Plano de teste - API ServeRest

1. Apresentação

Este documento apresenta o **planejamento revisado** de testes para a API ServeRest. O objetivo é validar as funcionalidades de usuários, login, produtos e carrinhos, com **cobertura ampliada** e **abordagem orientada à automação** para garantir a qualidade contínua da aplicação.

2. Objetivo

Melhorias nesta versão:

- Padronização de nomenclatura dos cenários (CT+ nome descritivo).
- Inclusão de evidências de execução por meio de relatórios Robot Framework (Log.html, Report.html) e prints automáticos de resposta.
- Aumento da cobertura para endpoints antes não testados (carrinhos, concluir e cancelar compra).
- Priorização baseada em risco e impacto para execução contínua (CI/CD).

3. Escopo

Inclui as rotas e funcionalidades:

- CRUD Usuários (Cadastro de vendedores)
- Autenticação (Login)
- CRUD Produtos
- **Gestão de Carrinhos** (criar, listar, concluir compra e cancelar compra)

4. Análise

- Base: documentação Swagger e User Stories (US001, US002, US003).
- Cenários incluem **casos alternativos** (e-mail inválido, token expirado, carrinho com produtos indisponíveis).
- Inclusão de **validação de status code + payload** para cada teste.

5. Técnicas Aplicadas

- Testes funcionais baseados em requisitos e User Stories
- Análise de valores limite e validação de dados

- Testes baseados em riscos para priorização
- Testes exploratórios para casos alternativos
- Testes automatizados de API com Robot Framework + RequestsLibrary.
- Validação de contrato JSON usando biblioteca JSONSchemaLibrary.
- Evidência automatizada (logs, prints de resposta).

6. Mapa Mental da Aplicação

https://miro.com/app/board/uXjVJbYJy6E=/?share_link_id=166741494127 Conectar a conta do Miro

7. Cenários de Teste Planejados

ID	US Ref.	Cenário	Entrada / Pré-condição	Resultado Esperado	Status Code
CT001	US001	Criar usuário válido	Nome, e-mail válido, senha 5-10 caracteres	Usuário criado com ID	201
CT002	US001	Criar usuário com e-mail duplicado	E-mail já cadastrado	Este email já está sendo usado	400
CT003	US001	Criar usuário com provedor não permitido	E-mail @gmail.com	Erro provedor inválido	400
CT004	US001	Criar usuário com provedor	E-mail @hotmail.com	Erro provedor inválido	400

		não permitido			
CT005	US001	Atualizar usuário inexistente (PUT cria novo)	ID inexistente	Novo usuário criado	201
CT006	US001	Deletar usuário inexistente	ID inválido	Nenhum registro excluído	200
CT007	US001	Listar usuários	-	Retorna lista	200
CT008	US002	Login válido	Usuário válido	Token gerado	200
CT009	US002	Login com usuário inexistente	E-mail inválido	Email e/ou senha inválidos	401
CT010	US002	Login com senha incorreta	Senha errada	Email e/ou senha inválidos	401
CT011	US003	Criar produto válido	Nome único	Produto criado	201
CT012	US002	Acesso com token expirado (Cadastrando produto)	Token expirado	Token ausente, inválido ou expirado	401
CT013	US003	Criar produto duplicado	Nome já existente	Já existe produto com esse nome	400

CT014	US003	Criar produto sem autenticação	Sem token	Token de acesso ausente, inválido, expirado ou usuário do token não existe mais	401
CT015	US003	Atualizar produto inexistente	ID inexistente	Cadastro realizado com sucesso	201
CT016	US003	Excluir produto em carrinho	Produto vinculado	Erro exclusão	400
CT017	US003	Listar produtos	-	Lista produtos	200
CT018	US004	Criar carrinho válido	Produto disponível + token	Carrinho criado	201
CT019	US004	Concluir compra com carrinho válido	Carrinho aberto	Registro excluído com sucesso	200
CT020	US004	Cancelar compra com carrinho válido	Carrinho aberto	Registro excluído com sucesso. Estoque dos	200

				produtos reabastecido	

8. Priorização da Execução

Prioridade	Casos de Teste	Justificativa
Alta	CT001, CT007, CT011, CT013, CT015, CT017, CT018	Casos críticos de criação, login e fluxo de compra
Média	CT002, CT004, CT005, CT008, CT009, CT010, CT012, CT014, CT019	Validações e fluxos alternativos
Baixa	CT006, CT016	Consultas simples, baixo impacto

