

# Relatório de Testes - API ServeRest

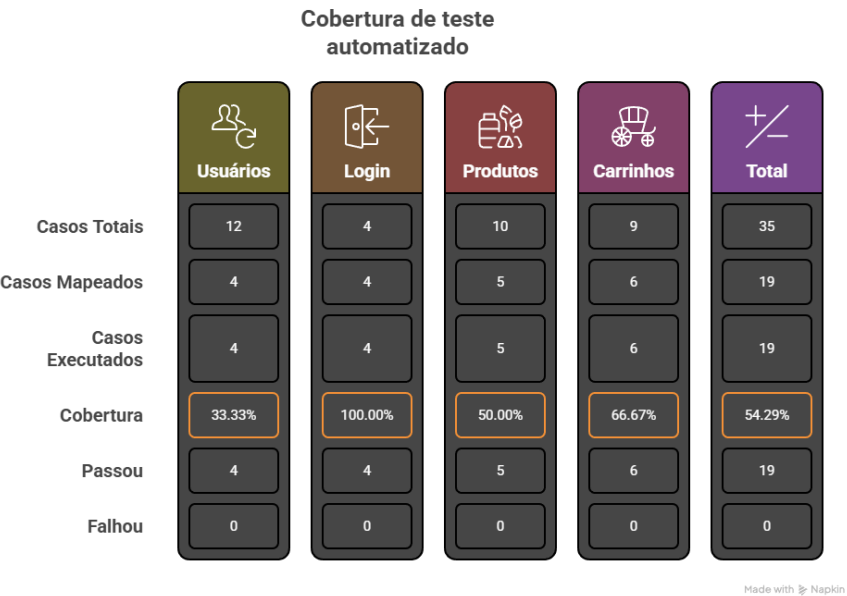
Este relatório apresenta os resultados obtidos a partir da execução dos testes definidos no [Plano de Testes](#) da API ServeRest. Foram contemplados testes manuais (executados via Postman e monitorados pelo QAlity) e testes automatizados (executados via Robot Framework em EC2 AWS). O objetivo é fornecer uma visão clara sobre a estabilidade, qualidade e conformidade da API com os requisitos definidos

## Ferramentas utilizadas:

- Postman ( testes manuais)
- Robot Framework ( teste automatizado)
- AWS AC2 ( hospedagem api e robot)
- VS Code (Desenvolvimento dos testes )
- Jira ( documentação issues )
- Quality(jira) (monitoramento dos testes)

## Cobertura de testes

No contexto deste projeto, a cobertura foi analisada separadamente para testes manuais e automatizados. Os testes manuais foram responsáveis por validar todos os casos planejados, abrangendo diferentes funcionalidades da API, incluindo casos positivos e negativos. Já os testes automatizados focaram nos fluxos principais e determinísticos, garantindo uma base confiável para execuções frequentes e regressões controladas



## Cobertura de teste automatizado

	Usuários	Login	Produtos	Carrinhos	Total
Casos Totais	12	4	10	9	35
Casos Mapeados	4	4	5	6	19
Casos Executados	4	4	5	6	19
Cobertura	33.33%	100.00%	50.00%	66.67%	54.29%
Passou	4	4	5	6	19
Falhou	0	0	0	0	0

Made with  Napkin

## Resumo da execução

A execução dos testes foi realizada com base nos 35 casos mapeados no plano de testes, contemplando as funcionalidades de cadastro de usuários, login, produtos e carrinhos. Todos os testes planejados foram executados manualmente utilizando o Postman, com registros integrados ao QAlity no Jira. Paralelamente, 19 testes automatizados foram executados via Robot Framework, hospedado em uma instância EC2 da AWS, com integração à API exposta em outra instância EC2.

