

# Plano de Testes



PB Compass - Sprint 7

Marina Webber



# Estratégia de testes

A estratégia de teste para a API ServeRest abrange uma abordagem, apoiada por ferramentas como Postman e Swagger, para garantir que as funções estejam de acordo como o esperado, levando em conta os critérios de aceite do Challenge.

Ela engloba testes manuais, testes de integração e testes funcionais, focando nos principais endpoints da API, principalmente nos métodos CRUD para usuários e POST para login. Porém dentro de cada um dos cenários foi utilizado, através das pré requestes e dos testes, endpoints de diferentes métodos:

- Rotas Login => POST/usuarios - DELETE/usuarios.
- Rotas Usuários => POST/login , POST/produtos-, POST/carrinhos,DELETE/produtos, DELETE/carrinhos

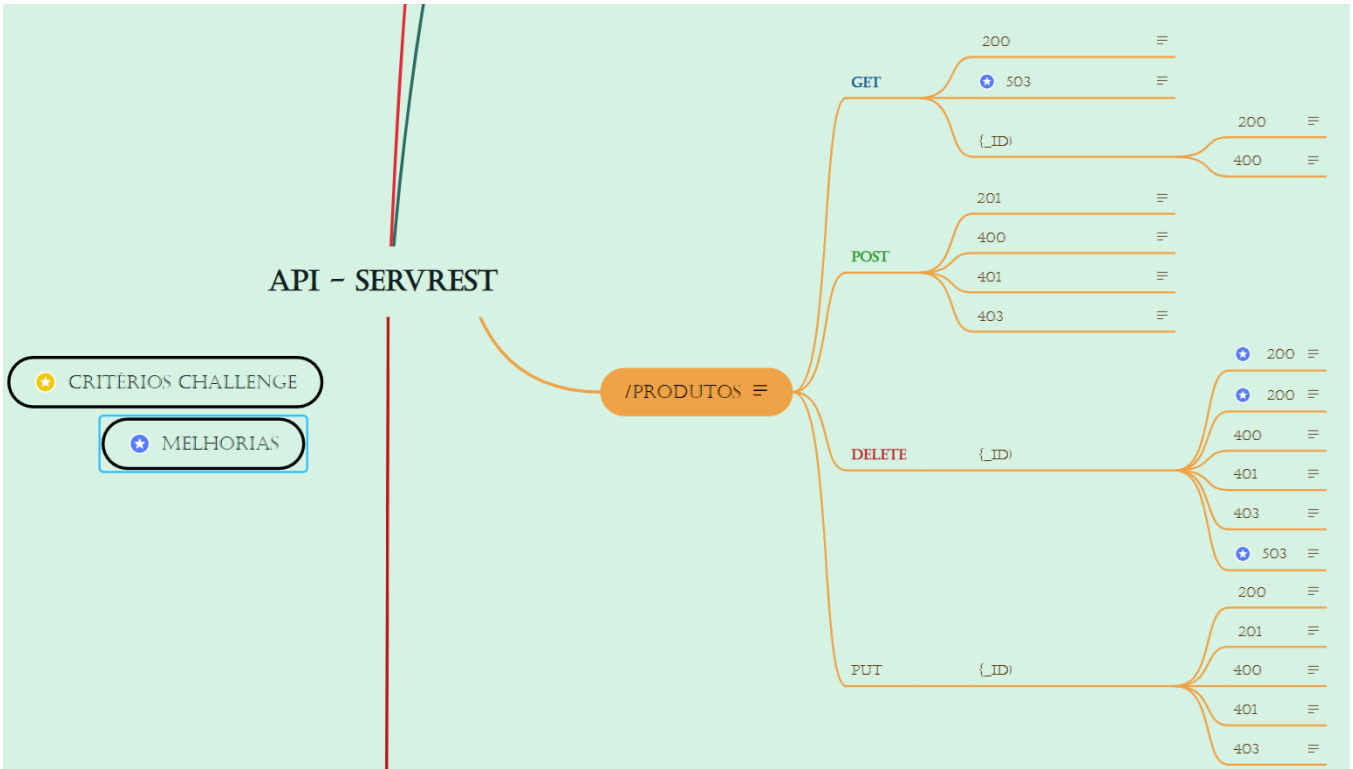
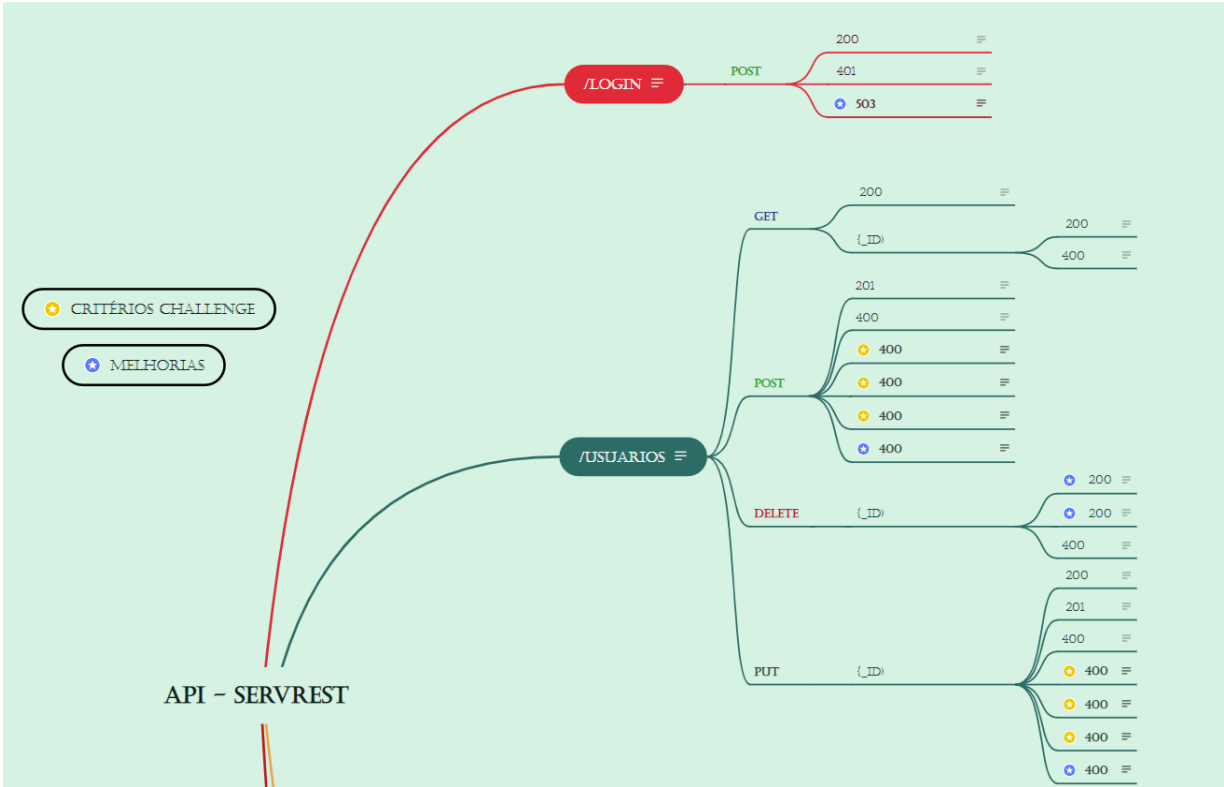
Será utilizado o mapa mental como guia para identificar cenários de teste, priorizando especialmente aqueles que são críticos para garantir a segurança e a integridade dos dados.