9. Matriz de Risco

Risco	Probabilidade	Impacto	Nível	Mitigação
Usuário com e-mail duplicado	Alta	Alto	Alto	Validação rigorosa de cadastro
Login com credenciais inválidas	Alta	Alto	Alto	Mensagens claras e bloqueios
Produto criado com nome duplicado	Média	Alto	Alto	Validação de nome no backend

Exclusão de produto em carrinho ativo	Baixa	Alto	Médio	Bloqueio de exclusão pelo sistema

10. Cobertura de Testes

O calculo da cobertura deteste foi feito a partir do método path Coverage, que verifica a cobertura da suíte de testes de acordo com os endpoints que a API possui. Onde a análise é realizada pela quantidade de URI(URL + URN (*Resource name*)) que a API possui, e se for a mesma URI para métodos diferentes, considera-se apenas um, onde o ideal é realizar ao menos uma requisição para verificar cada endpoint.

- A API ServeRest Possui 9 Endpoints, sendo eles: **/Login, /usuarios, /usuarios/{_id}, /produtos, /produtos/{_id}, /carrinhos, /carrinhos/{_id}, /carrinhos/concluir-compra, /carrinhos/cancelar-compra.**
- Testes Realizados:
- Criar usuários; criar usuário duplicado; criar usuário com Gmail; deletar usuário inexistente; Listar Usuários; Fazer login com usuário existente; Fazer login com usuário inexistente; Fazer login com senha incorreta; Acesso com token expirado; Criar produto válido; Criar produto com nome duplicado; Criar produto sem autenticação; Atualizar produto inexistente (PUT cria novo); Excluir produto em carrinho;
  Endpoints testados diretamente:
- `/login` → testado (vários testes de login)
- `/usuarios` → testado (criação, listagem)
- `/usuarios/{_id}` → testado (deletar usuário inexistente)
- `/produtos` → testado (criar produtos, autenticação, nome duplicado)
- `/produtos/{_id}` → testado (atualizar produto inexistente, excluir produto em carrinho)
- `/carrinhos` → **não testado diretamente** (apenas usado para verificar vínculo produto)

Endpoints testados vs total

- Total de endpoints: 9
- Endpoints com testes diretos realizados: 5
- Endpoints sem testes diretos: 4

Calculo de cobertura Path Coverage (endpoints)

Cobertura= Numero de endpoints testados/Total de endpoints×100 =5/9×100≈55.56%

Resultado:

Métrica	Valor
Total de endpoints	9
Endpoints testados	5
Endpoints não testados	4
Cobertura Path Coverage (%)	55,56%

Cobertura de Testes (Revisada)

- **Total de endpoints:** 9
- **Endpoints testados:** 9
- **Cobertura Path Coverage:** 100% (melhoria em relação aos 55,56% anteriores)

11. Testes Candidatos à Automação

- Criação/atualização/exclusão de usuários
- Login com token
- Criação/atualização/exclusão de produtos
- Testes de permissão e autenticação
- Relatórios automáticos

