



PLANO DE TESTE SERVEREST V2.26.2

SET 22 - DIEGO HERMYS DE SOUZA VIEIRA

COMPASS UOL

INTRODUÇÃO

O presente plano de teste descreve o planejamento para criar e implementar as validações de cobertura para a API ServeRest em sua versão 2.26.2 (<https://serverest.dev>), disponibilizando para todos os stakeholders do projeto (gerência, time de negócios, desenvolvedores, clientes, equipe de testes, etc) a abordagem de testes adotada que dará cobertura a todas as rotas da API com testes de integração e automação.

SUMÁRIO

- Objetivos.....04
- Escopo.....05
- Mapa Mental.....06
- Suíte de Casos de Teste07 - 12
- Estratégia de Testes.....13
- Testes Candidatos para a Automação....14 - 16
- Ferramentas.....17
- Equipe de Testes.....18
- Cronograma.....19
- Conclusão.....20

OBJETIVOS

Este documento descreve o plano de teste para a API ServeRest V2.26.2 apresentando os objetivos a seguir:

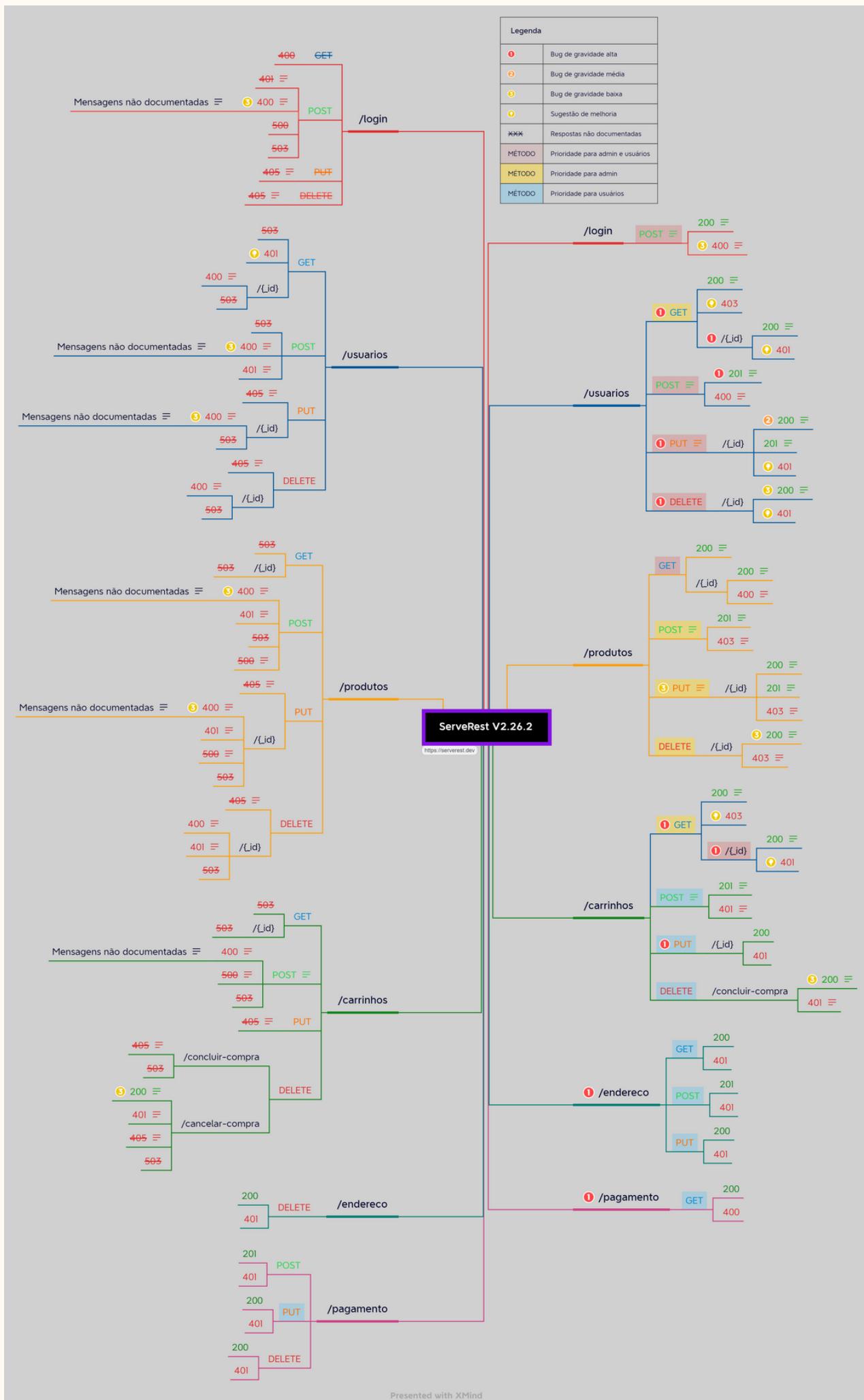
- Identificar e analisar todas as informações da API e sua documentação e mapear suas rotas e funcionalidades;
- Listar as suítes de caso de testes;
- Apresentar as estratégias de testes adotadas;
- Recomendar os testes candidatos para a automação;
- Estimar recursos necessários;
- Apresentar bugs e propor correções e/ou melhorias.

