

PLANO DE TESTE

Compass uol

Set 2022// João Ricardo Soares da Silva



INTRODUÇÃO

Este documento apresenta o planejamento das atividades de testes da API Serverest, o qual será utilizado como base para as atividades de acompanhamento, revisão, verificação e validação do projeto, desde seu início até sua conclusão. Foi adotada a abordagem de cobertura para teste de integração e automação de rotas da API.



OBJETIVOS

Esta documentação descreve um plano para testar a aplicação da API Serverest contendo os seguintes objetivos suportados:

- Listar as coberturas de testes.
- Identificar os candidatos à automação.
- Listar tarefas e recursos para utilização no trabalho.
- Descrever as estratégias de teste que serão empregadas.
- Planejar a estimativa de esforços necessários para conclusão do projeto.



ESCOPO

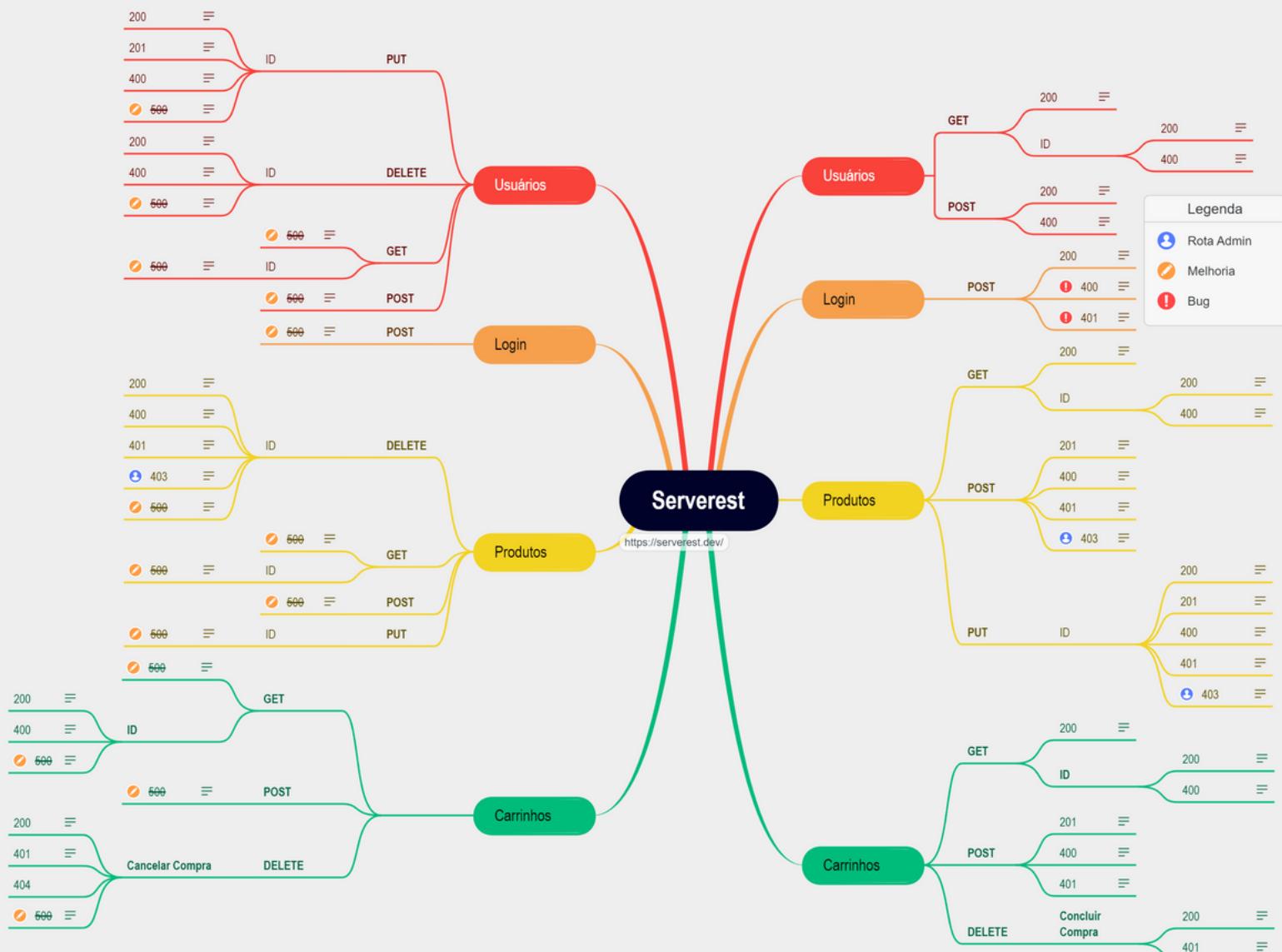
O seguinte plano de testes irá descrever os testes de integração e do sistema que serão conduzidos em ambiente de homologação, uma vez que os testes unitários já foram feitos as suas respectivas coberturas.