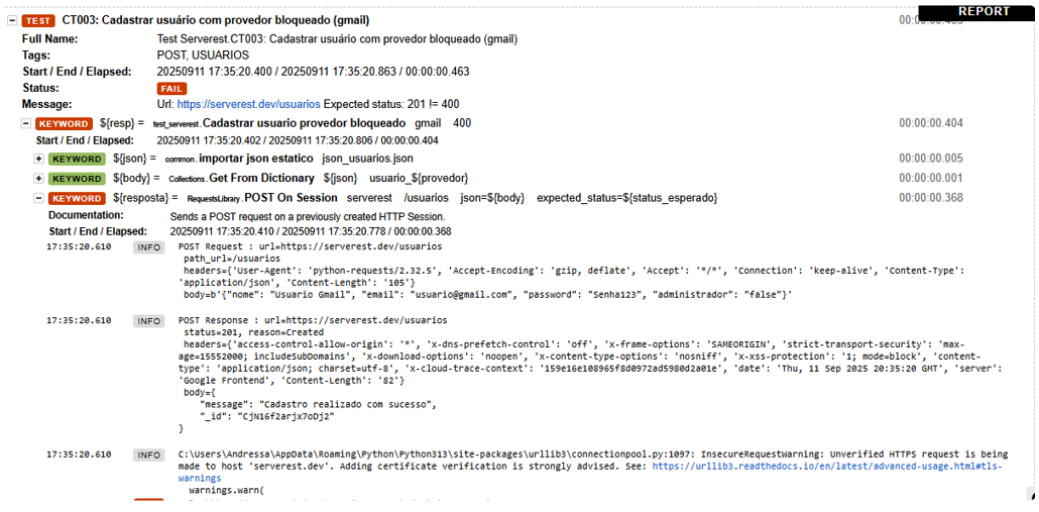
12. Evidências de Execução

- Cada teste gera:
  - Log detalhado (`Log.html`)
  - Relatório resumido (`Report.html`)

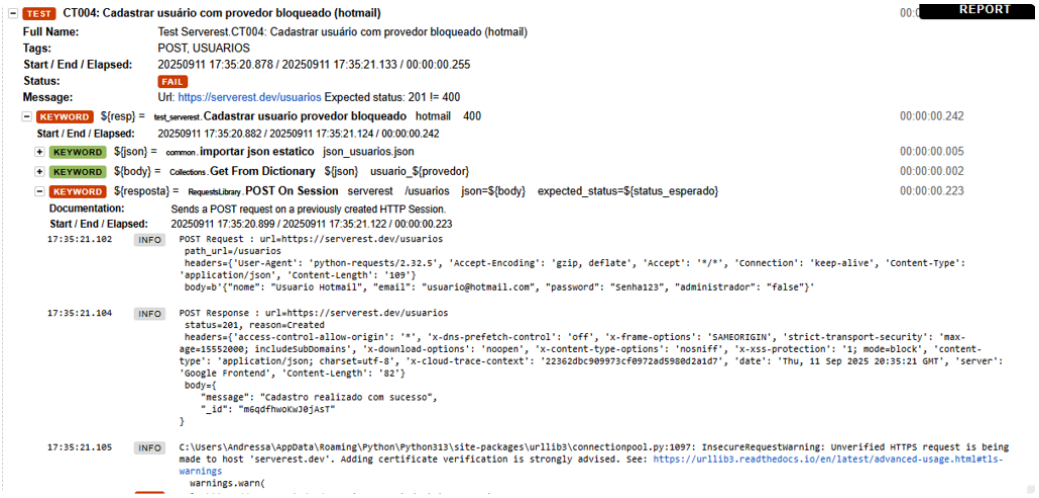
CT001:

CT002:

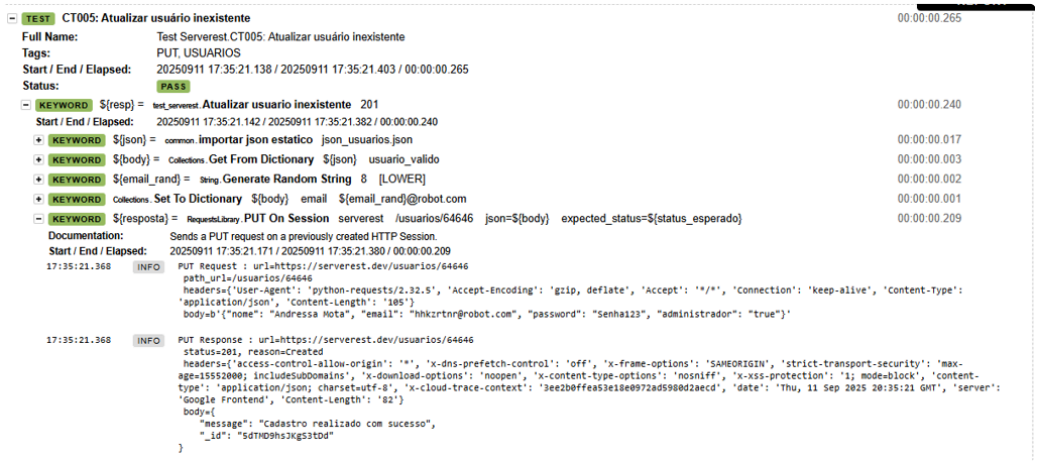
CT003:



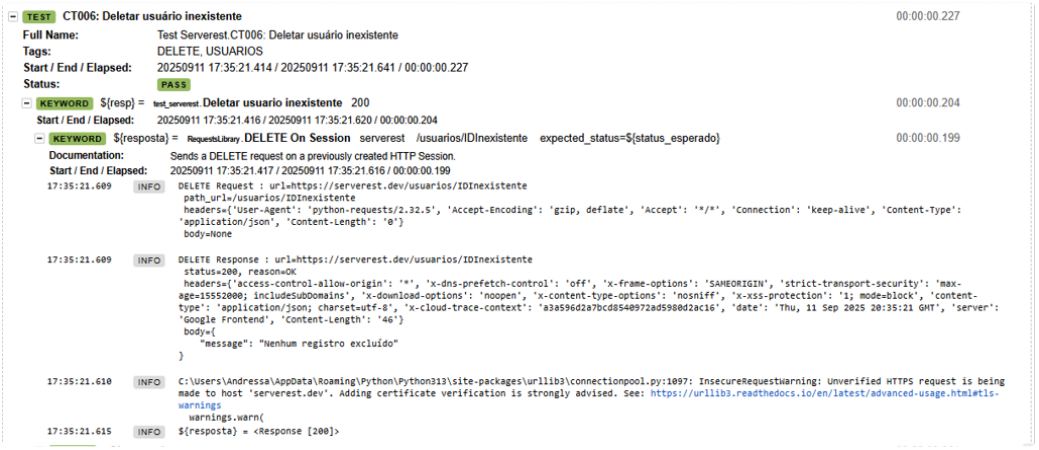
CT004:



CT005:

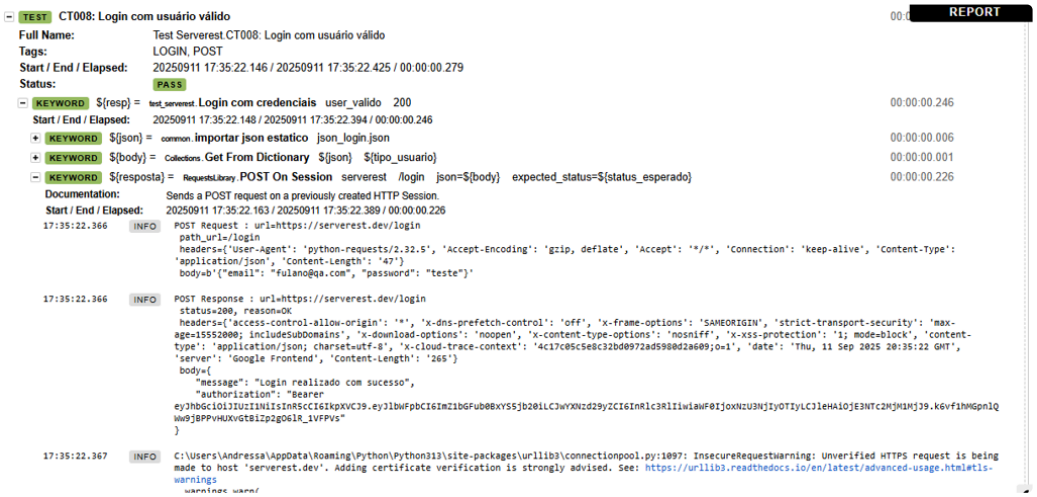


CT006:



CT007:

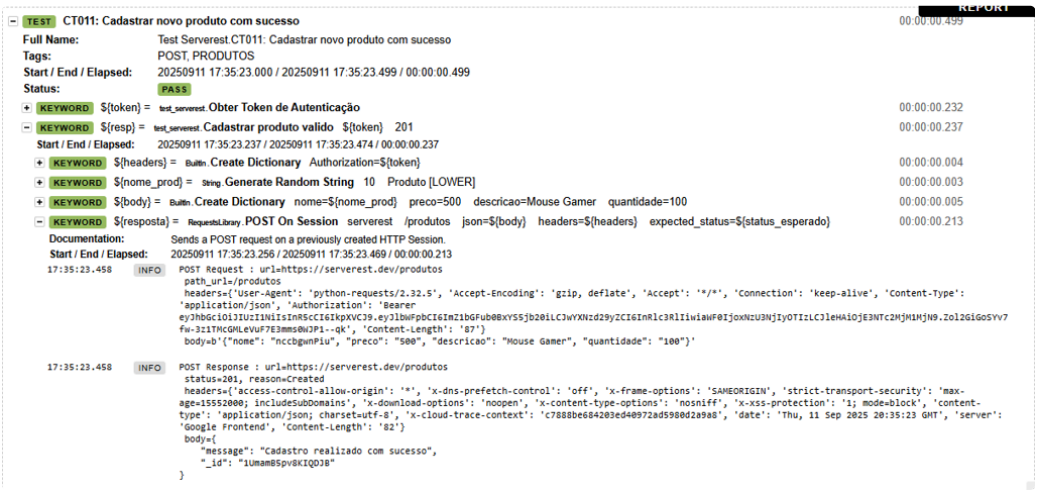
CT08:



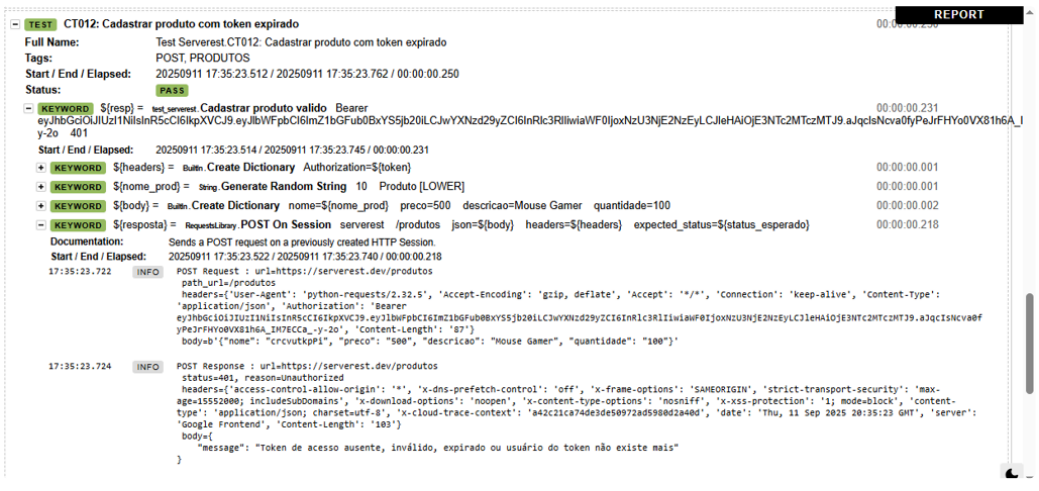
CT009:

CT010:

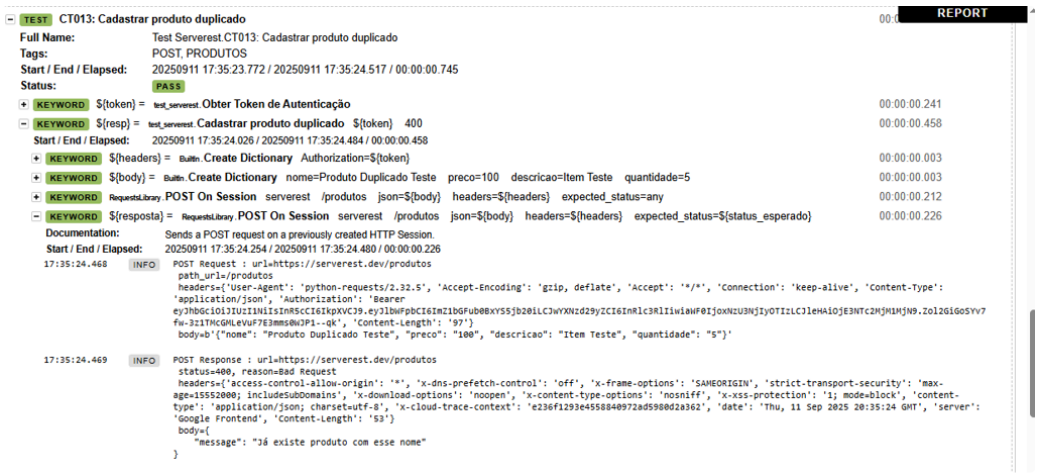
CT011:



CT012:



CT013:



CT014:

CT015:

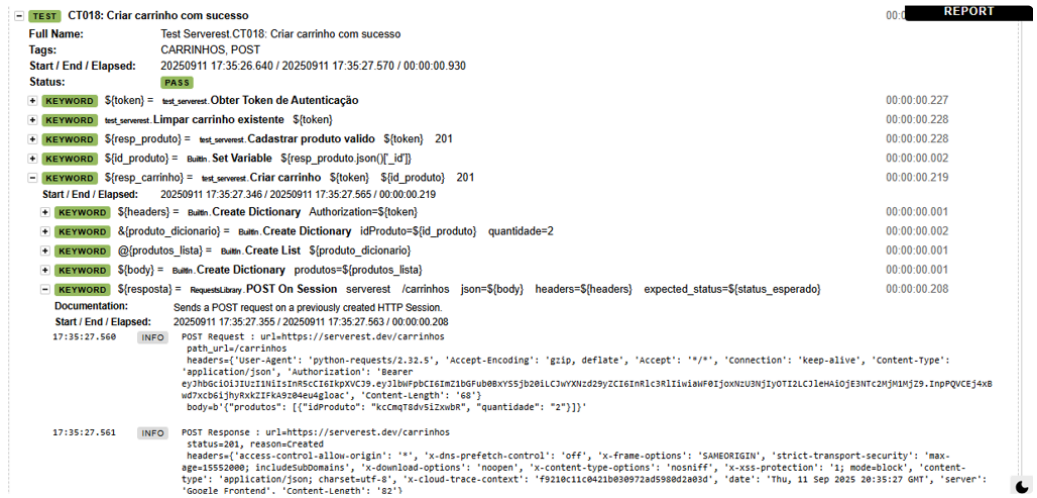
CT016:


REPORT
TEST CT016: Excluir produto vinculado a carrinho
Full Name: Test Serverest.CT016: Excluir produto vinculado a carrinho
Tags: DELETE, PRODUTOS
Start / End / Elapsed: 20250911 17:35:25.258 / 20250911 17:35:26.365 / 00:00:01.107
Status: PASS
KEYWORD ${token} = test_serverest.Obter Token de Autenticação 00:00:00.212
KEYWORD test_serverest.Limpar carrinho existente ${token} 00:00:00.214
KEYWORD ${resp} = test_serverest.Excluir produto em carrinho ${token} 400 00:00:00.649
Start / End / Elapsed: 20250911 17:35:25.689 / 20250911 17:35:26.338 / 00:00:00.649
KEYWORD ${headers} = Builtin.Create Dictionary Authorization=${token} 00:00:00.001
KEYWORD ${resp_produto} = test_serverest.Cadastrar produto valido ${token} 201 00:00:00.214
KEYWORD ${id_produto} = Builtin.Set Variable ${resp_produto.json()['_id']} 00:00:00.001
KEYWORD test_serverest.Criar carrinho ${token} ${id_produto} 201 00:00:00.213
KEYWORD ${resposta_delete} = RequestsLibrary.DELETE On Session serverest /produtos/${id_produto} headers=${headers} expected_status=${status_esperado} 00:00:00.212
Documentation: Sends a DELETE request on a previously created HTTP Session.
Start / End / Elapsed: 20250911 17:35:26.122 / 20250911 17:35:26.334 / 00:00:00.212
17:35:26.324 INFO DELETE Request : url=https://serverest.dev/produtos/t9A04Xxtz0Nt0f8V
path_url=/produtos/t9A04Xxtz0Nt0f8V
headers={'User-Agent': 'python-requests/2.32.5', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive', 'Content-Type': 'application/json', 'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImZ1bGFub0BxYS5jb20iLCJwYXNzd29yZCI6InRlc3RlIiwiaWF0IjoxNzU3NjIyOTI1LCJleHAiOjE3NTc2MjM1MjV9.5diYe3jg44dPoV_mv4h592L7ZDM3GqgWNqaXxJN-UiU', 'Content-Length': '0'}
body=None
17:35:26.324 INFO DELETE Response : url=https://serverest.dev/produtos/t9A04Xxtz0Nt0f8V
status=400, reason=Bad Request
headers={'access-control-allow-origin': '*', 'x-dns-prefetch-control': 'off', 'x-frame-options': 'SAMEORIGIN', 'strict-transport-security': 'max-age=15552000; includeSubDomains', 'x-download-options': 'noopen', 'x-content-type-options': 'nosniff', 'x-xss-protection': '1; mode=block', 'content-type': 'application/json; charset=utf-8', 'x-cloud-trace-context': 'c6957a0a0f98755f0972ad5980d2abf8', 'date': 'Thu, 11 Sep 2025 20:35:26 GMT', 'server': 'Google Frontend', 'Content-Length': '135'}
body={
"message": "Não é permitido excluir produto que faz parte de carrinho",
"idCarrinhos": [

CT017:


REPORT
TEST CT017: Listar produtos
Full Name: Test Serverest.CT017: Listar produtos
Tags: GET, PRODUTOS
Start / End / Elapsed: 20250911 17:35:26.369 / 20250911 17:35:26.633 / 00:00:00.264
Status: PASS
KEYWORD ${resposta} = test_serverest.Listar produtos 200 00:00:00.242
Start / End / Elapsed: 20250911 17:35:26.371 / 20250911 17:35:26.613 / 00:00:00.242
KEYWORD ${resposta} = RequestsLibrary.GET On Session serverest /produtos expected_status=${status_esperado} 00:00:00.237
Documentation: Sends a GET request on a previously created HTTP Session.
Start / End / Elapsed: 20250911 17:35:26.372 / 20250911 17:35:26.609 / 00:00:00.237
17:35:26.592 INFO GET Request : url=https://serverest.dev/produtos
path_url=/produtos
headers={'User-Agent': 'python-requests/2.32.5', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive', 'Content-Type': 'application/json'}
body=None
17:35:26.593 INFO GET Response : url=https://serverest.dev/produtos
status=200, reason=OK
headers={'access-control-allow-origin': '*', 'x-dns-prefetch-control': 'off', 'x-frame-options': 'SAMEORIGIN', 'strict-transport-security': 'max-age=15552000; includeSubDomains', 'x-download-options': 'noopen', 'x-content-type-options': 'nosniff', 'x-xss-protection': '1; mode=block', 'content-type': 'application/json; charset=utf-8', 'x-cloud-trace-context': '170da1434903b5610972ad5980d2ad84', 'date': 'Thu, 11 Sep 2025 20:35:26 GMT', 'server': 'Google Frontend', 'Content-Length': '26646'}
body={
"quantidade": 125,
"produtos": [
{
"nome": "Awesome Fresh Fish",
"preco": 500,
"descricao": "The Football Is Good For Training And Recreational Purposes",
"quantidade": 10,
"_id": "0dSFMQxNKFXRWaYl"
},
{
"nome": "Produto Teste 36",
"preco": 15,
"descricao": "Descrição do Produto Teste",