ESCOPO

Este Plano de Teste escreve os testes de integração da ServeRest que serão realizados em ambiente de produção. Assume-se que o teste de unidade já forneceu um cobertura extensiva do código fonte e o teste de todas as interfaces do módulo. Os itens a seguir estão no escopo dos testes:

- Área administrativa (login, administração de usuários, movimentações de produtos e consulta de carrinhos e formas de pagamento);
- Cadastro (criação de novos usuários e administradores);
- Login (atividades de autenticação na rota de login e alteração de senha em usuários);
- Itens fora do fluxo de compra (busca e exclusão em usuários, inclusão, alteração e exclusão em produtos, listar carrinhos);
- Fluxo de compra (login, cadastro e alteração em usuários, busca de produtos, criação e alteração de carrinhos e conclusão de compra através da exclusão de carrinho, informações de entrega e pagamento).





SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas da API ServeRest:

Login

- CT01 - Validar rota de login com sucesso;
- CT02 - Validar rota de login email não cadastrado;
- CT03 - Validar rota de login com senha incorreta;
- CT04 - Validar rota de login com email fora de padrão;
- CT05 - Validar rota de login com senha fraca;
- CT06 - Validar rota de login com campo email vazio;
- CT07 - Validar rota de login sem enviar parâmetro email;
- CT08 - Validar rota de login com campo senha vazio;
- CT09 - Validar rota de login sem enviar o parâmetro senha;
- CT10 - Validar rota de login com campo senha e email vazios;
- CT11 - Validar rota de login sem enviar parâmetros de email e senha;
- CT12 - Validar rota de login enviando parâmetros não aceitos;
- CT13 - Validar rota de login enviando informação sem aspas;
- CT14 - Validar rota de login em rota inválida;
- CT15 - Validar rota de login com método inválido (GET);
- CT16 - Validar rota de login com método inválido (PUT);
- CT17 - Validar rota de login com método inválido (DEL);

Usuários

- CT18 - Validar rota de lista de usuários;
- CT19 - Validar rota de lista de usuários sem autorização;
- CT20 - Validar rota de busca de usuário por id com sucesso;
- CT21 - Validar rota de busca de usuário por id sem sucesso;
- CT22 - Validar rota de busca de usuário por id sem sucesso com token inválido;
- CT23 - Validar rota de criação de usuário com sucesso;
- CT24 - Validar rota de criação de usuário com senha fraca com sucesso;
- CT25 - Validar rota de criação de usuário com campos vazios;



SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas da API ServeRest:

- CT26 - Validar rota de criação de usuário com email fora do padrão;
- CT27 - Validar rota de criação de usuário sem enviar parâmetros;
- CT28 - Validar rota de criação de usuário enviando informação sem aspas;
- CT29 - Validar rota de criação de usuário com email já cadastrado;
- CT30 - Validar rota de alteração de usuário com sucesso;
- CT31 - Validar rota de criação de usuário ao tentar alterar usuário inexistente;
- CT32 - Validar rota de alteração de usuário sem sucesso com token inválido;
- CT33 - Validar rota de alteração de usuário sem sucesso;
- CT34 - Validar rota de alteração de usuário com campos vazios;
- CT35 - Validar rota de alteração de usuário com email fora do padrão;
- CT36 - Validar rota de alteração de usuário sem enviar parâmetros;
- CT37 - Validar rota de alteração de usuário enviando informação sem aspas;
- CT38 - Validar rota de exclusão de usuário com sucesso;
- CT39 - Validar rota de exclusão de usuário sem sucesso com token inválido;
- CT40 - Validar rota de exclusão de usuário inexistente;
- CT41 - Validar rota de exclusão de usuário com carrinho;

