

PLANO DE TESTE DA API REST SERVEREST



Realizado por: Matheus Henrique Chagas

Supervisonado por: Larissa Campos.

Recife, 2022

**Sumário**

[Introdução 5](#_Toc108783695)

[Visão geral 5](#_Toc108783696)

[Objetivo 5](#_Toc108783697)

[Escopo 5](#_Toc108783698)

[Regra de negócio 6](#_Toc108783699)

[Tipos de teste realizados neste 6](#_Toc108783700)

[Teste Funcional: 6](#_Toc108783701)

[Teste de regressão: 6](#_Toc108783702)

[Teste de caixa preta: 6](#_Toc108783703)

[Teste de integração: 6](#_Toc108783704)

[Teste de Performance: 6](#_Toc108783705)

[Estratégias abordadas: 7](#_Toc108783706)

[Cadastro de Produtos: 7](#_Toc108783707)

[Finalização da compra: 7](#_Toc108783708)

[Cancelar da compra: 7](#_Toc108783709)

[Mapa Mental 8](#_Toc108783710)

[Casos de Teste 8](#_Toc108783711)

[Ferramentas utilizadas 18](#_Toc108783712)

[Cobertura de Teste 19](#_Toc108783713)

[Operator Coverage 20](#_Toc108783714)

[**Parameter Coverage** 20](#_Toc108783715)

[**Cobertura média** 20](#_Toc108783716)

Introdução

Visão geral

O ServeRest é uma API REST gratuita que simula uma loja virtual com intuito de servir de material de estudos de testes de API. Nela conseguimos utilizar os principais verbos (POST,GET,DELETE E PUT) de diferentes modos.

Objetivo

Testar as funcionalidades da API ServeRest, com um o fluxo de requisições necessários para realizar a confirmação das informações presentes na documentação baseando-se no comportamento da API durante os testes. Testar as rotas apresentadas construindo um fluxo de testes de maneira sequencial simulando caminhos lógicos dos seus usuários com a finalidade de confirmar seu pleno funcionamento.

Escopo

Neste plano teste iremos validar os seguintes  end points informados na no swagger da api ServeRest (v2.25.4):

•              /login

•              /usuarios

•              /usuarios/{\_id}

•              /produtos

•              /produtos/{\_id}

•              /carrinhos

•              /carrinhos/{\_id}

•              /carrinhos/concluir-compra

•              /carrinhos/cancelar-compra

Regra de negócio

A API apresentada para teste é um e-commerce onde possibilita os usuários realizarem compras selecionando produtos disponíveis e suas quantidades. Para então finalizarem a compra ou cancelarem.

Tipos de teste realizados neste

Teste Funcional:

Este teste tem como meta principal verificar se a aplicação está funcionando conforme a finalidade a qual foi desenvolvida para fazer.

Teste de regressão:

Caso ocorra uma atualização na api ou sistema atualizado, este teste irá certificar que as atualizações não geraram algum bug, ou mal funcionamento no sistema.

Teste de caixa preta:

Todos os testes realizados não são realizados com o código aberto para a equipe de teste, por este motivo se caracteriza este tipo de teste.

Teste de integração:

Esse teste é uma atividade sistemática que tem como objetivo verificar a construção da estrutura do software que está sendo desenvolvida e a sua comunicação entre módulos.