## Principais falhas detectadas

### [CT-011 -Listagem de usuários sem autenticação permitida](#)

Ao realizar uma requisição para listar os usuários sem enviar um token de autenticação, a API retorna um status 200 OK e lista todos os usuários cadastrados no sistema. Este comportamento é crítico, pois permite acesso a dados sensíveis sem qualquer tipo de validação, violando princípios de segurança e proteção de dados

### [CT-024- Listagem de produtos sem autenticação permitida pela API](#)

Ao realizar uma requisição para listar produtos sem enviar um token de autenticação, a API retorna um status 200 OK e lista todos os produtos cadastrados no sistema. Esse comportamento não condiz com as práticas de segurança para APIs RESTful, nas quais o acesso a recursos sensíveis deve ser restrito a usuários autenticados.

### [CT-003 - Cadastro permitido com domínio de e-mail bloqueado \(Gmail, Hotmail\).](#)

Ao realizar um cadastro de usuário utilizando um e-mail com domínio bloqueado a API retorna um status 201 Created, indicando que o cadastro foi realizado com sucesso, mesmo quando o domínio está explicitamente bloqueado.

Conforme as regras de negócio, não deveria ser permitido o cadastro de e-mails com domínios bloqueados, como Gmail e Hotmail.

### [CT-005 Cadastro permitido com senha fora do limite permitido \(menor que 5 ou maior que 10 caracteres\)](#)

Ao realizar um cadastro de usuário utilizando uma senha com menos de 5 caracteres ou mais de 10 caracteres, a API permite o cadastro com sucesso, retornando um 201 Created com a seguinte mensagem: "Cadastro realizado com sucesso". Contudo, o esperado é que a API retorne um erro 400 Bad Request, indicando que a senha deve ter entre 5 e 10 caracteres

### [CT-018-Mensagem de erro para preço negativo não documentada no Swagger](#)

### [CT-020-Mensagem de erro com quantidade negativa não documentada no Swagger](#)

Mensagens não está documentada no Swagger, gerando falta de alinhamento entre a documentação e o comportamento da API.

## Resultados gerados:

### Ciclo de teste manual via QAlity

[ServeRest Tests- QAlity](#)

Description:  
A report that summarizes tests that were executed in the selected period.



Testes manuais realizados via Postman

### Rota Usuarios

Test Cases [Overview](#)

Project: **ServeRest Tests- QAlity**

Version: No version

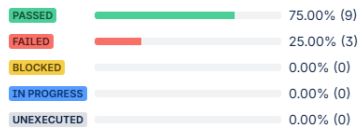
#### Description

Gerenciar os usuários, consultar dados para login e cadastro administrador.

Created: 30/May/2025

Due date: No date

12 Test Cases in the Test Cycle



### Rota Login

Test Cases [Overview](#)

Project: **ServeRest Tests- QAlity**

Version: No version

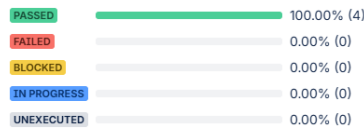
#### Description

Autentixar o seu usuário para montar um carrinho e, se for administrador, gerenciar os produtos

Created: 30/May/2025

Due date: No date

4 Test Cases in the Test Cycle



### Rota Produto

Test Cases [Overview](#)

Project: **ServeRest Tests- QAlity**

Version: No version

#### Description

Consultar produtos cadastrados ou, como administrador, gerenciar os produtos da loja

Created: 30/May/2025

Due date: No date

10 Test Cases in the Test Cycle



### Rota Carrinho

Test Cases [Overview](#)

Project: **ServeRest Tests- QAlity**

Version: No version

#### Description

Consultar os carrinhos cadastrados,

Created: 30/May/2025

Due date: No date

9 Test Cases in the Test Cycle



## Traceability Report

Description:  
This report summarizes bugs associated with test cases and their executions in the selected period.