Produtos

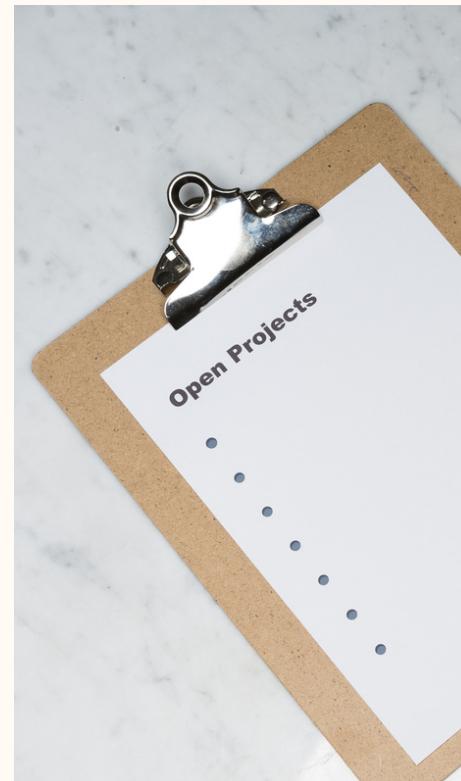
- CT42 - Validar rota de lista de produtos com sucesso;
- CT43 - Validar rota de busca de produtos por id com sucesso;
- CT44 - Validar rota de busca de produtos por id sem sucesso;



SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas da API ServeRest:

- CT45 - Validar rota de cadastro de produtos com sucesso;
- CT46 - Validar rota de cadastro de produto já existente;
- CT47 - Validar rota de cadastro de produto sem sucesso com token inválido;
- CT48 - Validar rota de cadastro de produto com token ausente;
- CT49 - Validar rota de cadastro de produto com token expirado;
- CT50 - Validar rota de cadastro de produtos com campos vazios;
- CT51 - Validar rota de cadastro de produtos sem enviar parâmetros;
- CT52 - Validar rota de cadastro de produtos enviando informação sem aspas;
- CT53 - Valida rota de alteração de produto com sucesso;
- CT54 - Valida rota de criação de produto ao tentar alterar produto com id inexistente;
- CT55 - Valida rota de alteração de produto com nome existente;
- CT56 - Validar rota de alteração de produto sem sucesso com token inválido;
- CT57 - Valida rota de alteração de produto com token ausente;
- CT58 - Valida rota de alteração de produto com token expirado;
- CT59 - Validar rota de alteração de produtos com campos vazios;
- CT60 - Validar rota de alteração de produtos sem enviar parâmetros;
- CT61 - Validar rota de alteração de produtos enviando informação sem aspas;
- CT62 - Validar rota de exclusão de produtos com sucesso;
- CT63 - Validar rota de exclusão de produto inexistente;
- CT64 - Validar rota de exclusão de produtos com produto no carrinho;
- CT65 - Validar rota de exclusão de produto sem sucesso com token inválido;



SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas da API ServeRest:

- CT66 - Validar rota de alteração de produto com token ausente;
- CT67 - Validar rota de alteração de produto com token expirado;

Carrinhos

- CT68 - Validar rota de lista de carrinhos;
- CT69 - Validar rota de lista de carrinhos sem sucesso com token inválido;
- CT70 - Validar rota de busca de carrinho por id com sucesso;
- CT71 - Validar rota de busca de carrinho por id sem sucesso com token inválido;
- CT72 - Validar rota de busca de carrinho por id sem sucesso;
- CT73 - Validar rota de criação de carrinho com sucesso;
- CT74 - Validar rota de criação de carrinho com produto inexistente;
- CT75 - Validar rota de criação de carrinho sem sucesso com token inválido;
- CT76 - Validar rota de criação de carrinho com token ausente;
- CT77 - Validar rota de criação de carrinho com token expirado;
- CT78 - Validar rota de criação de carrinho com campos vazios;
- CT79 - Validar rota de criação de carrinho sem enviar parâmetros;
- CT80 - Validar rota de criação de carrinho enviando informação sem aspas;
- CT81 - Validar rota de criação de carrinho enviando informação sem aspas;
- CT82 - Validar alteração de carrinho com sucesso;
- CT83 - Validar alteração de carrinho sem sucesso com token inválido;
- CT84 - Validar rota de exclusão de carrinho ao concluir compra;



SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas da API ServeRest:

- CT85 - Validar rota de exclusão de carrinho inexistente ao concluir compra;
- CT86 - Validar rota de exclusão de carrinho ao concluir compra com token inválido;
- CT87 - Validar rota de exclusão de carrinho ao concluir compra com token ausente;
- CT88 - Validar rota de exclusão de carrinho ao concluir compra com token expirado;
- CT89 - Validar rota de exclusão de carrinho ao cancelar compra;
- CT90 - Validar rota de exclusão de carrinho inexistente ao cancelar compra;
- CT91 - Validar rota de exclusão de carrinho ao cancelar compra com token inválido;
- CT92 - Validar rota de exclusão de carrinho ao cancelar compra com token ausente;
- CT93 - Validar rota de exclusão de carrinho ao cancelar compra com token expirado;