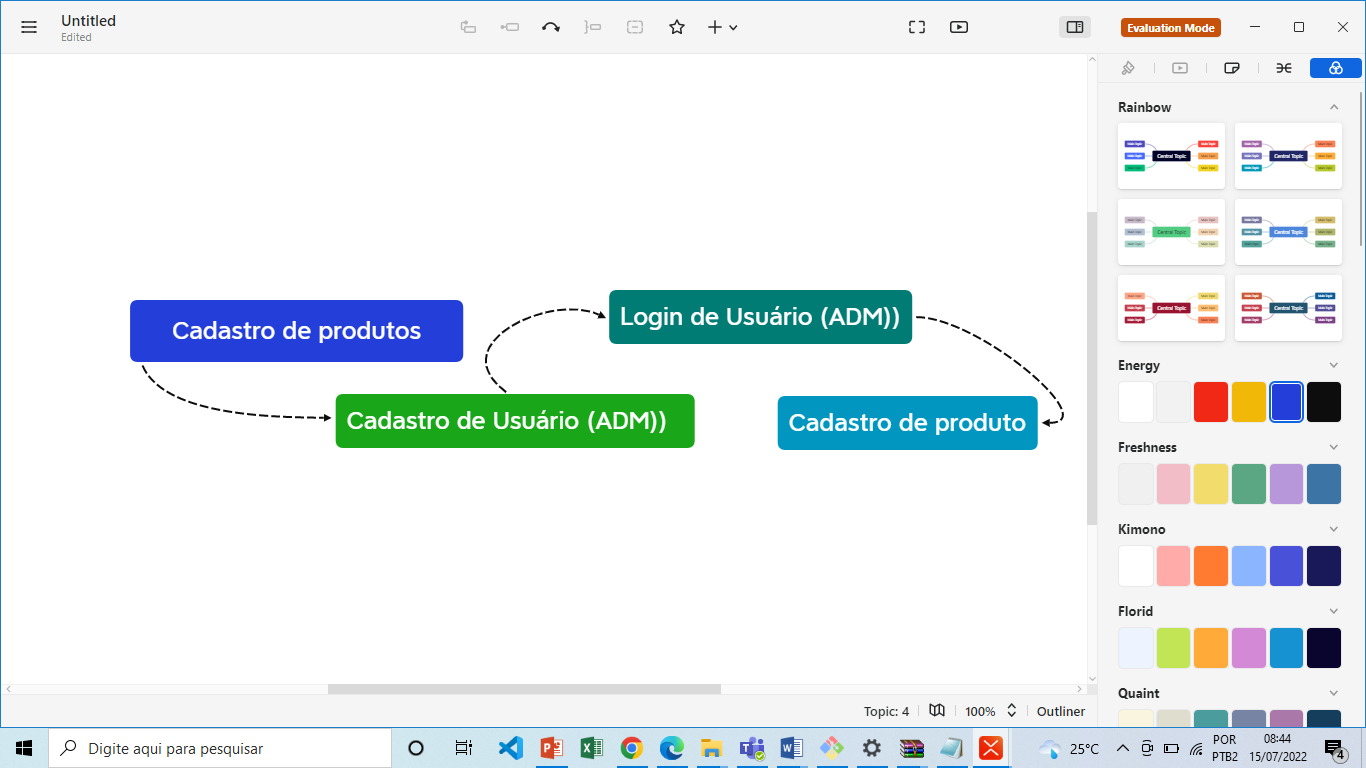
Teste de Performance:

Este teste valida que o sistema fornece tempo de resposta dentro do