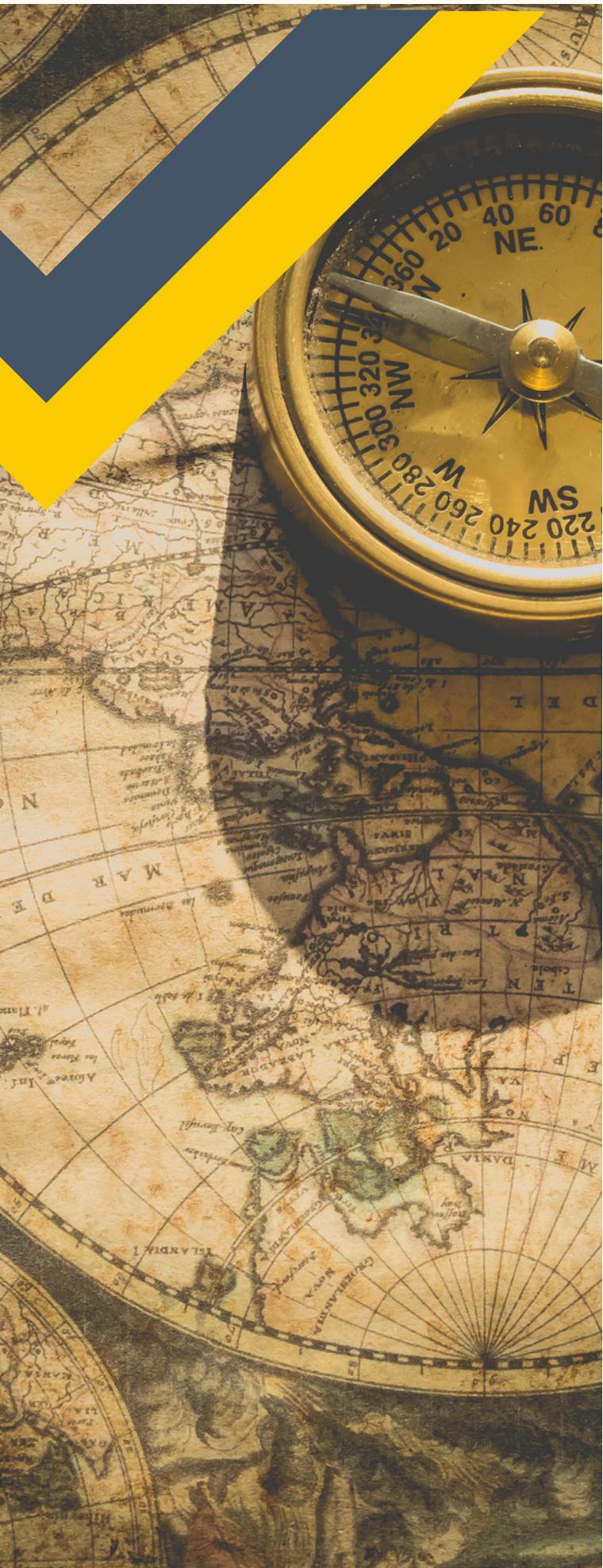
Os seguintes itens que estão no escopo dos testes são:

- **Área Administrativa** (pagina de login, pagina de cadastro de produto e etc.)
- **Cadastro** (/usuarios, Login/Cadastro)
- **Login** (login)
- **Itens fora do fluxo de compra** (excluir usuário, excluir produto, editar produto, editar usuário, editar carrinho e etc.)
- **Fluxo de compra** (realizar login, realizar cadastro, colocar produto no carrinho e cancelar carrinho para concluir a compra.)

