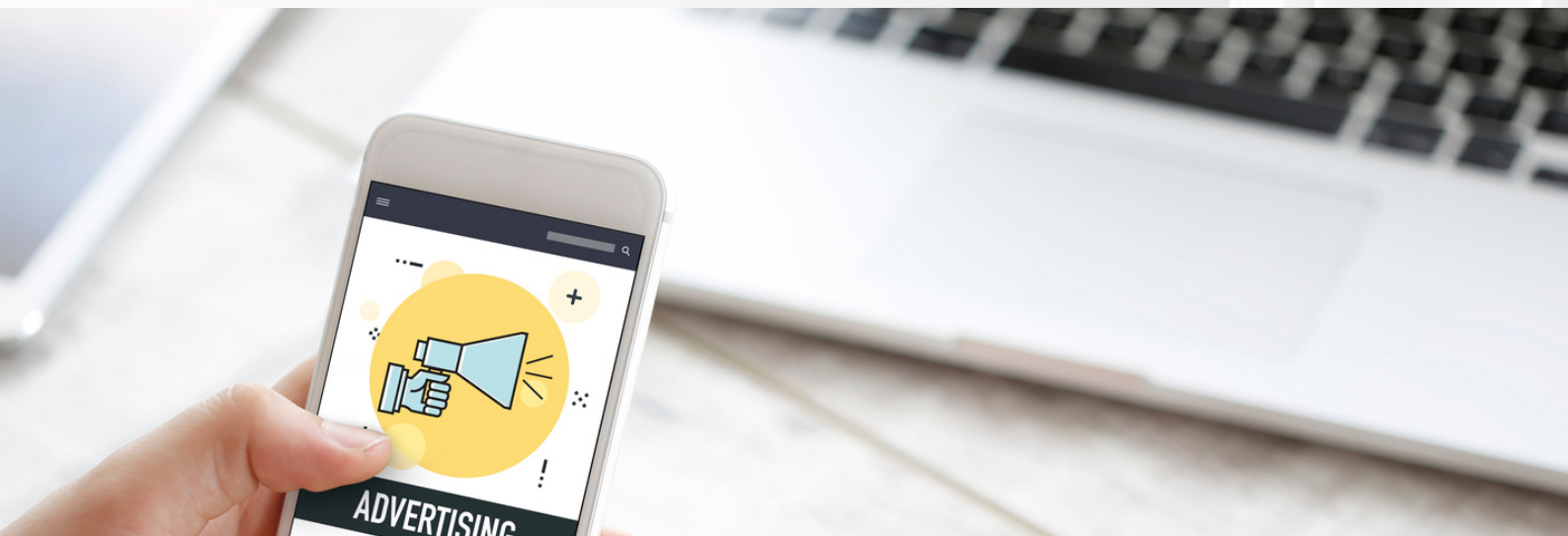
Introdução



Este plano de teste descreve o planejamento para testar as quatro rotas da API ServeRest.

O método de teste adotado, dará cobertura para o back-end com teste de integração e automação para as rotas prioritárias da API.

Objetivo



Este documento descreve o plano para testar as rotas da API Serverest, que suporta os seguintes objetivos:

- Identificar possíveis falhas na documentação da API;
- Recomendar possíveis melhorias;
- Listar os requisitos de testes;

Escopo

Este Plano de Teste descreve os testes de integração que serão conduzidos em ambiente de produção.

As seguintes rotas estão no escopo dos testes:

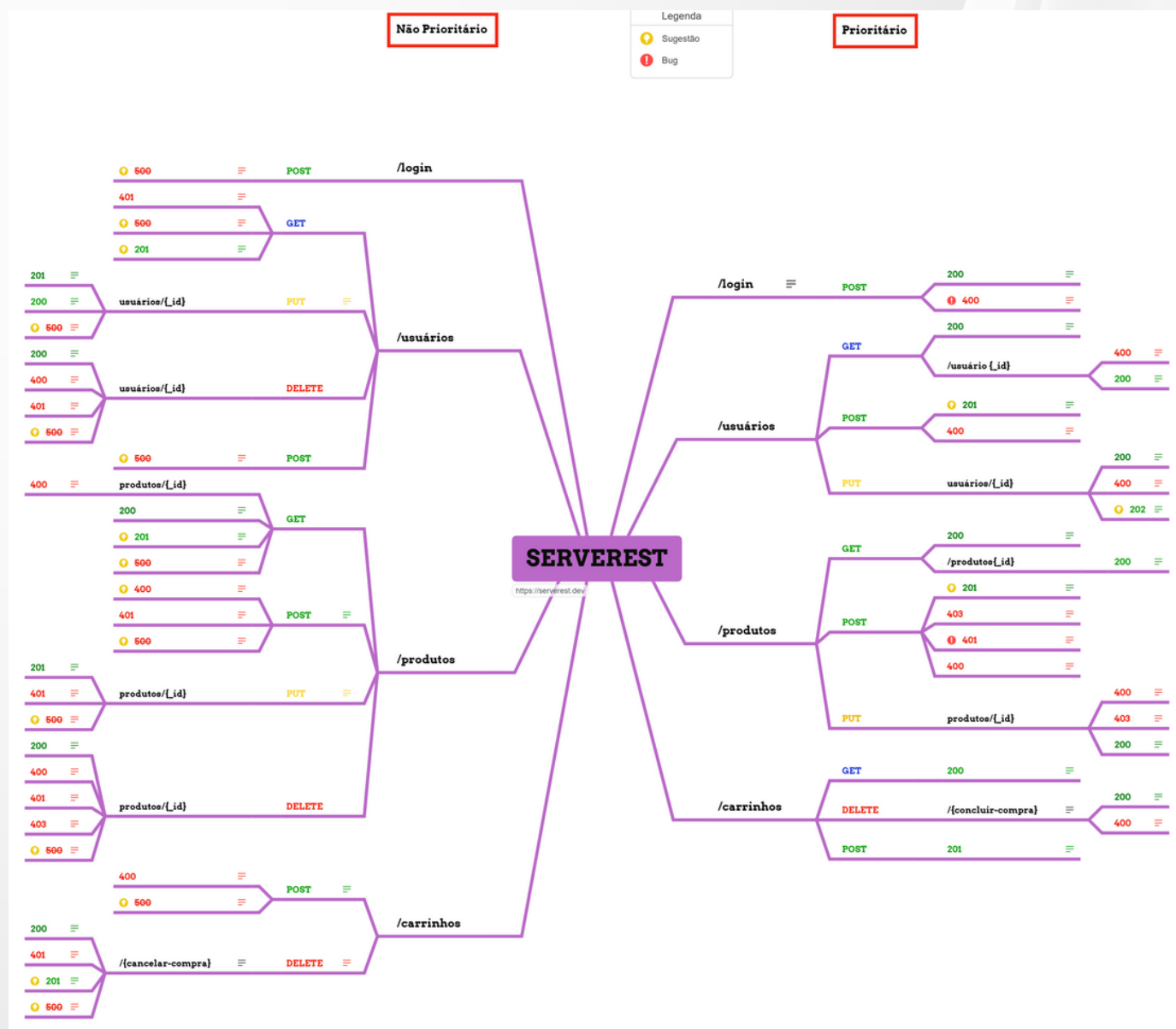
- Área administrativa (login, cadastro de produto);
- Cadastro (usuário/cadastrar usuários);
- Login Convencional (login/realizar login);
- Itens fora fluxo de compra (excluir produto, excluir usuário, editar produto etc.);
- Itens dentro do fluxo de compra (login, cadastrar usuário, listar produto, carrinho);

Para todos os itens citados acima, vão existir as respectivas validações das rotas de API.

Mapa Mental

O mapa mental exhibe todas as validações necessárias, interligadas de forma lógica.

Abaixo apresentamos o mapa mental da aplicação escopo:



Suíte de casos de teste

Lista de casos de teste para homologar o back-end da aplicação:

- CT01 - Validar login com sucesso;
- CT02 - Validar login com servidor fora do ar;
- CT03 - Validar login com e-mail e senha em branco;
- CT04 - Validar login com e-mail inválido;
- CT05 - Validar login com e-mail e senha com um caractere;
- CT06 - Validar listar usuário com servidor fora do ar;
- CT07 - Validar listar usuários com sucesso;
- CT08 - Validar listar usuários por data de nascimento;
- CT09 - Validar cadastrar usuário com servidor fora do ar;
- CT10 - Validar cadastrar usuário informando apenas um caractere no nome e senha;
- CT11 - Validar cadastrar usuário com sucesso;
- CT12 - Validar cadastrar usuário com email utilizado em outro cadastro;
- CT13 - Validar buscar usuário por _id com servidor fora do ar;
- CT14 - Validar busca de usuário por id com sucesso;
- CT15 - Validar busca de usuário colocando id errado sem sucesso;
- CT16 - Validar excluir usuário por _id com servidor fora do ar;
- CT17 - Validar recuperação de senha;
- CT18 - Validar excluir usuário informando id inválido;
- CT19 - Validar excluir usuário com sucesso;
- CT20 - Validar excluir usuário com carrinho ativo;