padrão aceitável em termos de resposta de código a nível local.

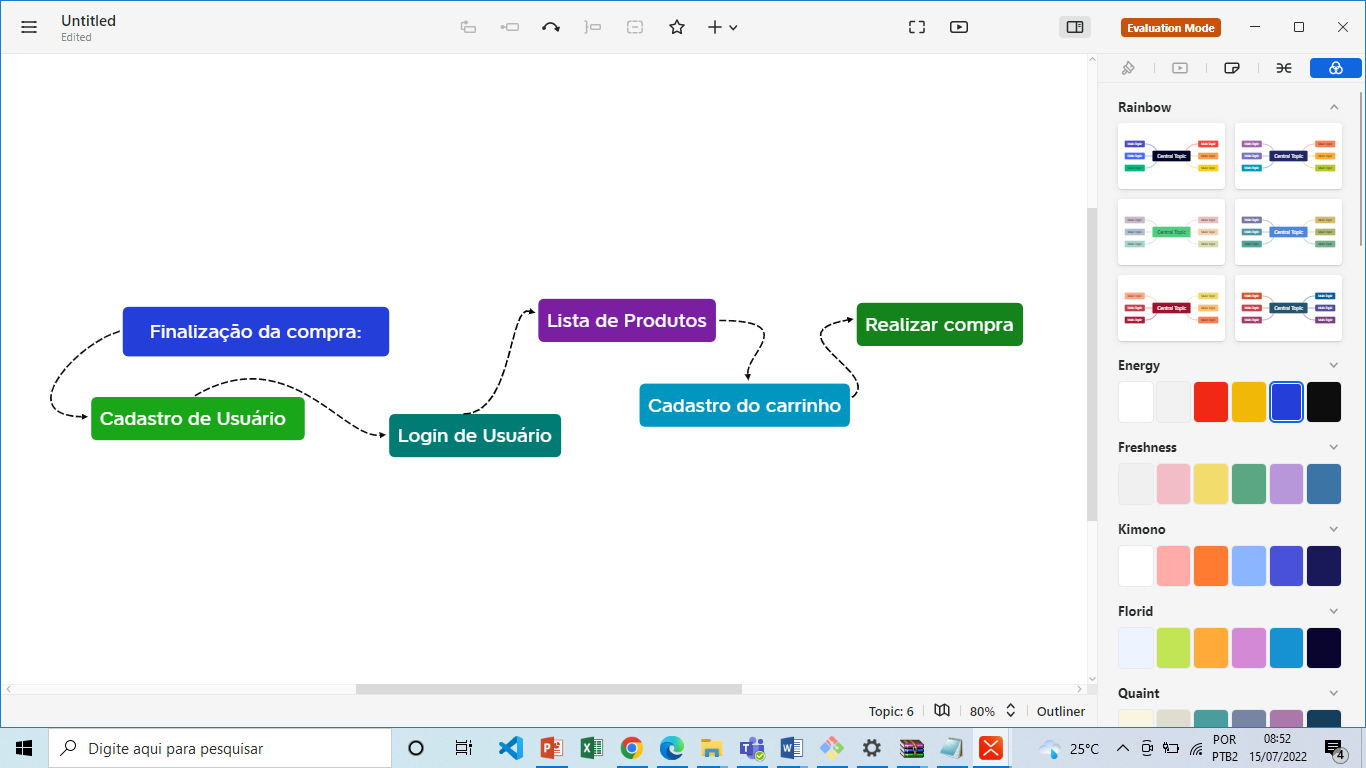
Estratégias abordadas:

Seguindo a regra de negócio, percebemos que existem fluxos de funcionamento os quais devem ser priorizados, inserido a metodologia ágil para os fluxos citados estão ilustrados nas imagens abaixo.

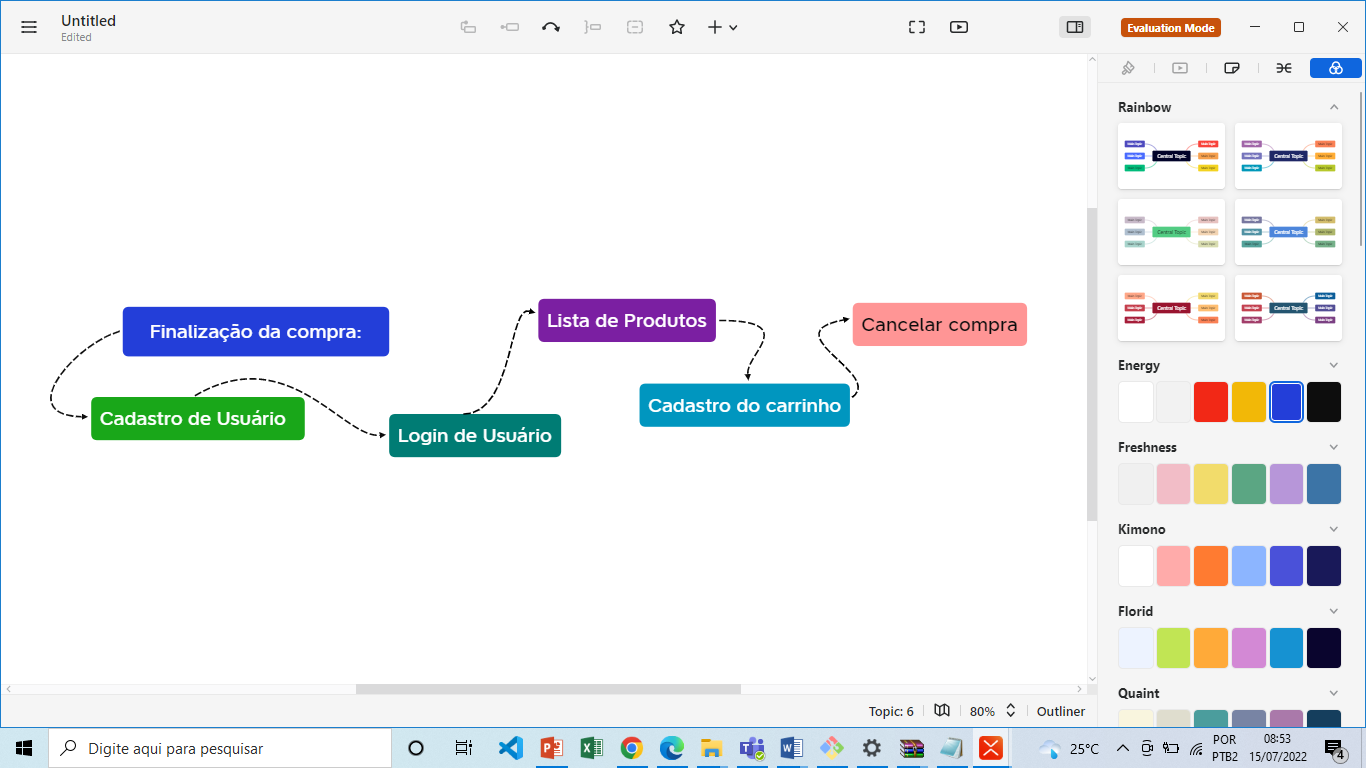
Cadastro de Produtos:

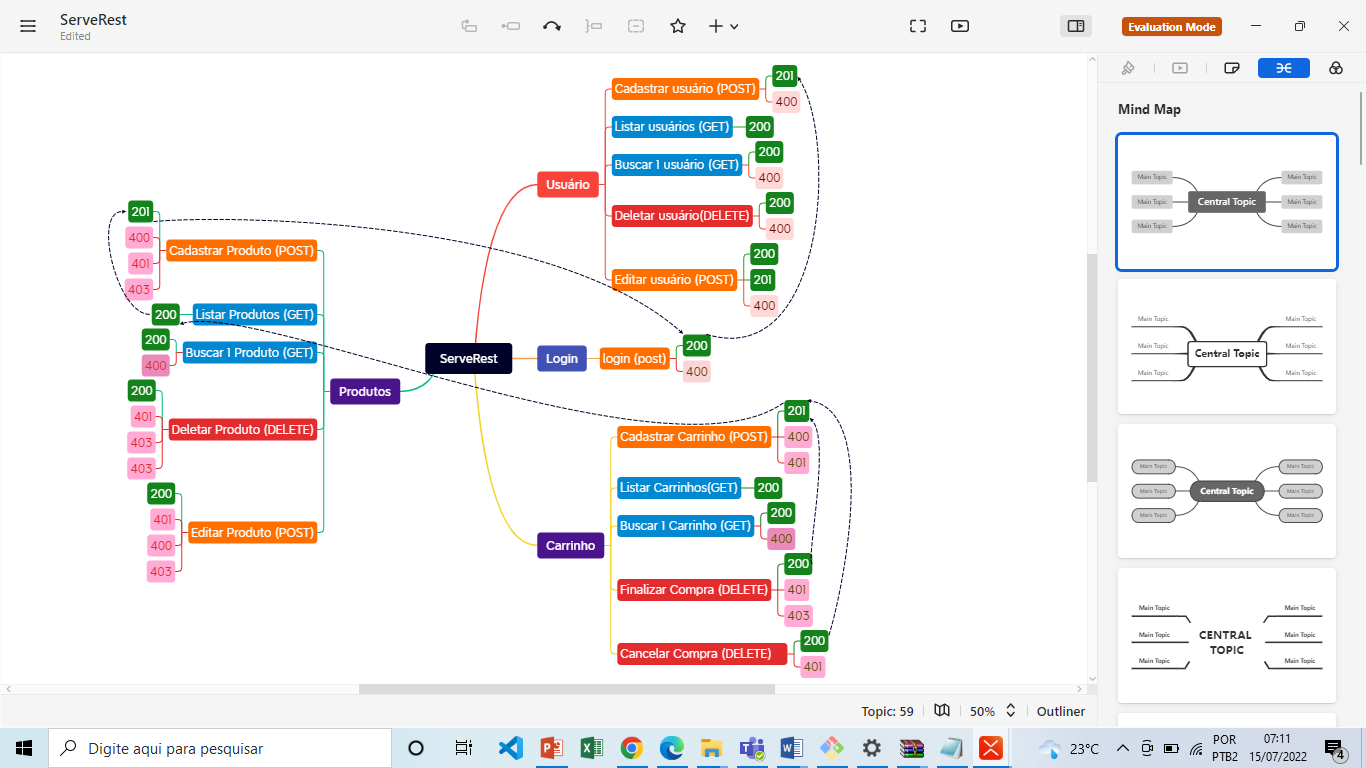


Finalização da compra:



Cancelar da compra:



Mapa Mental

Casos de Teste

**Login**

**End Point: http://localhost:3000/login**

|  |  |  |
| --- | --- | --- |
| **CT-0001: Realizar Login com e-mail e senha corretos:** | | **Verbo: POST** |
| **Pré-condição de execução:**  E-mail e senha precisam estar cadastrados na API ServeRest. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir E-mail de usuário válido** | |
| **2°** | **Inserir senha de usuário válido** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: ”Login realizado com sucesso”. | | |
| **Pós-condição de execução:**  Mostrar mensagem e gerar um Token de autorização de usuário. | | | |
| **CT-0002: Verificar Realizar Login com senha em branco:** | | **Verbo: POST** |
| **Pré-condição de execução: não possui.** | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir E-mail de usuário válido** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: ” Password não pode ficar em branco”. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0003: Verificar Realizar Login com e-mail em branco:** | | **Verbo: POST** |
| **Pré-condição de execução: não possui.** | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir senha de usuário válido** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: ” email não pode ficar em branco”. | | |
| **Pós-condição de execução:** | | | |
| **CT-0004: Verificar Realizar Login com e-mail inválido:** | | **Verbo: POST** |
| **Pré-condição de execução: não possui.** | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir Email sem cadastro ou email, sem @(plataforma de email) e .com** | |
| **2°** | **Inserir senha de usuário** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: ” email deve ser um email válido”. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0005: Verificar Realizar Login com senha Inválida:** | | **Verbo: POST** |
| **Pré-condição de execução: não possui.** | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir E-mail de usuário válido** | |
| **2°** | **Inserir senha de usuário inválida ou incorreta** | |
| **Resultados esperado:** | | | |
| **Status Code: 401** | Mensagem: ” Email e/ou senha inválidos”. | | |
| **Pós-condição de execução: não possui.** | | | |

