

**PROGRAMA DE BOLSAS – QUALITY & TEST AUTOMATION**

**PLANO DE TESTES**

SERVEREST

Marcos Vinícius Bedin Embarach

2023

**Sumário**

[1 Introdução 3](#_Toc147593442)

[2 Objetivos 3](#_Toc147593443)

[3 Escopo 3](#_Toc147593444)

[4 Mapa Mental 4](#_Toc147593445)

[5 Suíte de Casos de Teste 4](#_Toc147593446)

[5.1 Suíte de testes 1 – Autenticação/Login 4](#_Toc147593447)

[5.2 Suíte de Testes 2 - Usuários 5](#_Toc147593448)

[5.3 Suíte de testes 3 – Produtos 6](#_Toc147593449)

[5.4 Suíte de testes 4 - Carrinho de Compras 7](#_Toc147593450)

[6 Estratégia de Teste 7](#_Toc147593451)

[6.1 Testes de Unidade 7](#_Toc147593452)

[6.2 Testes Funcionais 8](#_Toc147593453)

[7 Prioridades 8](#_Toc147593454)

[8 Candidatos para automação 9](#_Toc147593455)

[9 Ferramentas 9](#_Toc147593456)

[10 Cronograma 9](#_Toc147593457)

[10.1 Etapa 1: Preparação (25/09/2023 a 28/09/2023) 9](#_Toc147593458)

[10.2 Etapa 2: Testes de unidade e Automatização (de 29/09/2023 a 04/10/2023) 10](#_Toc147593459)

[10.3 Etapa 3: Relatórios e Correções (de 05/10/2023 a 06/10/2023) 10](#_Toc147593460)

# Introdução

Este plano de teste é elaborado para garantir que a API ServeRest funcione de acordo com as expectativas e requisitos, fornecendo informações precisas e consistentes em resposta às solicitações.

O ServeRest é uma plataforma de simulação de loja virtual por meio de uma API REST, desenvolvida para fins de estudo e teste de APIs. Este plano de teste tem como objetivo validar a funcionalidade da API e identificar eventuais problemas ou desvios do comportamento esperado.

# Objetivos

Este plano de teste foi feito para garantir a funcionalidade e a confiabilidade das operações relacionadas ao login, gerenciamento de usuários, produtos e carrinhos na API ServeRest.

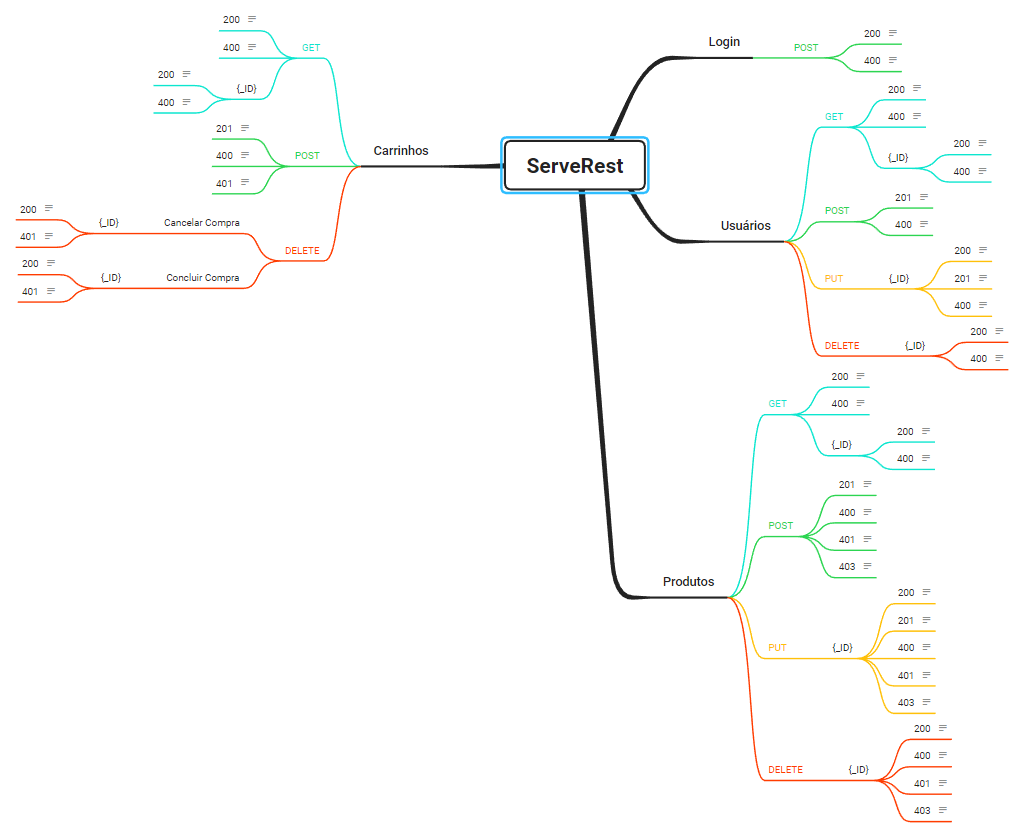
O objetivo é assegurar que todas as funcionalidades relacionadas a autenticação, gerenciamento de usuários, produtos e carrinhos estejam em conformidade com as especificações e funcionem de maneira confiável. Todos os problemas identificados durante os testes serão documentados e comunicados à equipe de desenvolvimento para correção.