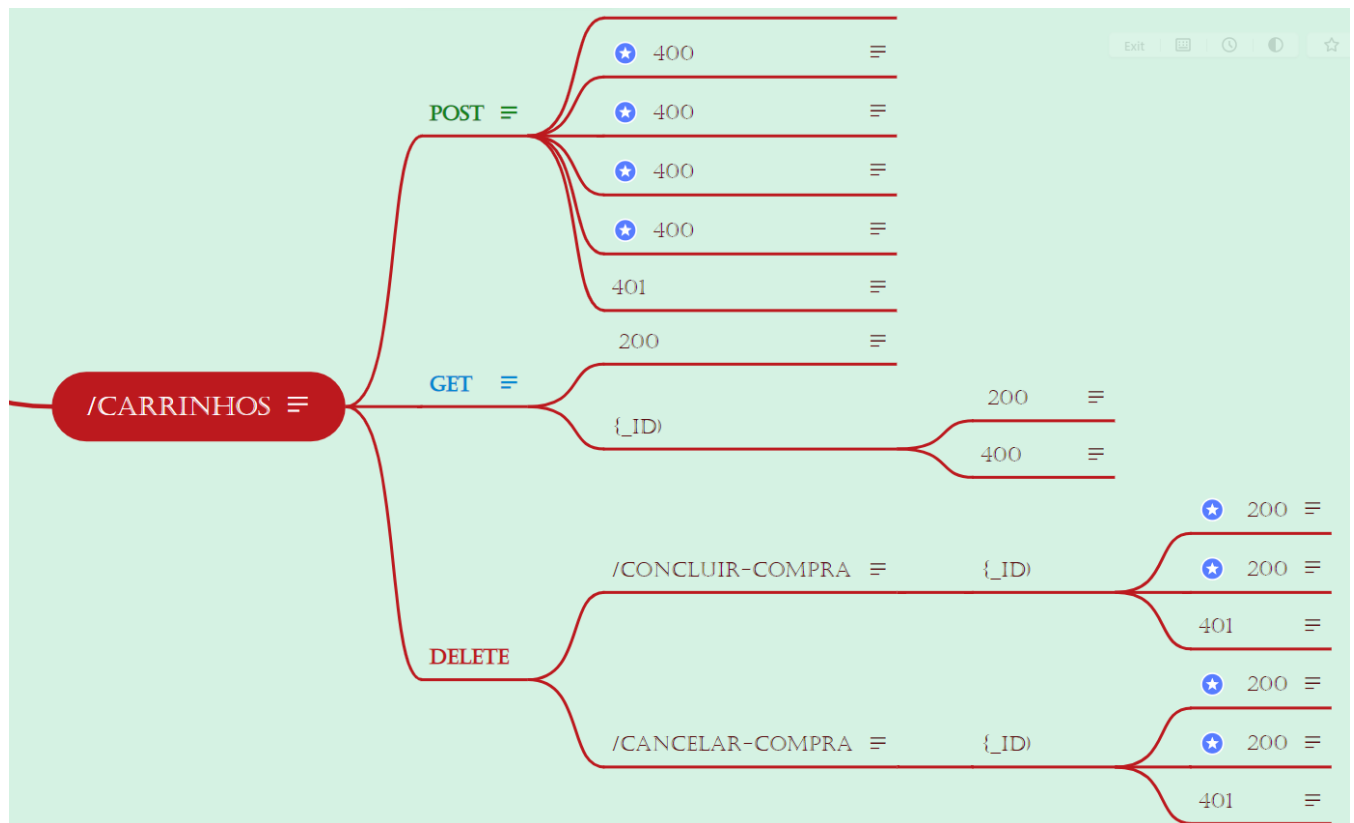
Qualquer problema identificado durante os testes será devidamente registrado como bugs ou melhorias nas issues do GitHub.