**Usuários**

**End Point: http://localhost:3000/usuarios**

|  |  |  |
| --- | --- | --- |
| **CT-0006: Verificar Listagem de usuário:** | | **Verbo: GET** |
| **Pré-condição de execução: Possuir usuários cadastrados** | | |
| **Passos** | **Descrição** | |
| **1°** | **Solicitar a listagem de usuários cadastrados** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Lista de usuários com as informações: Quantidade total de usuários, informações separadas de usuário (nome , e-mail, passoword, administrador, id. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0007: Cadastrar Usuário Administrador** | | **Verbo: POST** |
| **Pré-condição de execução:** Inserir e-mail não cadastrados, “Nome” não pode estar em branco, digitar um e-mail válido com “@” e “.com”, senha não pode está em branco. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: Nome, E-mail ,password, administrador:”true”.** | |
| **Resultados esperado:** | | | |
| **Status Code: 201** | "mensagem": "Cadastro realizado com sucesso" | | |
| **Pós-condição de execução:**  **Id de cliente formado.** | | | |
| **CT-0008: Cadastrar Usuário Comum** | | **Verbo: POST** |
| **Pré-condição de execução:** Inserir e-mail não cadastrados, “Nome” não pode estar em branco, digitar um e-mail válido com “@” e “.com”, senha não pode está em branco. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: Nome, E-mail ,password, administrador:”false”.** | |
| **Resultados esperado:** | | | |
| **Status Code: 201** | "mensagem": "Cadastro realizado com sucesso" | | |
| **Pós-condição de execução:**  **Id de usuário formado.** | | | |
| **CT-0009: Cadastrar Usuário sem nome.** | | **Verbo: POST** |
| **Pré-condição de execução:** Inserir e-mail não cadastrados, digitar um e-mail válido com “@” e “.com”, senha não pode estar em branco. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: deixar nome sem caracateres, E-mail ,password, administrador: “false”.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "nome": "nome não pode ficar em branco" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0010: Cadastrar Usuário sem senha.** | | **Verbo: POST** |
| **Pré-condição de execução: Inserir e-mail não cadastrados, “Nome” não pode estar em branco, digitar um e-mail válido com “@” e “.com”.** | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: nome, E-mail , deixar senha sem caracteres, administrador: “false” ou “true”**  **.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "password": "password não pode ficar em branco" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0011: Cadastrar Usuário sem e-mail.** | | **Verbo: POST** |
| **Pré-condição de execução: I**nserir e-mail não cadastrados, “Nome” não pode estar em branco, senha não pode estar em branco. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: nome, E-mail sem caracteres, inserir uma senha, administrador: “false” ou “true”.**  **.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "email": "email não pode ficar em branco" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0012: Cadastrar Usuário com e-mail inválido.** | | **Verbo: POST** |
| **Pré-condição de execução:** Inserir e-mail não cadastrados, “Nome” não pode estar em branco, “Senha” não pode estar em branco. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: nome, E-mail sem “@” ou “.com”, inserir uma senha, administrador: “false” ou “true”.**  **.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "email": "email deve ser um email válido" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0013: Cadastrar Usuário com administrador inválido.** | | **Verbo: POST** |
| **Pré-condição de execução:** Inserir e-mail não cadastrados, “Nome” não pode estar em branco, “Senha” não pode estar em branco. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir dados cadastrais: nome, E-mail sem “@” ou “.com”, inserir uma senha, administrador: vazio ou com outras palavras que não sejam false ou true.**  **.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "administrador": "administrador deve ser 'true' ou 'false'" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0014: Deletar usuário.** | | **Verbo: DELETE** |
| **Pré-condição de execução:** O usuário precisa estar cadastrado, obter Id do usuário | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir o /ID do usuário a ser deletado após o URL da API.**  **.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: "message": "Registro excluído com sucesso" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0015: Deletar usuário com carrinho.** | | **Verbo: DELETE** |
| **Pré-condição de execução:** O usuário precisa estar cadastrado. Obter Id do usuário, realizar um carrinho para o usuário. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir o /ID do usuário a ser deletado após o URL da API.**  **.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Não é permitido excluir usuário com carrinho cadastrado",  "idCarrinho": "ID do carrinho do usuário" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0016: Editar usuário.** | | **Verbo: PUT** |
| **Pré-condição de execução:** O usuário precisa estar cadastrado. Obter Id do usuário. Caso a alteração seja o e-mail, o e-mail não pode estar cadastrado. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir novas informações:**  **{**  **"nome": "informar",**  **"email": "informar",**  **"password": "informar",**  **"administrador": "informar"**    **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: "message": "Registro alterado com sucesso" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0017: Editar usuário sem informações.** | | **Verbo: PUT** |
| **Pré-condição de execução:** O usuário precisa estar cadastrado. Obter Id do usuário. Caso a alteração seja o e-mail, o e-mail não pode estar cadastrado. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir novas informações:**  **{**  **"nome": "vazio ",**  **"email": " vazio ",**  **"password": " vazio ",**  **"administrador": "vazio"**    **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "nome": "nome é obrigatório",  "email": "email é obrigatório",  "password": "password é obrigatório",  "administrador": "administrador é obrigatório" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0018: Editar usuário com e-mail existente.** | | **Verbo: PUT** |
| **Pré-condição de execução:** O usuário precisa estar cadastrado. Obter Id do usuário. Caso a alteração seja o e-mail, o e-mail não pode estar cadastrado. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir novas informações:**  **{**  **"nome": "informar",**  **"email": "informar e-mail existente",**  **"password": "informar",**  **"administrador": "informar"**    **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Este email já está sendo usado" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0019: Editar usuário com e-mail existente no cadastro.** | | **Verbo: PUT** |
| **Pré-condição de execução:** O usuário precisa estar cadastrado. Obter Id do usuário. Caso a alteração seja o e-mail, o e-mail não pode estar cadastrado. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir novas informações:**  **{**  **"nome": "informar",**  **"email": "informar e-mail existente",**  **"password": "informar",**  **"administrador": "informar"**    **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Este email já está sendo usado" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0020: Editar usuário com com ID não existente.** | | **Verbo: PUT** |
| **Pré-condição de execução:** O usuário precisa ser um novo cadastrado. Obter Id do usuário. Caso a alteração seja o e-mail, o e-mail não pode estar cadastrado. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir novas informações:**  **{**  **"nome": "informar",**  **"email": "informar ",**  **"password": "informar",**  **"administrador": "informar"**    **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 201** | Um novo cadastro será realizado  Mensagem: ""message": "Cadastro realizado com sucesso" | | |