Endereço

- CT94 - Validar rota de lista de endereços com sucesso;
- CT95 - Validar rota de lista de endereços sem sucesso com token inválido;
- CT96 - Validar rota de inclusão de endereço com sucesso;
- CT97 - Validar rota de inclusão de endereço sem sucesso com token inválido;
- CT98 - Validar rota de alteração de endereço com sucesso;
- CT99 - Validar rota de alteração de endereço sem sucesso com token inválido;
- CT100 - Validar rota de exclusão de endereço com sucesso;
- CT101 - Validar rota de exclusão de endereço sem sucesso com token inválido;

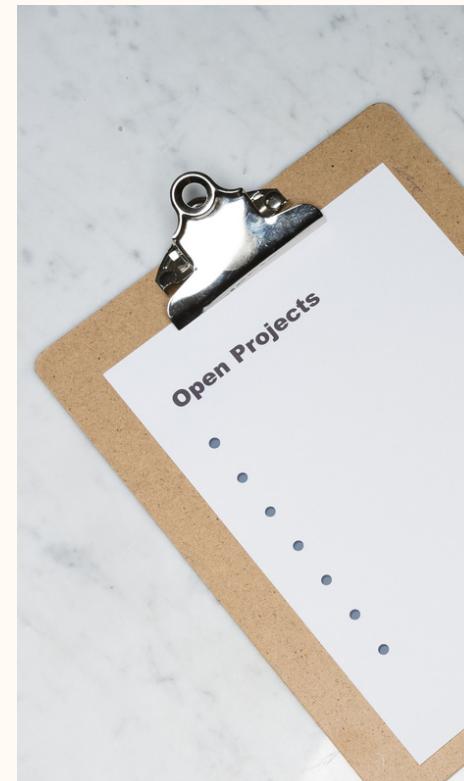


SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas da API ServeRest:

Pagamento

- CT102 - Validar rota de opções de pagamento com sucesso;
- CT103 - Validar rota de opções de pagamento sem sucesso;
- CT104 - Validar rota de criação de opções de pagamento com sucesso;
- CT105 - Validar rota de criação de opções de pagamento sem sucesso com token inválido;
- CT106 - Validar rota de alteração de opções de pagamento com sucesso;
- CT107 - Validar rota de alteração de opções de pagamento sem sucesso com token inválido;
- CT108 - Validar rota de exclusão de opções de pagamento com sucesso;
- CT109 - Validar rota de exclusão de opções de pagamento sem sucesso com token inválido.



ESTRATÉGIA DE TESTES

Níveis de testes:

Teste de Integração: Através da utilização da ferramenta POSTMAN, e acesso à documentação swagger da API ServeRest, deverá ser realizada a validação das respostas das rotas, em comparação com os resultados pré-determinados nos cenários de teste. Também deverá ser implementada automação dos testes da API ServeRest se utilizando o framework Cypress com a linguagem de programação JavaScript.

TESTES CANDIDATOS PARA A AUTOMAÇÃO

Lista de casos de teste para homologar as rotas da API ServeRest:

Login

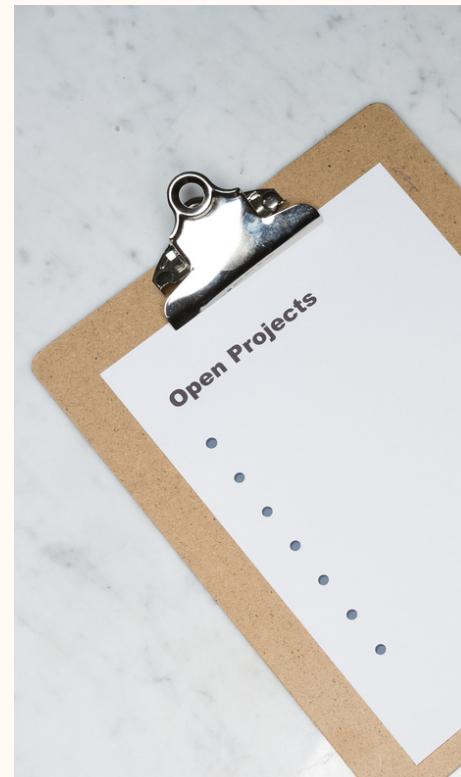
- CT01 - Validar rota de login com sucesso;
- CT02 - Validar rota de login email não cadastrado;
- CT03 - Validar rota de login com senha incorreta;