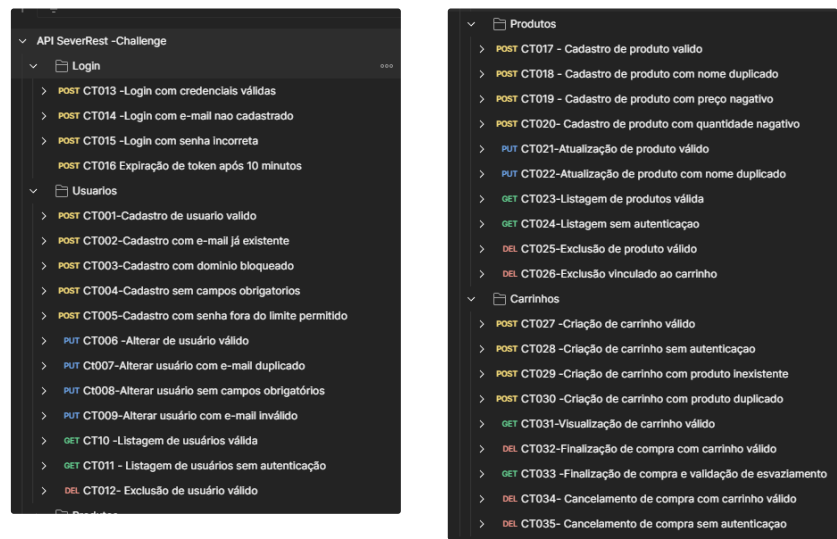
**Filters:**

Project: **ServeRest Tests- QAlity (ST)** Date range: **07/03/2025** → **06/06/2025** Requirement: **Bug** Search requirement: **More** **Refresh report**

Requirement	Test Case	Test Executions	Bugs
01-170 CT-003 - Cadastro permitido com domínio de e-mail bloqueado (Gmail, Hotmail).	01-170 CT-003 - Cadastro com domínio bloqueado Gmail, Hotmail	2 test executions Label: 30/May/2025 - Passed	1 bug found 0% resolved → 100% open → 1 new
01-171 CT-005 Cadastro permitido com senha fora do limite permitido (menor que 5 ou maior que 30 caracteres)	01-171 CT-005 - Cadastro com senha fora do limite permitido	2 test executions Label: 30/May/2025 - Failed	1 bug found 0% resolved → 100% open → 1 new
01-172 CT-011 - Listagem de usuários sem autenticação permitida	01-172 CT-011 - Listagem de usuários sem autenticação	2 test executions Label: 30/May/2025 - Failed	1 bug found 0% resolved → 100% open → 1 new
01-173 CT-019 Mensagem de erro para preço negativo não documentada no Swagger	01-140 CT-019 - Cadastro de produtos com preço negativo	2 test executions Label: 30/May/2025 - Passed	1 bug found 0% resolved → 100% open → 1 new
01-174 CT-020 Mensagem de erro com quantidade negativa não documentada no Swagger	01-142 CT-020 - Cadastro de produtos com quantidade negativa	2 test executions Label: 30/May/2025 - Passed	1 bug found 0% resolved → 100% open → 1 new
01-175 CT-024 - Listagem de produtos sem autenticação permitida pela API	01-150 CT-024 - Listagem de produtos sem autenticação	2 test executions Label: 30/May/2025 - Failed	1 bug found 0% resolved → 100% open → 1 new

Report de bugs realizados com teste manual via postman

Teste Manual via Postman- Cenarios Mapeados



Teste Altomatizado via Robot Framework

Tests Log							Generated 20250605 20:08:16 UTC-03:00 16 hours 28 minutes ago
Test Statistics							
Total Statistics							
All Tests	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
	19	19	0	0	00:10:11		
Statistics by Tag							
Deleteusuario	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
Getprodutos	1	1	0	0	00:00:00		
Getusuario	1	1	0	0	00:00:00		
Postlogin	1	1	0	0	00:00:01		
Postlogininvalido	1	1	0	0	00:00:00		
Postloginsenhainvalida	1	1	0	0	00:00:00		
Postproduto	1	1	0	0	00:00:01		
Postusuario	1	1	0	0	00:00:00		
Putproduto	1	1	0	0	00:00:00		
Putusuario	1	1	0	0	00:00:00		
tokenexpirado	1	1	0	0	00:10:01		
Statistics by Suite							
Tests	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
Tests_Carrinho Tests	19	19	0	0	00:10:12		
Tests_Login Tests	6	6	0	0	00:00:06		
Tests_Produtos Tests	4	4	0	0	00:10:02		
Tests_Usuarios Tests	5	5	0	0	00:00:03		
	4	4	0	0	00:00:01		
Test Execution Log							