## Prioridades

Sendo esta API uma loja fictícia de ecommerce, o fluxo prioritário refere-se ao fluxo de execução de compras para usuários padrão e fluxo de usuários administradores e suas funcionalidades, incluindo cadastro de usuário, realização de login, cadastro de produtos e cadastro de carrinho. Nas prioridades os reports de bugs e melhorias, serão descritos no projeto, levando em conta as respostas das requisições.

## Mapa Mental





## Suite de casos de teste

Segue abaixo os endpoints com os cenários de teste, os critérios de aceitação são baseados na documentação do swagger e nos critérios de aceitação da User Story da Challenge em questão. Os casos de teste que estão apenas com um sublinhado são cenários em que a validação não está presente na documentação mas nas respostas às requisições, elas estão presentes. Os casos de teste que estão em vermelho são validações complementares para atender aos quesitos do challenge, da documentação do swagger e alguma melhoria que achei pertinente adicionar para adicionar posteriormente, em um segundo momento, após conversar com o time de desenvolvimento.

### 1. Login (POST)

#### CT01 - Login Bem-sucedido

**Descrição:** Verificar se é possível realizar um login bem-sucedido com credenciais válidas.

#### **Critérios de Aceite:**

O sistema deve retornar um status code 200 (OK).

O sistema deve retornar uma mensagem "Login realizado com sucesso".

O sistema deve fornecer um token de autorização válido.

O token de autorização retornado deve ter uma duração de 600 segundos (10 minutos).

#### **CT02 - Login com E-mail ou Senha Inválidos**

**Descrição:** Verificar se o sistema trata corretamente o caso em que o e-mail ou a senha fornecidos são inválidos.

##### **Critérios de Aceite:**

O sistema deve retornar um status code 401 (Unauthorized).

O sistema deve retornar uma mensagem "E-mail e/ou senha inválidos".

#### **CT03 - Login com obrigatoriedade nos campos**

**Descrição:** Verificar se o sistema trata corretamente o caso em que o e-mail e/ou a senha estão em branco.

##### **Critérios de Aceite:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem informando que E-mail ou password não podem ficar em branco.

