

CHALLENGE SPRINT 5 - TECH

# Automação de Testes de API com cypress

| PLANO DE TESTE  
API ServeRest

# INTRODUÇÃO

O presente plano de teste descreve o planejamento para criar e implementar as validações de cobertura para a API ServeRest. Este documento deve estar disponível para todos os stakeholders do projeto (gerência, desenvolvedores, clientes, equipe de testes, etc). A abordagem de teste adotada, dará cobertura para o back-end (com testes de integração e automação para as rotas da API).

# OBJETIVOS

Este documento descreve o plano para testar a API <https://serverest.dev/> que suporta os seguintes objetivos:.

- Autenticar usuário através do login, para montar um carrinho, e se for administrador gerenciar produtos;
- Gerenciar os usuários, consultar dados para login, e cadastrar administrador;
- Consultar produtos cadastrados, ou como administrador, gerenciar os produtos da loja;
- Consultar os carrinhos cadastrados.

# ESCOPO

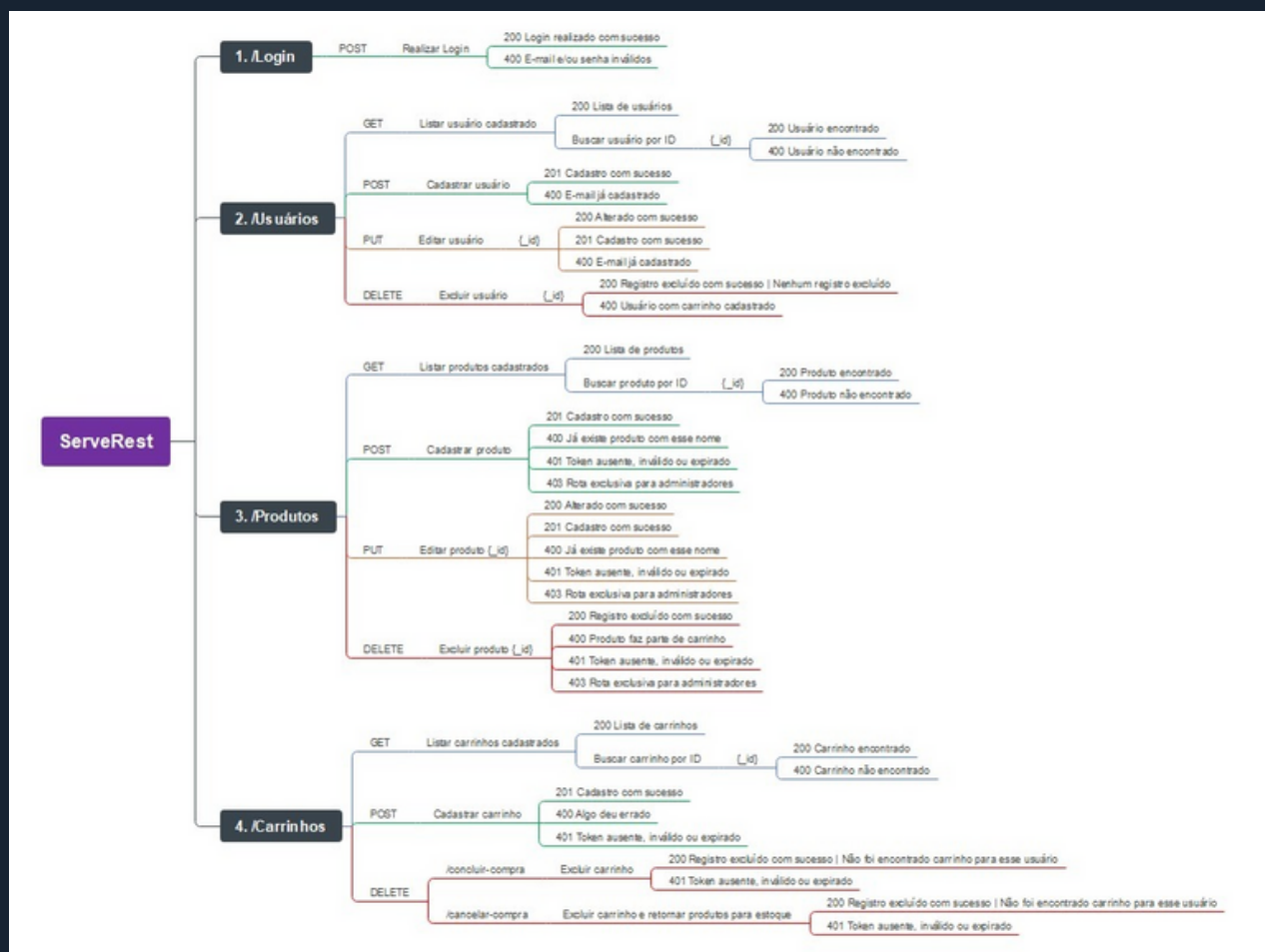
Este Plano de Teste descreve os testes de integração e rotas da API que serão conduzidos em ambiente de produção. Assume-se que o teste de unidade já forneceu uma cobertura extensiva do código fonte.

Os seguintes itens estão no escopo dos testes:

- Login (Realizar login).
- Usuários (Listar usuários cadastrados, cadastrar usuário, buscar usuário por ID, excluir usuário e editar usuário).
- Produtos (Listar produtos cadastrados, cadastrar produto, buscar produto por ID, excluir produto e editar produto).
- Carrinhos (Listar carrinhos cadastrados, cadastrar carrinho, buscar carrinho por ID, excluir carrinho e cancelar carrinho de compra e retornar produtos para estoque).