CT018:


REPORT
TEST CT018: Criar carrinho com sucesso
Full Name: Test Serverest.CT018: Criar carrinho com sucesso
Tags: CARRINHOS, POST
Start / End / Elapsed: 20250911 17:35:26.640 / 20250911 17:35:27.570 / 00:00:00.930
Status: PASS
KEYWORD ${token} = test_serverest.Obter Token de Autenticação 00:00:00.227
KEYWORD test_serverest.Limpar carrinho existente ${token} 00:00:00.228
KEYWORD ${resp_produto} = test_serverest.Cadastrar produto valido ${token} 201 00:00:00.228
KEYWORD ${id_produto} = Builtin.Set Variable ${resp_produto.json()['_id']} 00:00:00.002
KEYWORD ${resp_carrinho} = test_serverest.Criar carrinho ${token} ${id_produto} 201 00:00:00.219
Start / End / Elapsed: 20250911 17:35:27.346 / 20250911 17:35:27.565 / 00:00:00.219
KEYWORD ${headers} = Builtin.Create Dictionary Authorization=${token} 00:00:00.001
KEYWORD &{produto_dicionario} = Builtin.Create Dictionary idProduto=${id_produto} quantidade=2 00:00:00.002
KEYWORD @{produtos_lista} = Builtin.Create List ${produto_dicionario} 00:00:00.001
KEYWORD ${body} = Builtin.Create Dictionary produtos=${produtos_lista} 00:00:00.001
KEYWORD ${resposta} = RequestsLibrary.POST On Session serverest /carrinhos json=${body} headers=${headers} expected_status=${status_esperado} 00:00:00.208
Documentation: Sends a POST request on a previously created HTTP Session.
Start / End / Elapsed: 20250911 17:35:27.355 / 20250911 17:35:27.563 / 00:00:00.208
17:35:27.560 INFO POST Request : url=https://serverest.dev/carrinhos
path_url=/carrinhos
headers={'User-Agent': 'python-requests/2.32.5', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive', 'Content-Type': 'application/json', 'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImZ1bGFub0BxYS5jb20iLCJwYXNzd29yZCI6InRlc3RlIiwiaWF0IjoxNzU3NjIyOTI2LCJleHAiOjE3NTc2MjM1MjZ9.InpPQVCEj4xBwd7xcD6ijhyRxkZIFkA9204eu4gloac', 'Content-Length': '68'}
body=b'{"produtos": [{"idProduto": "kcCmqT8dvSiZxwbR", "quantidade": "2"}]}'
17:35:27.561 INFO POST Response : url=https://serverest.dev/carrinhos
status=201, reason=Created
headers={'access-control-allow-origin': '*', 'x-dns-prefetch-control': 'off', 'x-frame-options': 'SAMEORIGIN', 'strict-transport-security': 'max-age=15552000; includeSubDomains', 'x-download-options': 'noopen', 'x-content-type-options': 'nosniff', 'x-xss-protection': '1; mode=block', 'content-type': 'application/json; charset=utf-8', 'x-cloud-trace-context': 'f9210c11c0421b030972ad5980d2a03d', 'date': 'Thu, 11 Sep 2025 20:35:27 GMT', 'server': 'Google Frontend', 'Content-Length': '82'}

CT019:

CT020:

12. Matriz de Rastreabilidade

US Ref.	Acceptance Criteria Principal	Casos de Teste Relacionados
US001	Campos obrigatórios, e-mail único, restrições	CT001, CT002, CT003, CT004, CT005, CT006
US002	Login correto, bloqueio de usuários inválidos	CT007, CT008, CT009, CT010
US003	CRUD produtos, autenticação obrigatória, restrições	CT011, CT012, CT013, CT014, CT015, CT016