Teste automatizado realizado via Robot Framework

-	BOUTE	Produtos Tests		00:00:03.049
-		Full Name:	Tests.Produtos.Tests	
-		Documentation:	Análise de testes para Endpoint/produtos	
-		Source:	c:\Dcs-Compass\lojaapi_servernet_testes_automatizado\testes\produtos_tests.robot	
-		Start / End / Elapsed:	20250605 20:08:12.068 / 20250605 20:08:15.117 / 00:00:03.049	
-		Status:	5 tests total, 5 passed, 0 failed, 0 skipped	
-		+ SETUP -- Iniciar Sessão		00:00:00.052
-	+ TEST	CT.01F-Cadastro de produto válido		00:00:00.752
-	+ TEST	CT.G2I -- Atualização de produto válido		00:00:00.196
-	+ TEST	CT.G2J -- Listagem de produtos válidos		00:00:00.405
-	+ TEST	CT.G2S -- Exclução de produto válido		00:00:00.735
-	+ TEST	CT.G2K -- Exclução de produto vinculado a carrinho		00:00:00.532
-		-		00:00:00.000
-	BOUTE	Usuarios Tests		00:00:01.330
-		Full Name:	Tests Usuarios Tests	
-		Documentation:	Análise de testes para Endpoint/usuarios	
-		Source:	c:\Dcs-Compass\lojaapi_servernet_testes_automatizado\testes\usuarios_tests.robot	
-		Start / End / Elapsed:	20250605 20:08:15.117 / 20250605 20:08:16.447 / 00:00:01.330	
-		Status:	4 tests total, 4 passed, 0 failed, 0 skipped	
-		+ SETUP -- Iniciar Sessão		00:00:00.001
-	+ TEST	CT.00I: Cadastro de usuário válido		00:00:00.371
-	+ TEST	CT.00K Alterar usuário válido		00:00:00.182
-	+ TEST	CT.010: Listagem de usuários válida		00:00:00.465
-	+ TEST	CT.01Z Exclução de usuário válido		00:00:00.364

```
MINGW64~/Users/RASTOFF/Desktop/EC2-AWS - [x] x
teste o funcionamento acessando http://localhost:3000/usuarios no navegador

para saber as rotas disponíveis e como utilizá-las? Acesse http://localhost:3000 para alterar porta de execução, timeout do token, etc? Execute npm server=help para outras coisas execute github.com/serverless-testes

feito com * para todos os GETs e apoiado por Leandro Melo

POST /usuarios 200 76.818 ms - 82
POST /login 200 7.751 ms - 284
POST /produtos 200 21.155 ms - 82
POST /carritos 200 30.190 ms - 82
POST /usuarios 200 9.151 ms - 82
POST /login 200 0.775 ms - 288
POST /produtos 200 7.821 ms - 82
POST /carritos 400 1.827 ms - 149
POST /usuarios 200 9.570 ms - 82
POST /login 200 0.750 ms - 288
POST /carritos 200 5.360 ms - 82
GET /carritos 200 0.945 ms - 1355
POST /usuarios 200 1.674 ms - 82
POST /login 200 1.545 ms - 280
POST /produtos 200 11.906 ms - 82
POST /carritos 200 6.896 ms - 82
DELETE /carritos/concluir compra 200 2.438 ms - 51
POST /usuarios 200 1.445 ms - 82
POST /login 200 0.585 ms - 284
POST /produtos 200 7.400 ms - 82
POST /carritos 200 6.886 ms - 82
DELETE /carritos/concluir compra 200 0.979 ms - 45
ET /carritos/3foqtsdglgkelyk 400 0.539 ms - 45
POST /usuarios 200 12.093 ms - 82
POST /login 200 1.837 ms - 282
POST /produtos 200 16.490 ms - 82
POST /carritos 200 13.025 ms - 82
DELETE /carritos/cancelar compra 200 10.512 ms - 86
POST /usuarios 200 0.699 ms - 82
POST /login 200 0.718 ms - 320
POST /login 400 1.314 ms - 48
POST /login 400 2.746 ms - 48
POST /usuarios 200 10.154 ms - 82
POST /login 200 1.655 ms - 289
POST /produtos 400 1.374 ms - 103
POST /usuarios 200 15.216 ms - 82
POST /login 200 1.778 ms - 279
POST /produtos 200 4.540 ms - 82
PUT /produtos/djBnCwQ0IaITLH 200 19.374 ms - 50
POST /usuarios 200 3.212 ms - 82
ET /produtos 200 0.951 ms - 1688
POST /usuarios 200 5.854 ms - 82
POST /login 200 2.841 ms - 261
POST /produtos 200 7.472 ms - 82
DELETE /produtos/PoAFNcG03iaShS 200 1.430 ms - 51
POST /usuarios 200 0.624 ms - 82
POST /login 200 0.725 ms - 287
POST /produtos 200 2.137 ms - 82
POST /carritos 200 1.735 ms - 82
DELETE /produtos/mCvYedRshpUby 400 0.715 ms - 131
POST /usuarios 200 1.021 ms - 82
PUT /usuarios/kBFcdtCB7zgttlK 200 1.318 ms - 50
POST /usuarios 200 8.024 ms - 82
ET /usuarios 200 0.820 ms - 3304
POST /login 200 4.030 ms - 82
DELETE /usuarios/kBFcdtCB7zgttlK 200 5.106 ms - 51
[{"id": "ec2-user@ip-172-31-81-15 servers[api]", exit: 0}
Connection to ec2-44-202-51-7.compute-1.amazonaws.com closed.

MINGW64~/Users/RASTOFF/Desktop/EC2-AWS (main)
BLAST OFF DESKTOP-10B9NP6 MINGW64 ~/Desktop/EC2-AWS (main)
```