Usuários

- CT18 - Validar rota de lista de usuários;
- CT19 - Validar rota de lista de usuários sem autorização;
- CT20 - Validar rota de busca de usuário por id com sucesso;
- CT22 - Validar rota de busca de usuário por id sem sucesso com token inválido;
- CT23 - Validar rota de criação de usuário com sucesso;
- CT29 - Validar rota de criação de usuário com email já cadastrado;
- CT30 - Validar rota de alteração de usuário com sucesso;
- CT31 - Validar rota de criação de usuário ao tentar alterar usuário inexistente;
- CT32 - Validar rota de alteração de usuário sem sucesso com token inválido;
- CT38 - Validar rota de exclusão de usuário com sucesso;
- CT39 - Validar rota de exclusão de usuário sem sucesso com token inválido;

Produtos

- CT42 - Validar rota de lista de produtos com sucesso;
- CT43 - Validar rota de busca de produtos por id com sucesso;
- CT44 - Validar rota de busca de produtos por id sem sucesso;



TESTES CANDIDATOS PARA A AUTOMAÇÃO

Lista de casos de teste para homologar as rotas da API ServeRest:

- CT45 - Validar rota de cadastro de produtos com sucesso;
- CT47 - Validar rota de cadastro de produto sem sucesso com token inválido;
- CT53 - Validar rota de alteração de produto com sucesso;
- CT54 - Validar rota de criação de produto ao tentar alterar produto com id inexistente;
- CT56 - Validar rota de alteração de produto sem sucesso com token inválido;
- CT62 - Validar rota de exclusão de produtos com sucesso;
- CT65 - Validar rota de exclusão de produto sem sucesso com token inválido;

Carrinhos

- CT68 - Validar rota de lista de carrinhos;
- CT69 - Validar rota de lista de carrinhos sem sucesso com token inválido;
- CT70 - Validar rota de busca de carrinho por id com sucesso;
- CT71 - Validar rota de busca de carrinho por id sem sucesso com token inválido;
- CT73 - Validar rota de criação de carrinho com sucesso;
- CT75 - Validar rota de criação de carrinho sem sucesso com token inválido;
- CT82 - Validar alteração de carrinho com sucesso;
- CT83 - Validar alteração de carrinho sem sucesso com token inválido;



TESTES CANDIDATOS PARA A AUTOMAÇÃO

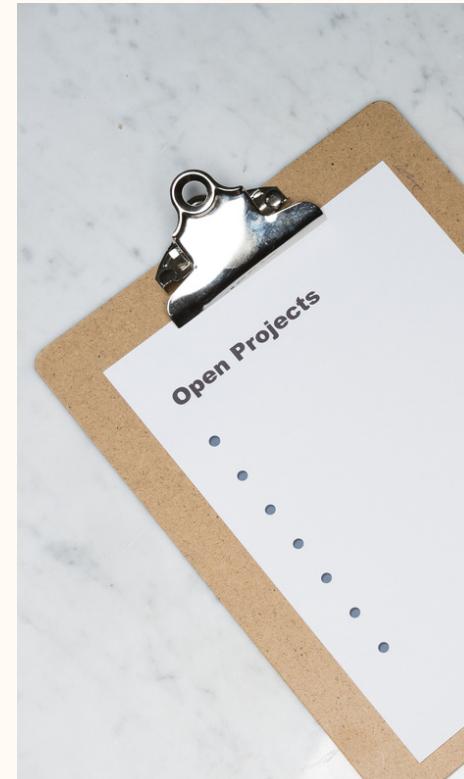
Lista de casos de teste para homologar as rotas da API ServeRest:

Endereços

- CT94 - Validar rota de lista de endereços com sucesso;
- CT95 - Validar rota de lista de endereços sem sucesso com token inválido;
- CT96 - Validar rota de inclusão de endereço com sucesso;
- CT97 - Validar rota de inclusão de endereço sem sucesso com token inválido;
- CT98 - Validar rota de alteração de endereço com sucesso;
- CT99 - Validar rota de alteração de endereço sem sucesso com token inválido;

Pagamento

- CT102 - Validar rota de opções de pagamento com sucesso;
- CT103 - Validar rota de opções de pagamento sem sucesso;