**Produtos**

**End Point: http://localhost:3000/produtos**

|  |  |  |
| --- | --- | --- |
| **CT-0021: Lista de produtos.** | | **Verbo: GET** |
| **Pré-condição de execução:** não possui. | | |
| **Passos** | **Descrição** | |
| **1°** | **Usar URL e end point já informado.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Informar todos os produtos cadastrados no sistema. E suas informações básicas como: nome, preço, descrição, quantidade, id. | | |
| **Pós-condição de execução:** | | | |
| **CT-0022: Buscar de produto** | | **Verbo: GET** |
| **Pré-condição de execução:** Cadastrar produto, obter o Id do produto | | |
| **Passos** | **Descrição** | |
| **1°** | **Usar URL e end point já informado/ID do produto.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Informações básicas do produto desejado como: nome, preço, descrição, quantidade, id. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0023: Buscar de produto sem id.** | | **Verbo: GET** |
| **Pré-condição de execução:** Cadastrar produto, obter o Id do produto | | |
| **Passos** | **Descrição** | |
| **1°** | **Usar URL e end point já informado/ID do produto.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Produto não encontrado" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0024: Cadastrar Produto.** | | **Verbo: POST** |
| **Pré-condição de execução:** token de usuário ADM. Preços não pode ser menor ou igual a zero. Quantidade não pode ser menor que a zero. | | |
| **Passos** | **Descrição** | |
| **1°** | **Informar:**  **{**  **"nome":" informar nome",**  **"preco": informar preço,**  **"descricao": “informar descrição”**  **"quantidade": informar quantidade**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Produto não encontrado" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0025: Cadastrar produto com nome já cadastrado.** | | **Verbo: POST** |
| **Pré-condição de execução:** token de usuário ADM. Preços não pode ser menor ou igual a zero. Quantidade não pode ser menor que a zero. | | |
| **Passos** | **Descrição** | |
| **1°** | **Informar:**  **{**  **"nome":" informar nome já cadastrado",**  **"preco": informar preço,**  **"descricao": “informar descrição”**  **"quantidade": informar quantidade**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Já existe produto com esse nome" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0026: Cadastrar produto com usuário de token ausente ou inválido.** | | **Verbo: POST** |
| **Pré-condição de execução:** token de usuário comun. Preços não pode ser menor ou igual a zero. Quantidade não pode ser menor que a zero. | | |
| **Passos** | **Descrição** | |
| **1°** | **Informar:**  **{**  **"nome":" informar nome",**  **"preco": informar preço,**  **"descricao": “informar descrição”**  **"quantidade": informar quantidade**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 401** | Mensagem: "message": "Já existe produto com esse nome" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0027: Cadastrar produto com usuário Comum.** | | **Verbo: POST** |
| **Pré-condição de execução:** token de usuário comum. Preços não pode ser menor ou igual a zero. Quantidade não pode ser menor que a zero. | | |
| **Passos** | **Descrição** | |
| **1°** | **Informar:**  **{**  **"nome":" informar nome",**  **"preco": informar preço,**  **"descricao": “informar descrição”**  **"quantidade": informar quantidade**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 401** | Mensagem: "message": "Rota exclusiva para administradores" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0028: Deletar um produto com Usuário ADM.** | | **Verbo: DELETE** |
| **Pré-condição de execução:** Obter ID do produto. Utilizar um token de um usuário administrador. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir /id do produto no end poit** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: "message": "Registro excluído com sucesso" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0029: Deletar um produto com usuário Comum.** | | **Verbo: DELETE** |
| **Pré-condição de execução:** Obter ID do produto. Utilizar um token de um usuário Comum. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir /id do produto no end point.** | |
| **Resultados esperado:** | | | |
| **Status Code: 403** | Mensagem: "message": "Rota exclusiva para administradores" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0030: Deletar um produto com carrinho.** | | **Verbo: DELETE** |
| **Pré-condição de execução:** Obter ID do produto no carrinho . Utilizar um token de um usuário ADM. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir /id do produto no end point.** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Não é permitido excluir produto que faz parte de carrinho", e irá informar os Id dos carrinhos com o produto selecionado. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0031: Editar um produto com Usuário ADM.** | | **Verbo: PUT** |
| **Pré-condição de execução:** Obter ID do produto. Utilizar um token de um usuário ADM. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir /id do produto no end point.** | |
| **2°** | **Adicionar alterações no body:**  **{**  **"nome": "Nome desejado",**  **"preco": preço desejado,**  **"descricao": "descrição desejada",**  **"quantidade": quantidade desejada**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem "message": "Registro alterado com sucesso" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0032: Editar um produto com Usuário Comum.** | | **Verbo: PUT** |
| **Pré-condição de execução:** Obter ID do produto no carrinho . Utilizar um token de um usuário ADM. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir /id do produto no end point.** | |
| **2°** | **Adicionar alterações no body:**  **{**  **"nome": "Nome desejado",**  **"preco": preço desejado,**  **"descricao": "descrição desejada",**  **"quantidade": quantidade desejada**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 403** | Mensagem: "message": "Rota exclusiva para administradores" | | |
| **Pós-condição de execução: não possui.** | | | |