Suíte de casos de teste

- CT21 – Validar editar usuário com sucesso;
- CT22 – Validar editar usuário apagando todos os dados;
- CT23 – Validar editar usuário com servidor fora do ar
- CT24 – Validar editar usuário com id em branco;
- CT25 – Validar editar usuário alterando e-mail para um e-mail já cadastrado;
- CT26 – Validar editar usuário com id errado;
- CT27 – Validar listar produtos com sucesso;
- CT28 – Validar listar produtos com servidor fora do ar;
- CT29 – Validar listar produtos mais vendidos;
- CT30 – Validar listar produtos com estoque zero;
- CT31 – Validar cadastrar produto informando dados corretos;
- CT32 – Validar cadastrar produtos com servidor fora do ar;
- CT33 – Validar cadastrar produto com nome já existente;
- CT34 – Validar cadastrar produto sem estar logado;
- CT35 – Validar cadastrar produto com um usuário administrador false;
- CT36 – Validar cadastrar produto com valor negativo;
- CT37 – Validar cadastrar produto com valor zero;
- CT38 – Validar cadastrar produto sem informar quantidade;
- CT39 – Validar cadastrar produto sem informar valor;
- CT40 – Validar cadastrar produto informando zero na quantidade;
- CT41 – Validar cadastrar produto informando número negativo na quantidade;
- CT42 – Validar cadastrar produto informando uma letra no valor;
- CT43 – Validar busca de produto por id correto;
- CT44 – Validar busca de produtos com servidor fora do ar;
- CT45 – Validar busca de produto por id inválido;
- CT46 – Validar excluir produto com sucesso;

Suíte de casos de teste

- CT47 – Validar excluir produtos com servidor fora do ar;
- CT48 – Validar excluir produto que está no carrinho;
- CT49 – Validar excluir produto sem estar logado;
- CT50 – Validar excluir produto com um usuário administrador false;
- CT51 – Validar editar produto com id inválido;
- CT52 – Validar editar produto com sucesso;
- CT53 – Validar editar produto sem token;
- CT54 – Validar editar produto com nome existente;
- CT55 – Validar editar produto com servidor fora do ar;
- CT56 – Validar lista de carrinhos cadastrados;
- CT57 – Validar listar carrinhos com servidor fora do ar;
- CT58 – Validar listar carrinho excluídos – cancelar compra;
- CT59 – Validar cadastrar carrinho com sucesso;
- CT60 – Validar cadastrar carrinhos com servidor fora do ar;
- CT61 – Validar cadastrar carrinho com produto duplicado;
- CT62 – Validar cadastrar mais de um carrinho para mesmo usuário;
- CT63 – Validar cadastrar carrinho sem token;
- CT64 – Validar busca de carrinho por id;
- CT65 – Validar busca de carrinhos com servidor fora do ar;
- CT66 – Validar busca de carrinho sem informar id;
- CT67 – Validar concluir compra com sucesso;
- CT68 – Validar concluir compra sem estar logado;
- CT69 – Validar cancelar compra com dados corretos;

Estratégia de testes



TESTE DE INTEGRAÇÃO através de acesso à documentação swagger da API, deverá ser realizada a validação das respostas das rotas, em comparação com resultados pré-determinados nos cenários de teste. Também será feito testes exploratórios na versão front da API e deverá ser implementada automação dos testes da API, utilizando o framework Cypress, com a linguagem de programação Java Script.

Testes candidatos a automação

Devem fazer parte dessa lista os fluxos principais da aplicação, que devem estar sempre com funcionamento regular, para garantir a operação da atividade fim da aplicação.

