


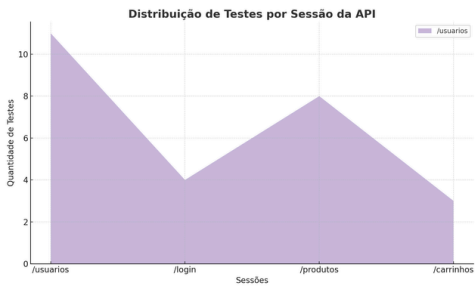
Relatório Analítico de Testes > ServeRest

Informações Gerais

Item	Descrição
Projeto	Serverest
Período de Execução	2 de jun. de 2025 – 6 de jun. de 2025
URL / Acesso	 ServeRest
Sistema Operacional	Windows 10
Responsável	Karen K.

Resumo Geral

- **Total de Casos de Teste Executados:** 27
- **Casos de Teste Aprovados:** 24
- **Casos de Teste Reprovados:** 3
- **Cobertura Funcional Estimada:** 100% das funcionalidades principais



Rotas Testadas:

- POST /usuarios
- POST /login
- POST /produtos
- POST /carrinhos

Falhas Encontradas:

- 3 falhas de validação de regras de negócio
- Todas as falhas permitiram cadastros indevidos no sistema, o que compromete a integridade da base de dados

Interpretação dos Resultados

Embora a maior parte dos testes tenha sido aprovada, **três falhas importantes foram identificadas** no endpoint de criação de usuários, todas relacionadas a **validações críticas de entrada** que afetam diretamente a qualidade dos dados e o cumprimento das regras de negócio esperadas.

Essas falhas permitem o cadastro de usuários com:

- Domínios de e-mail inválidos (como Gmail ou Hotmail)
- Senhas abaixo do mínimo aceitável
- Senhas acima do limite máximo permitido

Esses comportamentos, mesmo que não causem falhas técnicas visíveis de imediato, podem levar a:

- Riscos de segurança
- Dificuldade de rastreabilidade de usuários válidos
- Violação de políticas internas da aplicação

Falhas Identificadas x Impacto [↗](#)

✖ TC002 - Cadastro com e-mail de domínio público (gmail/hotmail)



- **Descrição:** Cadastro com e-mails de domínio público, como Gmail e Hotmail.
- **Resultado Esperado:** Rejeição com `400 Bad Request`
- **Resultado Obtido:** `201 Created` – Usuário foi cadastrado
- **Impacto: Alta**
- **Risco:** A aplicação não filtra domínios de e-mail proibidos, o que pode violar políticas de negócio (como uso exclusivo de domínios corporativos).
- **Recomendação:** Implementar validação de domínio no backend com feedback adequado ao usuário.

✖ TC008 - Cadastro com senha inferior a 5 caracteres



- **Descrição:** Cadastro com senha com apenas 3 caracteres.
- **Resultado Esperado:** Rejeição com `400 Bad Request`
- **Resultado Obtido:** `201 Created` – Usuário foi cadastrado
- **Impacto: Crítico**
- **Risco:** Reduz significativamente a segurança da plataforma ao permitir senhas fracas, aumentando o risco de ataques por força bruta.
- **Recomendação:** Implementar validação de tamanho mínimo no backend com retorno informativo.

✖ TC009 - Cadastro com senha superior a 10 caracteres



- **Descrição:** Cadastro com senha de 11 caracteres.
- **Resultado Esperado:** Rejeição com `400 Bad Request`
- **Resultado Obtido:** `201 Created` – Usuário foi cadastrado
- **Impacto: Médio**
- **Risco:** A falha não compromete diretamente a segurança, mas gera inconsistência com a documentação da API e possíveis falhas de UX em sistemas integrados. Se houver uma justificativa de segurança (como controle de buffer ou performance), o impacto deve ser reclassificado como alto e de segurança.
- **Recomendação:** Corrigir o limite no back-end, validando corretamente o tamanho máximo da senha e garantindo coerência com a documentação e qualquer limitação técnica esperada.

Indicadores de Qualidade [↗](#)

Indicador	Valor
Casos de teste executados	27
Casos aprovados	24
Casos reprovados	3
Sucesso geral dos testes	88,9%
Regras de negócio validadas	Sim
Módulos com falhas	/usuarios
Módulos estáveis	/login, /produtos, /carrinhos
Falhas com impacto no negócio	3

Conclusões e Ações Recomendadas [↗](#)

- A aplicação **ServeRest** apresenta **estabilidade geral nas rotas de login, produtos e carrinhos**, com comportamento consistente e adequado.
- **Três falhas graves foram encontradas na rota /usuarios**, todas ligadas à ausência de validações básicas de segurança e integridade.
- As falhas representam **risco real de vazamento de dados, falhas de segurança e inconsistência com a documentação**.
- Recomenda-se:
 - Correção imediata das validações nos campos de e-mail e senha;
 - Atualização da documentação caso novas regras sejam adotadas;
 - Nova rodada de testes de regressão após os ajustes;
 - Monitoramento da rota com testes automatizados periódicos.

Anexos e Evidências [↗](#)

Bugs registrados no Jira:

[ServeRest \(Tests API\) — Backlog - Jira](#)

Plano de Testes:

[📄 Challenge 02 - Plano de Testes x Relatório \(ServeRest\)](#)



TC002



TC008



TC009