# MAPA MENTAL

O mapa mental exhibe todas as validações necessárias, interligadas de forma lógica. Abaixo apresentamos o mapa mental da aplicação escopo (API):



# SUÍTE DE CASOS DE TESTE

Lista de casos de teste para validar as rotas da API:

- CT01 - Validar login realizado com sucesso;
- CT02 - Validar listagem de usuários;
- CT03 - Validar cadastro de usuário com sucesso;
- CT04 - Validar e-mail já cadastrado;
- CT05 - Validar busca de usuário;
- CT06 - Validar busca de usuário não encontrado;
- CT07 - Validar exclusão de registro com sucesso;
- CT08 - Validar usuário com carrinho cadastrado;
- CT09 - Validar usuário alterado com sucesso;
- CT10 - Validar usuário cadastrado com sucesso;
- CT11 - Validar usuário com e-mail já cadastrado;
- CT12 - Validar listagem de produtos;
- CT13 - Validar cadastro de produto com sucesso;
- CT14 - Validar nome de produto já existente;
- CT15 - Validar token ausente, inválido ou expirado;
- CT16 - Validar rota exclusiva para administradores;

## SUÍTE DE CASOS DE TESTE

- CT17 - Validar rota de produtos exclusiva para administradores;
- CT18 - Validar produto encontrado com sucesso;
- CT19 - Validar produto não encontrado;
- CT20 - Validar registro de produto excluído com sucesso;
- CT21 - Validar nenhum registro excluído;
- CT22 - Validar se o produto faz parte do carrinho;
- CT23 - Validar rota de produtos exclusiva para administradores;
- CT24 - Validar produto alterado com sucesso;
- CT25 - Validar produto cadastrado com sucesso;
- CT26 - Validar se *já existe produto com o mesmo nome*;
- CT27 - Validar rota de cadastro de produto, exclusiva para administradores;
- CT28 - Validar cadastro de produto com sucesso;
- CT29 - Validar lista de carrinhos;
- CT30 - Validar cadastro de carrinho por usuário;
- CT31 - Validar se algo deu errado no cadastro de carrinho;

## SUÍTE DE CASOS DE TESTE

- CT32 - Validar carrinho encontrado por ID;
- CT33 - Validar carrinho não encontrado por ID;
- CT34 - Validar registro de carrinho excluído com sucesso;
- CT35 - Validar carrinho não encontrado para o usuário;
- CT36 - Validar registro excluído com sucesso para cancelar compra;



# ESTRATÉGIA DE TESTE

Níveis de teste:

**TESTE DE INTEGRAÇÃO:** Através da utilização da ferramenta POSTMAN, e acesso à documentação Swagger da API, deverá ser realizada a validação da aplicação, em comparação com resultados pré-determinados nos cenários de teste. Também deverá ser implementada automação dos testes da API através da utilização da ferramenta Cypress.

## PRIORIDADES

Será priorizado as rotas da API de maior interesse ao usuário final, em ordem de relevância, à saber: Login e cadastro do usuário, busca e listagem de usuários, busca e listagem de produtos, busca de carrinho por usuário e cancelamento da compra.

# CANDIDATOS PARA AUTOMAÇÃO

Deverá ser implementada a cobertura prioritária dos casos de testes da API, para aprendizado, estudando e analisando todas as suítes de casos de testes de forma sistemática, utilizando a ferramenta Postman para Análise e mapeamento da API, com a linguagem de programação Java Script. E priorizar os estudos da ferramenta de Cypress e frameworks da aplicação, para consolidar os estudos de Estrutura e comandos personalizados, Service Object, Massa de dados, Teste de contrato e Reports automatizados.

Candidatos para automação de Back-end:

- CT01 - Validar a buscar todos os usuários cadastrados na Serverest;
- CT02 - Validar que não deve postar um novo usuário administrador existente;
- CT03 - Validar a realização de login com sucesso;
- CT04 - Validar a buscar e salvar um usuário em um arquivo json;
- CT05 - Validar ao buscar o usuário de um arquivo json;
- CT06 - Validar a realização de login com sucesso;
- CT07 - Validar a busca de todos os produtos cadastrados;
- CT08 - Validar ao postar um novo produto com sucesso;

# FERRAMENTAS

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

- Canva (para a elaboração da documentação de testes);
- Microsoft Teams (para as comunicações);
- XMind (Para versionamento do mapa mental);
- JavaScript (a linguagem de programação utilizada para realização de automação de testes);
- Node.js (para a execução de códigos JavaScript fora do navegador web, faça a compilação deles e os execute);
- Cypress (para automatização de testes);
- Faker (para gerar dados fake dos testes);
- Ajv (para validar os schemas para teste de contrato);
- Mocha (para reports automatizados);
- Swagger (para o versionamento da documentação);
- Postman (uma ferramenta para analisar a documentação dos arquivos Swagger e interagir com as rotas da API e automatização de testes);
- GitHub (para o versionamento de código, apresentação do projeto, e versionamento da documentação dos frameworks);

# OBSERVAÇÕES COMPLEMENTARES

- ServeRest 2.26.2 é uma API REST gratuita que simula uma loja virtual com intuito de servir de material de estudos de testes de API.
- O autor do projeto se chama Paulo Gonçalves e vive em Portugal.
- O projeto conta com vários colaboradores e apoio financeiro que são doados para a ONG Todas as letras.

## 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 extendidas.

**Esse Plano de Testes deve estar disponível e acessível para todos os stakeholders do projeto.**