#### **CT04 - Indisponibilidade do Servidor**

**Descrição:** Verificar se o sistema trata corretamente o cenário em que o servidor está temporariamente indisponível durante a tentativa login do usuário.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 503 (Service Unavailable).

O sistema deve retornar uma mensagem indicando que o serviço está temporariamente indisponível ou fora do ar

## **2. *Usuário(POST)***

#### **CT05 - Cadastro de Usuário com Sucesso**

**Descrição:** Verificar se um novo usuário pode ser cadastrado com sucesso.

##### **Critérios de Aceitação:**

O cadastro é bem-sucedido quando todos os campos obrigatórios são fornecidos corretamente.

O sistema deve retornar um status code 201.

O sistema não permite o cadastro de um usuário com um e-mail já existente.

#### **CT06 - Cadastro de Usuário com E-mail Já Existente**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta se cadastrar com um e-mail que já está sendo utilizado por outro usuário.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Este email já está sendo usado".

#### **CT07 - Cadastro de Usuário com Email Fora do Padrão**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta se cadastrar email fora do padrão.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar a mensagem "email deve ser um email válido".

#### **CT08 - Cadastro de Usuário com E-mail do Gmail ou Hotmail**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta se cadastrar com um e-mail do Gmail ou Hotmail.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Não é permitido cadastrar usuários com e-mails de provedor gmail e hotmail".

#### **CT09 - Cadastro de Usuário com Senha Fora do Tamanho Permitido**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta se cadastrar com uma senha com menos de 5 caracteres ou mais de 10 caracteres.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem indicando que a senha deve conter no mínimo 5 caracteres e no máximo 10 caracteres

#### **CT10 - Cadastro de Usuário com Espaço no Campo Nome**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta se cadastrar com espaço no campo nome.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem indicando que campo nome não aceita um espaço como nome

#### **CT11 - Cadastro de Usuário com Espaço no Campo Senha**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta se cadastrar com espaço em branco no campo senha

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem indicando que campo senha não aceita um espaço em branco como senha.

### **CT12 - Cadastro de Usuários Rota Exclusiva para Administradores**

**Descrição:** Verificar se o sistema trata corretamente o caso usuário padrão( administrador = false ) quer acessar rota de cadastro de usuário .

**Critérios de Aceitação:**

O sistema deve retornar um status code 403.

O sistema deve retornar uma mensagem de "Rota exclusiva para administradores".

## **2. Usuário(DELETE)**

### **CT013 - Deletar Usuário por ID com Sucesso**

**Descrição:** Verificar se um usuário pode ser deletado com sucesso.

**Critérios de Aceitação:**

O sistema deve retornar um status code 200 (OK).

A função deletar é bem-sucedida e a mensagem "Registro excluído com sucesso" é exibida para o usuário.

### **CT014 - Deletar Usuário por ID com Sucesso**

**Descrição:** Verificar se mensagem informando que não há registros é exibida com sucesso quando tenta excluir usuário.

**Critérios de Aceitação:**

O sistema deve retornar um status code 200 (OK).

A função deletar é bem-sucedida e a mensagem "Nenhum registro excluído"

### **CT15 - Deletar Usuário com Carrinho Cadastrado**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta excluir um usuário que tenha carrinho cadastrado.

**Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Não é permitido excluir usuário com carrinho cadastrado" e o ID do carrinho.

### **CT16 - Deletar Usuário Rota Exclusiva para Administradores**

**Descrição:** Verificar se o sistema trata corretamente o caso usuário padrão( administrador = false ) quer acessar rota de deletar usuário .

**Critérios de Aceitação:**

O sistema deve retornar um status code 403.

O sistema deve retornar uma mensagem de "Rota exclusiva para administradores".

## **2. Usuário(GET)**

### **CT17 - Listar Usuários Cadastrados com Sucesso**

**Descrição:** Verificar se o sistema trata corretamente quando um usuário acessa a lista de usuários.

#### **Critérios de Aceitação:**

O sistema deve retornar um status code 200.

O sistema deve retornar uma lista com a quantidade de usuários, os parâmetros nome, email, senha e administrador e o ID.

### **CT18 - Buscar Usuário por ID existente**

**Descrição:** Verificar se o sistema retorna usuário requerido corretamente.

#### **Critérios de Aceitação:**

O sistema deve retornar um status code 200.

O sistema deve retornar os dados do usuário encontrado e seu ID correspondente.