Dessa forma, abaixo listamos os cenários de teste candidatos para automação:

- CT01 – Validar login com sucesso;
- CT04 – Validar login com e-mail inválido;
- CT07 – Validar listar usuários com sucesso;
- CT11 – Validar cadastrar usuário com sucesso;
- CT12 – Validar cadastrar usuário com e-mail utilizado em outro cadastro;
- CT14 – Validar busca de usuário por id com sucesso;
- CT15 – Validar busca de usuário colocando id errado sem sucesso;
- CT21 – Validar editar usuário com sucesso;
- CT25 – Validar editar usuário alterando e-mail para um e-mail já cadastrado;
- CT27 – Validar listar produtos com sucesso;
- CT31 – Validar cadastrar produto com sucesso;
- CT33 – Validar cadastrar produto com nome já existente;
- CT35 – Validar cadastrar produto com um usuário administrador false;
- CT36 – Validar cadastrar produto com valor negativo;
- CT43 – Validar busca de produto por id correto;
- CT52 – Validar editar produto com sucesso;
- CT56 – Validar lista de carrinhos cadastrados;
- CT59 – Validar cadastrar carrinho com sucesso;
- CT61 – Validar cadastrar carrinho com produto duplicado;
- CT67 – Validar concluir compra com sucesso;

Ferramentas

- Cypress: (framework para automação dos testes back-end)
- Visual Studio Code (IDE para criação do código da automação de testes)
- GitHub (para versionamento do código da automação)
- Node. Js (para desenvolvimento de código em Javascript)
- Mochosome (para reports)
- Pacote office (para elaboração da documentação de testes)
- Microsoft Teams (para comunicação com a equipe)
- Postman (ferramenta para alguns testes manuais)
- Xmind; (para criação do mapa mental)

Equipe do projeto



- Scrum Master: Luís Augusto
- PO: Karoline Nascimento
- Analista de testes: Regina Azzi
- Testador 1: João Ricardo
- Testador 2: Vinicius Alexandre
- Testador 3: Diego Hermys
- 02 Dev. para Automação de back-end: Patrícia Ferreira e Alisson Roberto
- Arquiteto de automação: Ester Carvalho

Relatório de bug

Segue abaixo bugs encontrados

CT04 - Validar login com e-mail invalido;

Descrição: Tentar efetuar login com um e-mail que não está cadastrado.

Resultado esperado: Retornar com status code 400

Resultado obtido: Retornou status code 401

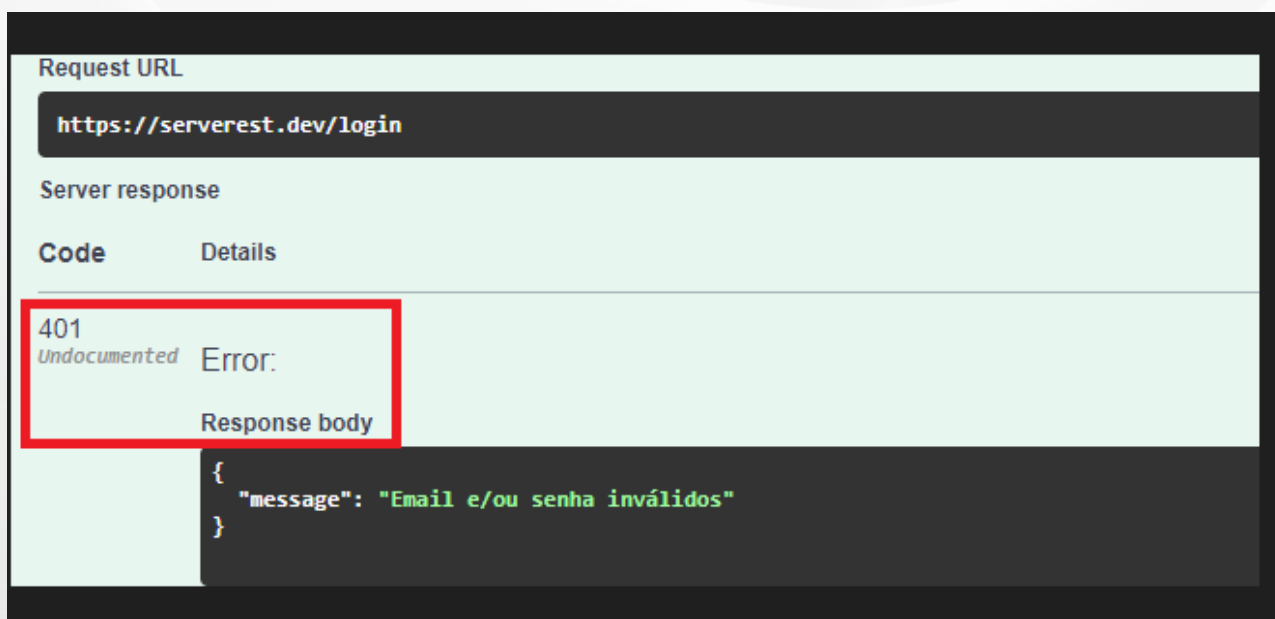
Passos para reproduzir:

Passo 1: No campo e-mail informar um email invalido

Passo 2: No campo senha informar uma senha

Passo 3 : Executar

Gravidade: baixa



Relatório de bug

CT36 – Validar cadastrar produto com valor negativo;

Descrição: Tentar cadastrar um produto informando um número negativo no campo preço.