FERRAMENTAS

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

- **Navegadores de internet** (para consulta da documentação swagger da ServeRest e outras informações necessárias);
- **XMind** (ferramenta para elaborar o mapa mental da API);
- **Postman ou Insomnia** (para fazer interação manual com as rotas da ServeRest);
- **Visual Studio Code** (IDE para criação de código de automação dos testes);
- **JavaScript** (linguagem de programação para realizar a automação dos testes);
- **Cypress, Faker-JS, AJV, Mocha e Mochawesome** (frameworks para automação dos testes da API, emissão de reports e evidências de testes);
- **GIT e GitHub** (para versionamento e compartilhamento do código);
- **Captura de Tela do Windows** (para captura de tela com evidências dos testes);
- **Loom** (para evidências em vídeo);
- **Scrum** (metodologia ágil);
- **Jira** (para gerenciamento dos testes);
- **Zephyr** (extensão do Jira para a criação e execução dos testes);
- **Canva e Pacote Office** (para elaboração da documentação de testes);
- **Microsoft Teams** (para comunicações);
- **Outlook** (para formalizações por e-mail);



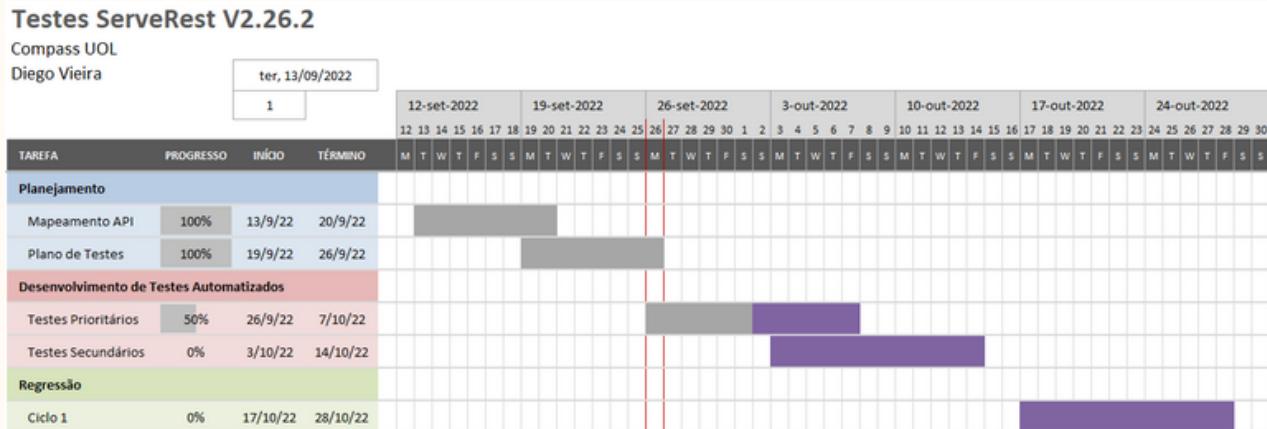
EQUIPE DE TESTES

A equipe de testes será formada pelos seguintes profissionais:

- **Líder de QA:** Diego Vieira
- **Analista de Testes:** Celina Maria
- **Arquiteta de Automação:** Camila Cristina
- **Dev. Automação Back-end 1:** Daniel Glen
- **Dev. Automação Back-end 2:** Silvia Souza
- **Testadora 1:** Sofia Silva
- **Testador 2:** João Bosco
- **Testadora 3:** Rosa Maria

CRONOGRAMA

No gráfico abaixo apresentamos o cronograma de atividades de teste que deverá ser respeitado. Nele temos atividades de execução organizadas por ordem de prioridade de testes:



CONCLUSÃO

O presente documento determina as atividades e processos mínimos para a criação e execução das atividades de teste da API ServeRest em sua versão 2.26.2. É possível e recomendado, 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.

Salientamos que já na fase de mapeamento e planejamento de testes verificamos alguns bugs que consideramos graves e que afetam o bom funcionamento da API, além de bugs de grávidade média e baixa. São exemplos de bugs graves a falta de algumas rotas como o método de alteração de carrinho pelo usuário, falta de rotas para cadastro de informações de entrega e endereços e a rota de pagamento. Também consideramos graves a falta de autenticação para assegurar as informações sensíveis de massa de dados dos usuários e carrinhos.

O relatório inicial de bugs anexo a este documento detalha todos os bugs encontrados até o momento e tende a sofrer modificações a medida que os testes avancem, ficando mais completo e detalhado.

Por fim, recomendamos que este Plano de Testes deve estar disponível e acessível para todos os interessados na API ServeRest.