### **CT19- Buscar Usuário por ID inexistente -**

**Descrição:** Verificar se o sistema retorna mensagem de usuário não encontrado caso o ID não contenha na base de dados.

#### **Critérios de Aceitação:**

O sistema deve retornar um status code 400.

O sistema deve retornar a mensagem "Usuário não encontrado"

## **2. *Usuário(PUT)***

### **CT20 - Editar Usuário por ID com Sucesso**

**Descrição:** Verificar se usuário pode ser editado e cadastrado com sucesso.

#### **Critérios de Aceitação:**

O cadastro é bem-sucedido quando todos os campos obrigatórios são preenchidos corretamente.

O sistema não permite o cadastro de um usuário com um e-mail já utilizado.

O sistema não permite o cadastro de usuários com e-mails do Gmail ou Hotmail.

Senhas com menos de 5 caracteres ou mais de 10 caracteres não são permitidas.

O sistema deve retornar um status code 200.

O sistema deve retornar uma mensagem de "Registro alterado com sucesso".

### **CT21 - Editar Usuário cujo ID não existe**

**Descrição:** Verificar se o sistema trata corretamente caso o ID editado não exista, cadastrando um novo usuário neste caso.

#### **Critérios de Aceitação:**

O sistema deve retornar um status code 201.



O sistema deve retornar uma mensagem de "Cadastro realizado com sucesso" e um ID.

#### **CT22 - Editar Usuário com E-mail já cadastrado**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta utilizar e-mail já existente.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400.

O sistema deve retornar uma mensagem de "Este email já está sendo usado".

#### **CT23 - Cadastro de Usuário com E-mail do Gmail ou Hotmail**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta cadastrar com um e-mail do Gmail ou Hotmail.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Não é permitido cadastrar usuários com e-mails de provedor gmail e hotmail".

#### **CT24 - Cadastro de Usuário com Senha Fora do Tamanho Permitido**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta cadastrar com uma senha com menos de 5 caracteres ou mais de 10 caracteres.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem indicando que a senha deve conter no mínimo 5 caracteres e no máximo 10 caracteres

#### **CT25 - Cadastro de Usuário com Espaço no Campo Nome**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta cadastrar com espaço no campo nome.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem indicando que campo nome não aceita um espaço como nome

#### **CT26 - Cadastro de Usuário com Email Fora do Padrão**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um usuário tenta cadastrar com email fora do padrão.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem indicando que campo E-mail deve seguir um padrão, conter '@' e um '.' após provedor.

### 3. **Produto (POST)**

#### **CT27 - Cadastro de Produto com Sucesso**

**Descrição:** Verificar se é possível cadastrar um novo produto com informações válidas.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 201 (Created).

O sistema deve retornar uma mensagem de "Cadastro realizado com sucesso".

O produto deve ser registrado corretamente na base de dados.

#### **CT28 - Nome de Produto Duplicado no Cadastro**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um produto com o mesmo nome já existe no cadastro.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Já existe produto com esse nome".

#### **CT29 - Cadastro de Produto quando Token Ausente, Inválido ou Expirado**

**Descrição:** Verificar com o sistema se comporta quando o token de acesso é ausente, inválido ou expirado para cadastrar produtos.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 401.

O sistema deve retornar uma mensagem de "Token de acesso ausente, inválido, expirado ou usuário do token não existe mais".

#### **CT30 - Cadastro de produto não Aceita Preço Decimal**

**Descrição:** Verificar como o sistema se comporta quando, ao cadastrar um produto, é inserido um preço com numero decimal no campo 'preço'.

##### **Critérios de Aceitação:**

O sistema deve retornar um code de 200.

O sistema deve aceitar preços com números decimais.

#### **CT31 - Cadastro de Produtos Rota Exclusiva para Administradores**

**Descrição:** Verificar se o sistema trata corretamente o caso usuário padrão( administrador = false ) quer acessar rota de cadastro de produto .

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 403.

O sistema deve retornar uma mensagem de "Rota exclusiva para administradores".

### 3. **Produto (GET)**

#### **CT32 - Listar Produtos com Sucesso**

**Descrição:** Verificar como sistema se comporta quando usuário solicita listar os produtos.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 200.

O sistema deve retornar a mensagem contendo a quantidade de produtos e a descrição, com nome, preço, descrição, quantidade e o ID, dos mesmos.

#### **CT33 - Listar produtos sem Sucesso**

**Descrição:** Verificar como sistema se comporta quando usuário solicita listar os produtos mas algo inesperado acontece, como servidor temporariamente indisponível.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 503 (Service Unavailable).