Resultado esperado: Retornar mensagem “preço deve ser um número positivo”.

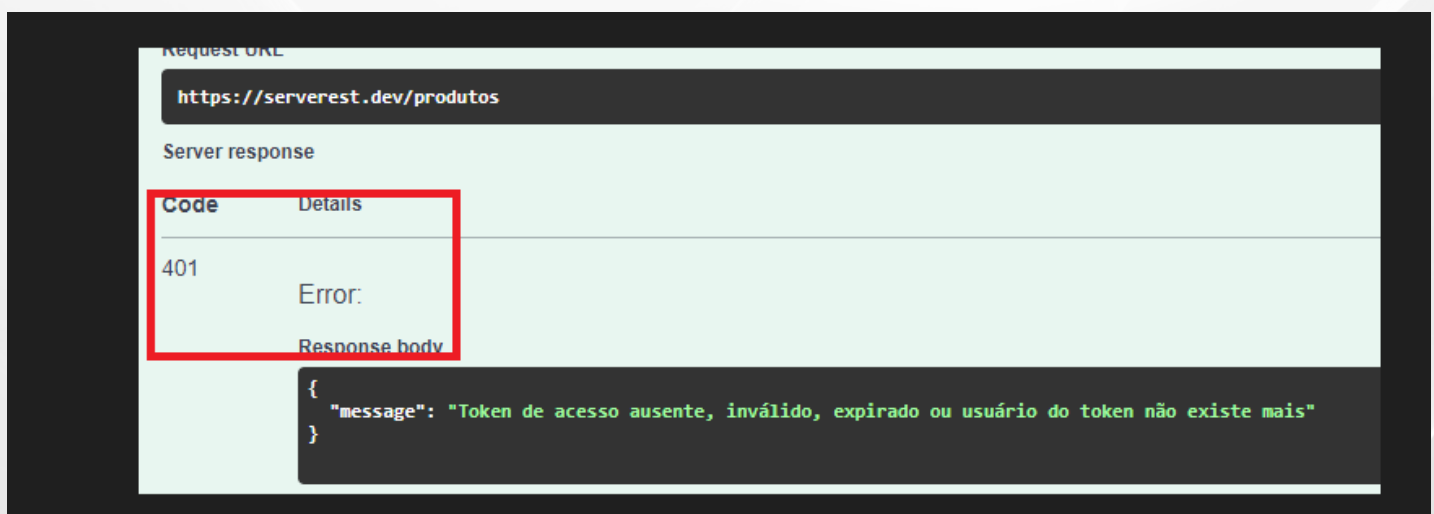
Resultado obtido: mensagem não definida (por falta de documentação).

Passos para reproduzir:

Passo 01: No campo preço informar um número negativo.

Passo 02: Executar

Gravidade: baixa (não deixa cadastrar com valor negativo, falta apenas documentar)



Relatório de melhorias

Durante a execução do plano de testes, encontrei algumas melhorias que se forem implementadas vão melhorar o funcionamento e a qualidade da API Serverest.

Segue abaixo algumas sugestões:

- Incluir uma rota para recuperação de senha;
- Incluir um limite mínimo de caractere para senha, cadastro de produto e cadastro de usuários;
- Incluir data de nascimento no cadastro de usuários;
- Incluir uma rota de relatório por data de nascimento;
- Incluir relatório de produtos mais vendidos;
- Incluir relatório de produtos com estoque zerado;
- Incluir um status para todas as rotas de servidor fora do ar;

Conclusão

O presente documento determina as atividades e processos mínimos, para a criação e execução das atividades de teste. É possível e recomendável, que conforme as atividades vão sendo realizadas, esse documento seja alimentado com novos processos que se façam necessários.

A única regra é que todas as determinações presentes nesse documento, não podem ser excluídas ou alteradas, apenas estendidas.