Relatório de Bugs API ServeRest V2.26.2

Id	Descrição	Gravidade	Passos para reproduzir	Evidências
BG01	Deveria responder com status 400 ao fazer login com e-mail não cadastrado, mas responde com status 401	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método POST; 3 – Inserir endereço da rota: https://serverest.dev/login 4 – Inserir e-mail não cadastrado e senha no body em formato JSON; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EUAGylATacZPqtP-1sqbhwBsWlD7wi9wnRoX8p83WZW8Ew?e=h1G088
BG02	Na rota de login há mensagens diversas não documentadas, deveriam estar documentadas.	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método POST; 3 – Inserir endereço da rota: https://serverest.dev/login 4 – Enviar body sem informações; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EeVv4H6Kjx5JmGDbLIMN_rIB_WYz3pzFdjk3S5T7b0uFKg?e=aFt9Lq
BG03	API fornece dados sensíveis de seus administradores e usuários sem solicitar autenticação, deveria manter dados em sigilo.	Alta	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método GET; 3 – Inserir endereço da rota: https://serverest.dev/usuarios 4 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EQyPVQpZ_49Dv_Nr0VxWfuwByS3bwYHQibhjpFIGO2ZDg?e=gawz7a
BG04	API fornece dados na busca de usuário por id sem solicitar token válido, deveria fornecer apenas para administradores ou usuário do próprio id.	Alta	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método GET; 3 – Inserir endereço da rota: https://serverest.dev/usuarios/{_ID} 4 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EWwWwST1kwVDqXHy40luO60BNAmIbqDXiVzdtY7TqCJ7EA?e=RVe5SW
BG05	API permite criação de administradores e usuários sem exigir provas de identificação, senhas fortes e sem endereço, deveria melhorar identificação e força das senhas.	Alta	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método POST; 3 – Inserir endereço da rota: https://serverest.dev/usuarios 4 – Inserir body em formato JSON com informações básicas e senha fraca; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EcJ2V_fWY7tFuchsIDWl4QkbX2jug81lxR90T67t6W6-eQ?e=6AfXbY
BG06	Na rota de criação de usuários há mensagens diversas não documentadas, deveriam estar documentadas.	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método POST; 3 – Inserir endereço da rota: https://serverest.dev/usuarios 4 – Enviar body sem informações; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EZtyCZNGrMtJiFX1OAbrc5sBCbh7MGD5RIHM6ALMP-CdFQ?e=gzyfwn

BG07	API permite alterar dados de usuários sem solicitar token válido, deveria permitir apenas usuário do próprio id.	Alta	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método PUT; 3 – Inserir endereço da rota: https://serverest.dev/usuarios/{_ID} 4 – Inserir alterações de usuário no body em formato JSON; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com_br/EWuKUISA465Kos71kVQyzfkBRSMTGnT5jZ_HEfWBdtolPw?e=K2hsS
BG08	API informa que registro do usuário foi alterado sem nenhuma informação ter sido alterada, deveria informar que nada foi alterado.	Médio	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método PUT; 3 – Inserir endereço da rota: https://serverest.dev/usuarios/{_ID} 4 – Inserir as mesmas informações do usuário no body em formato JSON; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com_br/EbEOnVmzrv1FtyMp7g3h44BHLMNm2YkgKInc-nP1BG4IA?e=Z5z3sJ
BG09	API permite excluir dados de usuários sem solicitar token válido, deveria permitir apenas usuário do próprio id.	Alta	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método DELETE; 3 – Inserir endereço da rota: https://serverest.dev/usuarios/{_ID} 4 – Inserir alterações de usuário no body em formato JSON; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com_br/EdKVtpsWDgJKiElP7r3E6MB7ew-ZPT1-kTVQrBJCtY9eQ?e=6EjjWd
BG10	Na rota de alteração de usuários há mensagens diversas não documentadas, deveriam estar documentadas.	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método PUT; 3 – Inserir endereço da rota: https://serverest.dev/usuarios/{_ID} 4 – Enviar body sem informações; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EZtpWImuXM9FgrDry4IKtLYBhZl1zOXTz7J5Z5pxt5kCNQ?e=URzJxo
BG11	Na rota de exclusão de usuários a API usa o mesmo status code para enviar mensagens com sentidos opostos, deveria aproveitar melhor a sintaxe dos status code.	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método DELETE; 3 – Inserir endereço da rota com usuário inexistente: https://serverest.dev/usuarios/{_ID} 4 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EZ5qB8guFoxOgEel6BwqTdUBJZpayKVBoCKqaTl3Teut9A?e=m3CMIj
BG12	Há falhas na redação da descrição do método em sua documentação swagger.	Baixa	No navegador: 1 – Inserir endereço da API: https://serverest.dev/ 2 – Abrir a aba PUT em Produtos;	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EZxFW6_WRTFHuwptPctWgB78qEDkPORCcYsBhHpLGeRg?e=lpD89v