# Escopo

Este plano de testes abrange as seguintes áreas e atividades:

* Teste de Autenticação/Login
* Teste de Gerenciamento de Usuários
* Teste de Gerenciamento de Produtos
* Teste de Gerenciamento de Carrinhos

# Mapa Mental



# Suíte de Casos de Teste

## Suíte de testes 1 – Autenticação/Login

CT\_001: Realizar Login válido;

CT\_002: Tentativa de Login com senha vazia;

CT\_003: Tentativa de Login com e-mail vazio;

CT\_004: Tentativa de Login com dados vazios;

CT\_005: Tentativa de Login com dados inválidos;

CT\_006: Tentativa de Login com e-mail inválido;

CT\_007: Tentativa de Login com senha invalida;

## Suíte de Testes 2 - Usuários

CT\_008: Listar os Usuários cadastrados;

CT\_009: Buscar Usuário por Id;

CT\_010: Buscar Usuário por Id inexistente;

CT\_011: Cadastrar Usuário;

CT\_012: Cadastrar Usuário sem nome;

CT\_013: Cadastrar Usuário sem e-mail;

CT\_014: Cadastrar Usuário sem senha;

CT\_015: Cadastrar Usuário sem ser administrador;

CT\_016: Cadastrar Usuário vazio;

CT\_017: Cadastrar Usuário com Gmail;

CT\_018: Cadastrar Usuário com Hotmail;

CT\_019: Cadastrar Usuário com senha muito pequena;

CT\_020: Cadastrar Usuário com senha muito grande;

CT\_021: Deletar Usuário;

CT\_022: Deletar Usuário invalido;

CT\_023: Deletar Usuário vazio;

CT\_024: Editar Usuário;

CT\_025: Editar Usuário invalido;

CT\_026: Editar Usuário em branco;

CT\_027: Editar Usuário não cadastrado;

## Suíte de testes 3 – Produtos

CT\_028: Listar produtos;

CT\_029: Listar Produto por Id;

CT\_030: Listar produto por Id invelido;

CT\_031: Cadastrar Produto;

CT\_032: Cadastrar Produto sem o token;

CT\_033: Cadastrar Produto vazio;

CT\_034: Cadastrar Produto duplicado;

CT\_035: Cadastrar Produto sem ser administrador;

CT\_036: Deletar Produto;

CT\_037: Deletar Produto vazio;