O sistema deve retornar uma mensagem indicando que o serviço está temporariamente indisponível ou fora do ar.

#### **CT34 - Busca de Produto por ID**

**Descrição:** Verificar se é possível buscar um produto por ID com sucesso.

##### **Critérios de Aceite:**

O sistema deve retornar um status code 200 (OK).

O sistema deve retornar os detalhes do produto corretamente.

#### **CT35 - Busca de Produto por ID Inexistente**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um produto com o ID especificado não existe.

##### **Critérios de Aceite:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Produto não encontrado".

### **3. *Produto(DELETE)***

#### **CT36 - Exclusão de Produto por ID**

**Descrição:** Verificar se é possível excluir um produto com sucesso.

##### **Critérios de Aceitação:**

O sistema deve retornar um status code 200 (OK).

O sistema deve retornar uma mensagem de "Registro excluído com sucesso".

#### **CT37 - Exclusão de Produto Sem Registros do Produto**

**Descrição:** Verificar se o sistema trata corretamente quando tenta excluir produto sem registro do produto.

**Critérios de Aceitação:**

O sistema deve retornar um status code 200 (OK).

O sistema deve retornar uma mensagem de "Nenhum registro excluído".

**CT38 - Exclusão de Produto que Faz Parte de Carrinho**

**Descrição:** Verificar se o sistema trata corretamente o caso em que um produto que faz parte de um carrinho é excluído.

**Critérios de Aceitação:**

O sistema deve retornar um status code 400 (Bad Request).

O sistema deve retornar uma mensagem de "Não é permitido excluir produto que faz parte de carrinho" e juntamente mostra o ID do carrinho.

**CT39 - Exclusão de Produto quando Token Ausente, Inválido ou Expirado**

**Descrição:** Verificar com o sistema se comporta quando o token de acesso é ausente, inválido ou expirado quando tenta excluir produtos.

**Critérios de Aceitação:**

O sistema deve retornar um status code 401.

O sistema deve retornar uma mensagem de "Token de acesso ausente, inválido, expirado ou usuário do token não existe mais".

**CT40 - Exclusão de Produtos Rota Exclusiva para Administradores**

**Descrição:** Verificar se o sistema trata corretamente o caso usuário padrão( administrador = false ) quer acessar rota para excluir um produto .

**Critérios de Aceitação:**

O sistema deve retornar um status code 403.

O sistema deve retornar uma mensagem de "Rota exclusiva para administradores".

**CT41 - Indisponibilidade do Servidor durante Exclusão de Produto**

**Descrição:** Verificar se o sistema trata corretamente o cenário em que o servidor está temporariamente indisponível durante a tentativa de exclusão de um produto.

**Critérios de Aceitação:**

O sistema deve retornar um status code 503 (Service Unavailable).

O sistema deve retornar uma mensagem indicando que o serviço está temporariamente indisponível ou fora do ar.

## **4. Carrinho(POST)**

**CT47 - Cadastro de Carrinho**

**Descrição:** Verificar se é possível cadastrar um carrinho com sucesso.

**Critérios de Aceite:**

O sistema deve retornar um status code 201 (Created).

O sistema deve retornar uma mensagem de "Cadastro realizado com sucesso".

A quantidade de cada produto inserido no carrinho deve ser reduzida no cadastro de produtos.

#### **4. Carrinho(DELETE)**

**CT56 - Exclusão de Carrinho ao Concluir Compra**

**Descrição:** Verificar se é possível excluir um carrinho ao concluir uma compra com sucesso.

**Critérios de Aceitação:**

O sistema deve retornar um status code 200 (OK).

O carrinho vinculado ao usuário do token deve ser excluído.

O sistema deve retornar uma mensagem de "Registro excluído com sucesso".

## Candidatos para automação

Serão realizados os testes automatizados em todos os métodos da API ServeRest, cumprindo assim, os fluxos desta API de ecommerce fictício. Dentro de cada um dos cenários foi utilizado, através das pré requestes e dos testes, endpoints de diferentes métodos:

- Rotas Login => POST/usuarios - DELETE/usuarios.
- Rotas Usuários => POST/login , POST/produtos-, POST/carrinhos,DELETE/produtos, DELETE/carrinhos

Os bugs e melhorias serão relatados, através das Issues no GitHub, tendo como base sempre a documentação do swagger e os critérios de aceitação da Challenge para cumprir o projeto proposto.