Mapa mental

Segue o mapa mental que exibirá todas as validações necessárias, interligadas de forma lógica. Abaixo apresentamos o mapa mental da aplicação escopo:

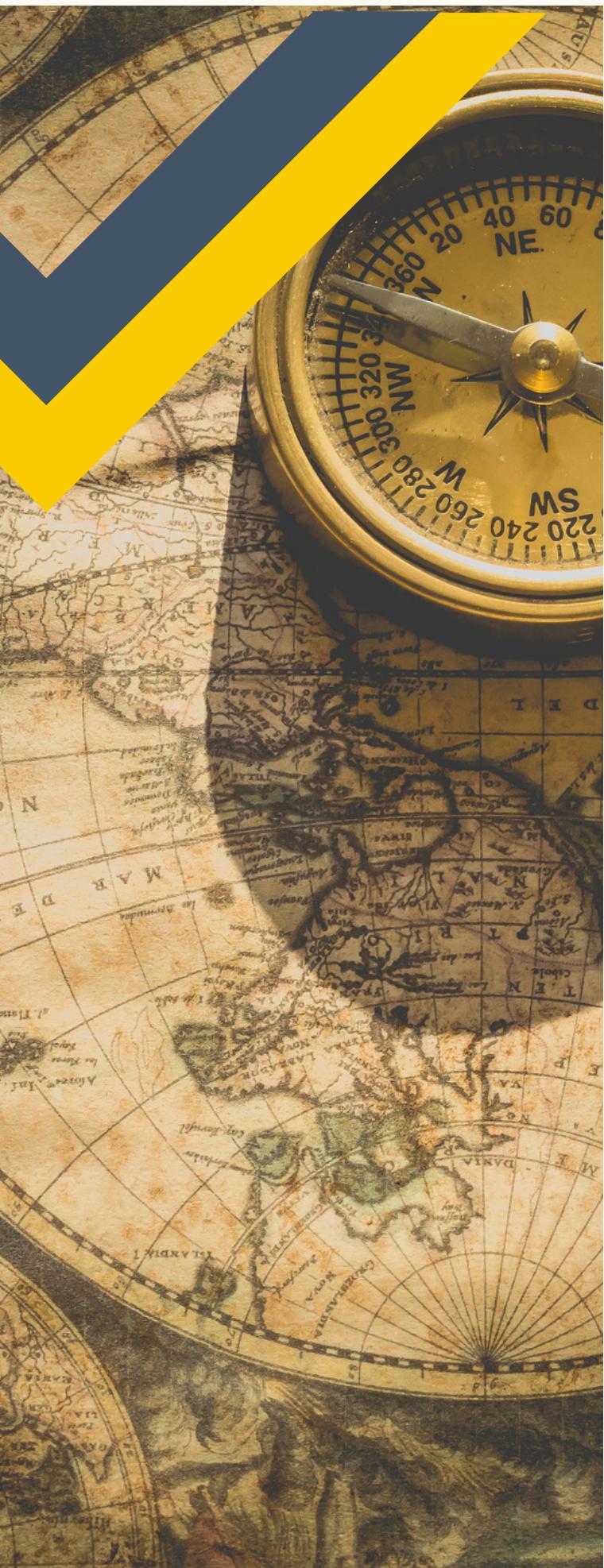




SUÍTE DE CASOS DE TESTE

Lista de casos de teste referentes as rotas da API.