**Carrinho**

**End Point: http://localhost:3000/carrinhos**

|  |  |  |
| --- | --- | --- |
| **CT-0033: Listar carrinhos de usuários** | | **Verbo: GET** |
| **Pré-condição de execução:** ter carrinhos cadastrados | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir: end point.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: informar a quantidade total de carrinhos e produtos selecionados, assim como Id Produtos, suas quantidades, preços unitários, preço totais, quantidade totais, id do usuário e id do carrinho. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0034: Cadastrar carrinho** | | **Verbo: POST** |
| **Pré-condição de execução:** ter um id de usuário cadastrado, solicitar produtos existem com quantidades menores ou iguais a quantidade em estoque. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir end point.** | |
| **2°** | **Inserir informações de produtos que deseja adicionar no carrinho.**  **{**  **"produtos": [**  **{**  **"idProduto": "do produto desejado",**  **"quantidade": desejada**  **}**  **]**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 201** | Mensagem:"message": "Cadastro realizado com sucesso", | | |
| **Pós-condição de execução: Gerar id do carrinho.** | | | |
| **CT-0035: Cadastrar mais de um carrinho por usuário.** | | **Verbo: POST** |
| **Pré-condição de execução:** ter um id de usuário cadastrado, solicitar produtos existem com quantidades menores ou iguais a quantidade em estoque. Após realizar o cadastro do carrinho repetir os mesmos passos. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir end point.** | |
| **2°** | **Inserir informações de produtos que deseja adicionar no carrinho.**  **{**  **"produtos": [**  **{**  **"idProduto": "do produto desejado",**  **"quantidade": desejada**  **}**  **]**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Não é permitido ter mais de 1 carrinho" | | |
| **Pós-condição de execução: não possui..** | | | |
| **CT-0036: Cadastrar Carrinho e solicitar produto com quantidade maior que em estoque.** | | **Verbo: POST** |
| **Pré-condição de execução:** ter um id de usuário cadastrado e com carrinho, solicitar produtos existem com quantidades maiores em estoque. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir end point.** | |
| **2°** | **Inserir informações de produtos que deseja adicionar no carrinho.**  **{**  **"produtos": [**  **{**  **"idProduto": "do produto desejado",**  **"quantidade": maior que em estoque**  **}**  **]**  **}** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Produto não possui quantidade suficiente",  Informar a quantidade solicitada e quantidade disponível em estoque. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0037: Buscar carrinho por ID** | | **Verbo: GET** |
| **Pré-condição de execução:** informar o id do carrinho. | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir end point.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: informações do carrinho como: id do produto, quantidade solicitada, valor unitário, preço total do carrinho, id do usuário, id do carrinho. | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0038: Buscar carrinho por ID não encontrado** | | **Verbo: GET** |
| **Pré-condição de execução:** informar o id do carrinho incorreto ou inválido | | |
| **Passos** | **Descrição** | |
| **1°** | **Inserir end point/ID inválido** | |
| **Resultados esperado:** | | | |
| **Status Code: 400** | Mensagem: "message": "Carrinho não encontrado" | | |
| **Pós-condição de execução: não possui.** | | | |
| **CT-0039: Finalizar carrinho** | | **Verbo: DELETE** |
| **Pré-condição de execução:** informar o token do usuário. | | |
| **Passos** | **Descrição** | |
| **1°** | **Fazer login do usuário.** | |
| **2°** | **Inserir token do usuário em authorization.** | |
| **3°** | **Cadastrar carrinho.** | |
| **4°** | **Inserir end point/carrinhos/cancelar-compra e adicionar o token do usuário.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: "message": "Registro excluído com sucesso" | | |
| **Pós-condição de execução: o carrinho fica vazio, os produtos são retirados do estoque, porém, o usuário permanece com o carrinho ativo.** | | | |
| **CT-0040: Cancelar carrinho** | | **Verbo: DELETE** |
| **Pré-condição de execução:** informar o token do usuário. | | |
| **Passos** | **Descrição** | |
| **1°** | **Fazer login do usuário.** | |
| **2°** | **Inserir token do usuário em authorization.** | |
| **3°** | **Cadastrar carrinho.** | |
| **4°** | **Inserir end point/carrinhos/cancelar-compra e adicionar o token do usuário.** | |
| **Resultados esperado:** | | | |
| **Status Code: 200** | Mensagem: "message": "Registro excluído com sucesso. Estoque dos produtos reabastecido" | | |
| **Pós-condição de execução: o carrinho fica vazio, os produtos são retornam para o estoque, porém, o usuário permanece com o carrinho ativo.** | | | |

Ferramentas utilizadas

Hardware

Um sistema controlado, em separado, será necessário para a fase inicial de teste, montado como um ambiente-padrão do negócio. Para manter a integridade do ambiente de teste, sua rede não deve ser acessível a ninguém de fora do projeto. As impressoras também devem ser exclusivamente para uso da rede de teste.

Software

Testar Postman

Ferramenta utilizada para testar comunicações com APIs, ele possui ambiente para documentação, testes e todo o tipo de requisições em geral.

ClickUP

Ferramenta de produtividade para gerência das etapas do projeto.

Seu uso ajudara a ter uma produção rápida e crescente onde serão gerências as demandas de cada etapa do processo de desenvolvimento do software, assim como a arquitetura de organização das equipes.

Pacote office

Ferramenta usada para construção da documentação necessária para a entrega do software.

Microsoft Teams

Ferramenta que será usada para comunicação constante entre as equipes e terá o caráter formal. Uma parte das documentações oficiais serão transmitidas através dele.

XMind

Ferramenta que será usada para construção do mapa mental a fim de estrutura uma melhorar visualização das atividades de teste.

Cobertura de Teste

As ações do teste proposto teremos os seguintes percentuais de cobertura de teste:

**Path Coverage**

**A aplicação possui 6 end points distintos e estaremos testando os 6 citados.**

**6/6 = 1 = 100%**

## ****Operator Coverage****

**A aplicação possui 16 métodos distintos e estaremos testando os 16 citados.**

**16/16 = 1 = 100%**

**Parameter Coverage**

**A aplicação possui 38 parâmetros distintos e estaremos testando os 35 citados.**

**38/35 = 0,92 = 92%**

**Cobertura média**

100% + 100% + 92% = 97,3