Os testes demonstraram que a API atende aos requisitos principais em sua maioria, com estabilidade nos fluxos automatizados. Entretanto, falhas críticas foram identificadas nos testes manuais, como falta de autenticação para recursos sensíveis, e ausência de validação de regras de negócio básicas como formato de e-mail e limites de senha. Essas falhas

indicam riscos à segurança e à integridade do sistema e precisam ser tratadas com prioridade.

## Implicações para negócios [↗](#)

As falhas encontradas podem impactar diretamente a confiança dos usuários na API, além de representar riscos de segurança e de não conformidade com práticas recomendadas. A exposição de dados sem autenticação e a inconsistência entre a documentação e o comportamento real da API afetam negativamente a manutenção, integração com terceiros e o ciclo de vida do produto.

## Recomendações [↗](#)

- Corrigir imediatamente os problemas de segurança relacionados à autenticação.
- Implementar validações consistentes para entrada de dados conforme regras de negócio.
- Atualizar documentação Swagger para refletir o comportamento real da API.
- Expandir a cobertura de testes automatizados para incluir cenários negativos e de exceção.
- Realizar revisões regulares do plano de testes conforme o sistema evolui.

## Conclusão [↗](#)

O projeto apresentou bons resultados nos testes automatizados e cobriu integralmente os casos manuais planejados. No entanto, as falhas críticas encontradas reforçam a necessidade de melhorias estruturais na validação de dados e nos controles de segurança da API. Com a adoção das recomendações propostas, a aplicação estará mais alinhada com as boas práticas de qualidade e segurança exigidas em ambientes de produção.