CT\_038: Deletar Produto invalido;

CT\_039: Deletar Produto no carrinho;

CT\_040: Editar Produto;

CT\_041: Editar Produto vazio;

CT\_042: Editar Produto sem o token;

CT\_043: Editar Produto não cadastrado;

CT\_044: Editar Produto duplicado;

## Suíte de testes 4 - Carrinho de Compras

CT\_045: Listar Carrinhos cadastrados;

CT\_046: Listar Carrinho por Id;

CT\_047: Listar Carrinho por Id invalido;

CT\_048: Cadastrar Carrinho;

CT\_049: Cadastrar Carrinho sem o token;

CT\_050: Cadastrar Carrinho vazio;

CT\_051: Concluir compra;

CT\_052: Concluir compra no Carrinho;

CT\_053: Concluir compra sem o token;

CT\_051: Cancelar compra;

CT\_055: Cancelar compra no Carrinho;

CT\_056: Cancelar compra sem o token;

# Estratégia de Teste

## Testes de Unidade

O plano de teste de unidade abrange uma ampla gama de casos de teste relacionados à funcionalidade da API. Esses casos de teste visam verificar a implementação correta das operações de autenticação/login, gerenciamento de usuários, produtos e carrinhos de compras.

No total, são 56 casos de teste que abrangem cenários como login válido, tentativas de login com dados em branco, cadastro de usuários com diferentes configurações, manipulação de produtos e carrinhos de compras, entre outros.

Cada caso de teste será executado de forma independente, com dados de teste apropriados, e os resultados serão verificados para garantir que a API funcione conforme especificado.

## Testes Funcionais

Os testes funcionais serão realizados para verificar a validação dos campos dos formulários em várias situações, incluindo:

* **Dados válidos**: Os campos serão preenchidos com dados válidos de acordo com as regras de validação estabelecidas.
* **Dados inválidos**: Os campos serão preenchidos com dados que não atendem às regras de validação estabelecidas, como formatos incorretos ou informações inconsistentes.
* **Campos em branco**: Os campos serão deixados em branco para verificar se as mensagens de erro apropriadas são exibidas.

Além disso, os testes funcionais cobrirão casos de teste específicos que abrangem todas as funcionalidades da autenticação/login da API.

Os resultados dos testes de compatibilidade também serão documentados, indicando quaisquer problemas encontrados em navegadores específicos.

# Prioridades

Priorizamos os seguintes casos de teste, que estão diretamente relacionados com a principal função do site, que é a jornada de compra:

CT\_011: Cadastrar Usuário;

CT\_001: Realizar Login válido;

CT\_048: Cadastrar Carrinho;

CT\_052: Concluir compra no Carrinho;

Esses testes são priorizados nessa ordem para simular a jornada de compra completa do usuário, garantindo que problemas críticos que possam dificultar ou impedir a conclusão da compra sejam identificados e corrigidos o mais cedo possível.

# Candidatos para automação

Todos os testes são candidatos à automação.

# Ferramentas

* Postman
* JavaScript

# Cronograma

## Etapa 1: Preparação (25/09/2023 a 28/09/2023)

* Definição dos casos de teste;
* Preparação do ambiente de teste;
* Instalação e configuração das ferramentas de teste.

## Etapa 2: Testes de unidade e Automatização (de 29/09/2023 a 04/10/2023)

* Automação das Suítes de Testes;
* Execução dos Testes Automatizados.

## Etapa 3: Relatórios e Correções (de 05/10/2023 a 06/10/2023)

* Documentação dos resultados dos testes;
* Identificação e correção de problemas encontrados durante os testes.

# Observações Complementares

Conquistamos com sucesso uma cobertura de testes de 100% para todas as funcionalidades do projeto. Isso significa que todos os casos de teste foram executados, abrangendo todos os caminhos possíveis no código-fonte e validando amplamente as funcionalidades do sistema/API.