BG13	Na rota de alteração de produtos há mensagens diversas não documentadas, deveriam estar documentadas.	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método PUT; 3 – Inserir endereço da rota: https://serverest.dev/produtos/{_ID} 4 – Enviar body sem informações; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EQkdXhIlvo1EkzDKvCH4SI4BDIrcfEjGT3Q7lcUqliAf6Q?e=dd98CA
BG14	Na rota de criação de produtos há mensagens diversas não documentadas, deveriam estar documentadas.	Baixa	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método POST; 3 – Inserir endereço da rota: https://serverest.dev/produtos/{_ID} 4 – Enviar body sem informações; 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EWtcd88De7NDke5AqOzHKrcB8PcbVu5z892UDdIE2l8Dw?e=KlEFYz
BG15	Na rota carrinhos a API fornece dados sensíveis dos carrinhos de todos os usuários sem solicitar autenticação, deveria manter dados em sigilo.	Alta	No Postman: 1 – Abrir aba de nova requisição; 2 – Selecionar o método GET; 3 – Inserir endereço da rota: https://serverest.dev/carrinhos 4 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EVWXPUfcA1DljJbfBCWOMoB1RWkFeluiwhbeW5CraSaw?e=hWneHi
BG16	API fornece dados na busca de carrinhos por id sem solicitar token válido, deveria fornecer apenas para administradores ou usuário do próprio id.	Alta	No Postman: 1 – Listar carrinhos de usuários e copiar id de carrinho existente; 2 – Abrir aba de nova requisição; 3 – Selecionar o método GET; 4 – Inserir endereço da rota: https://serverest.dev/carrinhos/{_ID} 5 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/ETKevPrTMoVGqlmZUCq3O18BfE4CERnBG39is52gVmbgqA?e=072ohW
BG17	A API não oferece o método PUT na rota carrinhos, impossibilitando que o usuário faça alterações, deveria permitir ações nesse sentido.	Alta	No navegador: 1 – Inserir endereço da API: https://serverest.dev/ 2 – Abrir a aba Carrinhos;	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/ESfmkWDjIEIKkoNz9u8nyb8Bg6CKTqlZoHGemsGhxphJkA?e=BFavDe
BG18	Na rota de exclusão de carrinho ao concluir compra API usa mesmo status code para enviar mensagens com sentidos opostos, deveria aproveitar melhor a sintaxe dos status code.	Baixa	No Postman: 1 – Logar com usuário existente sem carrinho e copiar token; 4 – Inserir endereço da rota: https://serverest.dev/carrinhos/concluir-compra 5 – Clicar em “Send” usando o token.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com.br/EUO8TFWZw0FGt4WCXUqxxPYBStS-G76-m7wmUyN1GWNFcA?e=OABw19

BG19	Na rota de exclusão de carrinho ao cancelar compra API usa mesmo status code para enviar mensagens com sentidos opostos, deveria aproveitar melhor a sintaxe dos status code.	Baixa	No Postman: 1 – Logar com usuário existente sem carrinho e copiar token; 4 – Inserir endereço da rota: https://serverest.dev/carrinhos/cancelar-compra 5 – Clicar em “Send” usando o token.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com_br/ERYBjAcrS8xIild1ua1-pzwBYNU-78acvVHwJUsqzqwOdw?e=BwNTgo
BG20	A API não apresenta a rota de endereços, deveria possibilitar o cadastro, busca, alteração e exclusão de endereços.	Alta	No navegador: 1 – Inserir endereço da API: https://serverest.dev/ 2 – Analisar documentação.	
BG21	A API não apresenta a rota de pagamento, deveria possibilitar o cadastro, busca, alteração e exclusão de endereços.	Alta	No navegador: 1 – Inserir endereço da API: https://serverest.dev/ 2 – Analisar documentação.	
BG22	Na rota de exclusão de produtos API usa mesmo status code para enviar mensagens com sentidos opostos, deveria aproveitar melhor a sintaxe dos status code.	Baixa	No Postman: 1 – Logar com usuário existente e copiar token; 2 – Selecionar o método DELETE; 3 – Inserir endereço da rota com produto inexistente: https://serverest.dev/produtos/{_ID} 4 – Clicar em “Send”.	https://compasso-my.sharepoint.com/:i/g/personal/diego_vieira_pb_compasso_com_br/Ed41sWIub-BOuFuBUZPbkr0BPiVPdXryn46yrHAx1BNR8Q?e=DIIWB5