- CT01 - Validar login com email e senha corretos.
- CT02 - Validar login com senha em branco.
- CT03 - Validar login com email em branco.
- CT04 - Validar login com email sem o @.
- CT05 - Validar busca de todos os usuários cadastrados.
- CT06 - Validar busca de usuário por "id" correto.
- CT07 - Validar busca de usuários por "id" incorreto.
- CT08 - Validar alteração de cadastro de usuário.
- CT09 - Validar alteração de cadastro de usuário com novo email.
- CT10 - Validar cadastro de usuário no caso de "id" não cadastrado.
- CT11 - Validar exclusão de usuário com sucesso com "id" correto.
- CT12 - Validar exclusão de usuário sem sucesso com "id" incorreto.
- CT13 - Validar cadastro de produto com itens corretos.
- CT14 - Validar cadastro de produto com produto com nome somente com caracteres especiais.
- CT15 - Validar cadastro de produtos com o valor igual a zero.
- CT16 - Validar cadastro de produtos com valor negativo.
- CT17 - Validar cadastro de produtos somente com uma letra no campo descrição.
- CT18 - Validar cadastro de produto com rota exclusiva de administrador.
- CT19 - Validar busca de todos produtos cadastrados.
- CT20 - Validar busca de produtos com 'id' correto.
- CT21 - Validar busca de produtos com 'id' incorreto.
- CT22 - Validar edição de cadastro de produto com 'id' correto.
- CT23 - Validar novo cadastro de produto no caso de 'id' não encontrado.
- CT24 - Validar edição de cadastro de produto com nome já existente.
- CT25 - Validar edição de cadastro de produto com Token ausente, inválido ou expirado.
- CT26 - Validar edição de cadastro de produto com rota exclusiva de administradores.
- CT27 - Validar exclusão de produto com sucesso.
- CT28 - Validar exclusão de produto que já faz parte de um carrinho.
- CT29 - Validar exclusão de produto com Token ausente, inválido ou expirado.
- CT30 - Validar cadastro de carrinho com sucesso.



SUÍTE DE CASOS DE TESTE

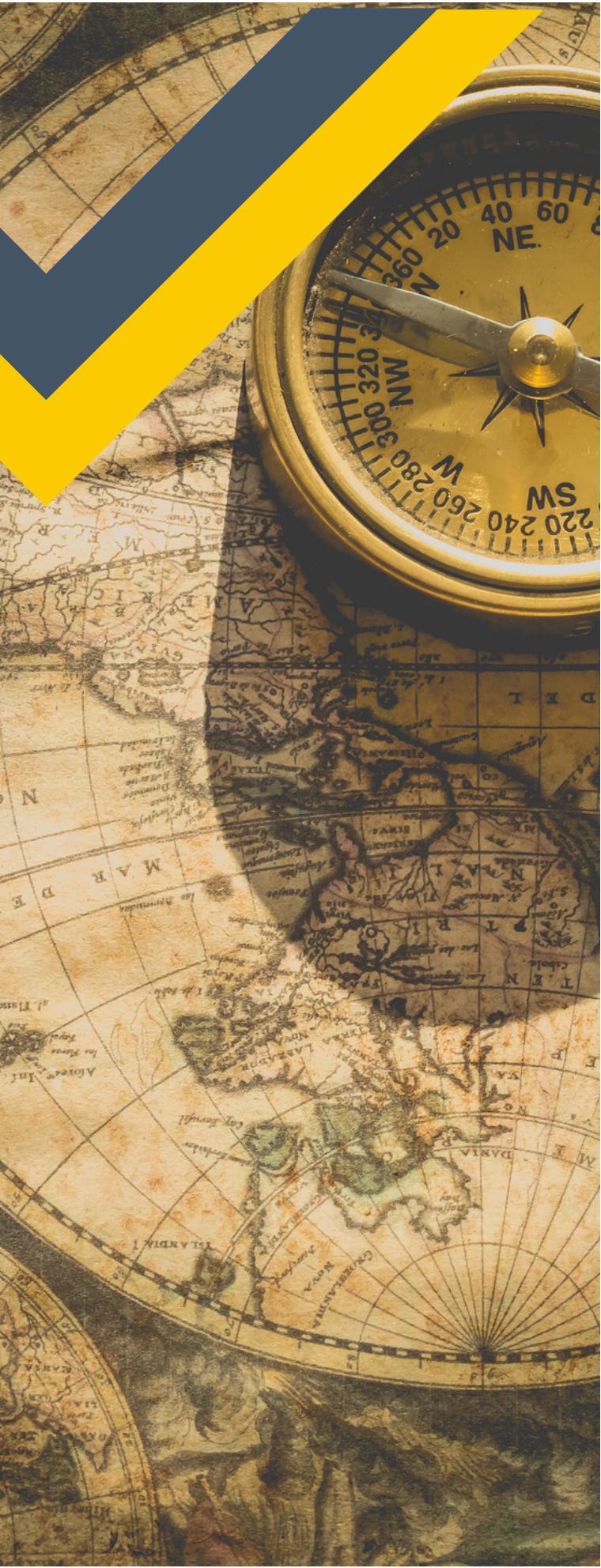
Lista de casos de teste referentes as rotas da API.

- CT31 - Validar cadastro de carrinho com produto duplicado.
- CT32 - Validar cadastro de carrinho com Token ausente, inválido ou expirado.
- CT33 - Validar busca de todos os carrinhos.
- CT34 - Validar busca de carrinhos com 'id' válido.
- CT35 - Validar busca de carrinhos com 'id' inválido.
- CT36 - Validar exclusão de carrinho com sucesso para concluir compra.
- CT37 - Validar exclusão de carrinho sem sucesso para concluir compra.
- CT38 - Validar exclusão de carrinho com sucesso para cancelar compra.
- CT39 - Validar exclusão de carrinho sem sucesso para cancelar compra.
- CT40 - Validar alteração de usuário pelo "id" com servidor fora do ar.
- CT41 - Validar exclusão de usuário pelo "id" com servidor fora do ar.
- CT42 - Validar busca de usuários com servidor fora do ar.
- CT43 - Validar busca de usuários pelo "id" com servidor fora do ar.
- CT44 - Cadastro de usuário com servidor fora do ar.
- CT45 - Validar login de usuário com servidor fora do ar.
- CT46 - Validar exclusão de produto com servidor fora do ar.
- CT47 - Validar busca de produtos com servidor fora do ar.
- CT48 - Validar busca de produtos pelo "id" com servidor fora do ar.
- CT49 - Validar cadastrar produto com servidor fora do ar.
- CT50 - Validar alteração de produto com servidor fora do ar.
- CT51 - Validar busca de carrinhos com servidor fora do ar.
- CT52 - Validar busca de carrinhos pelo "id" com servidor fora do ar.
- CT53 - Validar cadastro de carrinhos com servidor fora do ar.
- CT54 - Validar deletar carrinho com servidor fora do ar.
- CT55 - Validar cadastro de produto com Token de usuário inválido.



ESTRATÉGIA DE TESTE

TESTE DE INTEGRAÇÃO: Será utilizado a ferramenta Postman juntamente e à documentação swagger da API Severest, onde deverá ser realizada a validação das respostas das rotas, para ser comparado com resultados pré-determinados nos cenários de teste. Serão implementados automação dos teste da API, utilizando o framework Cypress, com a linguagem de programação Java Script e Node.js.



TESTES CANDIDATOS PARA A AUTOMAÇÃO

- CT01 - Validar login com email e senha corretos.
- CT02 - Validar login com senha em branco.
- CT03 - Validar login com email em branco.
- CT05 - Validar busca de todos os usuários cadastrados.
- CT06 - Validar busca de usuário por "id" correto.
- CT08 - Validar alteração de cadastro de usuário.
- CT13 - Validar cadastro de produto com itens corretos.
- CT19 - Validar busca de todos produtos cadastrados.
- CT20 - Validar busca de produtos com 'id' correto.
- CT22 - Validar edição de cadastro de produto com 'id' correto.
- CT30 - Validar cadastro de carrinho com sucesso.
- CT33 - Validar busca de todos os carrinhos.
- CT34 - Validar busca de carrinhos com 'id' válido.
- CT38 - Validar exclusão de carrinho com sucesso para cancelar compra.



FERRAMENTAS

Abaixo listamos as ferramentas que deverão ser utilizadas para as atividades de teste:

- **Jira Software** (para gerenciamento dos testes)
- **Zephyr** (extensão do Jira para criação e execução de cenários de teste)
- **Pacote Office** (para elaboração da documentação de testes)
- **Microsoft Teams** (para as comunicações);E-mail 365 (para as formalizações por e-mail)
- **Visual Studio Code** (IDE para a criação do código da automação de testes)
- **Cypress** (framework para a automação)
- **JavaScript** (linguagem de programação para realizar a automação de testes)
- **Postman** (ferramenta para interagir com as rotas de API da aplicação)
- **GitHub** (para versionamento do código da automação)
- **Node.js** (software que permite a execução do javascript)
- **Mochasome** (Ferramenta para gerar Report)



EQUIPE DE TESTES

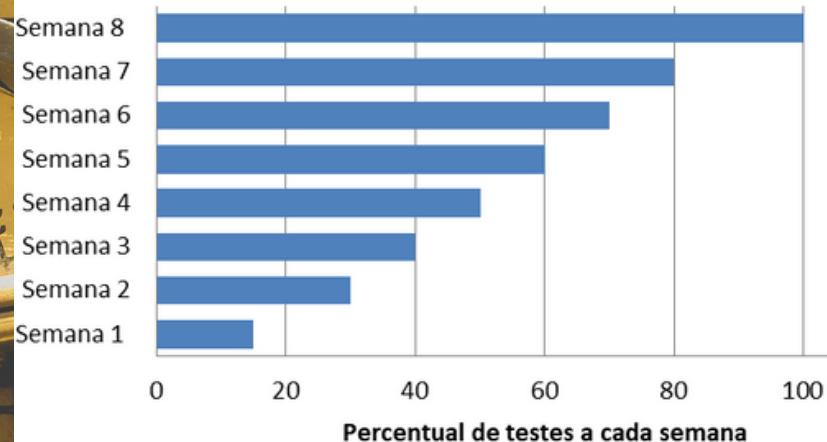
Abaixo está sendo apresentado a equipe de testes do projeto:

- **Líder de QA:** João Ricardo
- **Analista de testes:** Mário João
- **Testador 1:** Ricardo José
- **Testador 2:** Regina Azzi
- **Testador 3:** Vinicius Alexandre
- **Dev. Automação API:** Luiz Carlos
- **Arquiteto de automação:** Maria José

CRONOGRAMA

Segue abaixo cronograma referente a evolução dos testes ao longo de 8 semanas:

Evolução dos testes





MELHORIAS

Seguem abaixo seguintes melhorias identificadas na API Serverest:

Criação de **Status Codes** para os seguintes cenários:

- Validar busca de usuários com servidor fora do ar.
 - Validar busca de usuários pelo "id" com servidor fora do ar.
 - Cadastro de usuário com servidor fora do ar.
 - Validar login de usuário com servidor fora do ar.
 - Validar exclusão de produto com servidor fora do ar.
 - Validar busca de produtos com servidor fora do ar.
 - Validar busca de produtos pelo "id" com servidor fora do ar.
 - Validar alteração de produto com servidor fora do ar.
 - Validar busca de carrinhos com servidor fora do ar.
 - Validar busca de carrinhos pelo "id" com servidor fora do ar.
 - Validar cadastro de carrinhos com servidor fora do ar.
 - Validar deletar carrinho com servidor fora do ar.
 - Validar cadastro de produto com Token de usuário inválido.



RELATÓRIO DE BUG

CT03 - Validar login com e-mail e senha em branco.

Descrição: Realizar login com e-mail e senha em branco.

Passos para reproduzir:

Passo 01: Acessar a API (<https://serverest.dev>)

Passo 02: No menu Post/Login clicar em "Try it out"

Passo 03: No campo e-mail deixar em branco

Passo 04: No campo senha não deixar em branco

Passo 05: Clicar na opção Execute

Resultado esperado: Retornar com status code 400.

Resultado obtido: Está retornando com status code 401.

Gravidade: Baixa

Code	Details
401 <small>Undocumented</small>	Error: Response body <pre>{ "message": "Email e/ou senha inválidos" }</pre>

Conclusão

Neste documento é determinado 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 extendidas.