

# Plano de testes

Ruptura ( @Luis Magris de Sousa @Cassia Yumi @Raique Alfredo Pereira de Ramos @Thais Nogueira @Ádony Lagares )

## ServeRest

versão 1.0

## Histórico das alterações

Data	Versão	Descrição	Autor(es)
04/04/2025	1.0	Release inicial	Ruptura ( @Cassia Yumi @Luis Magris de Sousa @Raique Alfredo Pereira de Ramos @Thais Nogueira @Ádony Lagares )

### 1. Introdução

Este plano de teste descreve a abordagem para testar a aplicação web ServeRest, que tem como objetivo permitir que os usuários se cadastrem, realizem login e selecionem produtos para serem adicionados ao carrinho. Este documento descreve os requisitos a testar, os tipos de testes definidos para cada iteração, os recursos de hardware e software a serem empregados e o cronograma dos testes. O objetivo deste plano de teste é garantir a qualidade e confiabilidade do software antes do lançamento.

### 2. Objetivos do Teste

Os objetivos do teste são:

- Verificar se todas as funcionalidades do sistema estão implementadas corretamente.
- Validar se o sistema atende aos requisitos funcionais e não funcionais especificados.
- Identificar e corrigir defeitos encontrados durante o teste.

### 3. Escopo

O teste abrangerá todas as funcionalidades principais da aplicação e requisitos não funcionais, incluindo:

### Casos de uso:

Identificador do caso de uso	Nome do caso de uso
id UC1	Cadastrar usuário
id UC2	Login
id UC3	Adicionar produto ao carrinho de compras
id UC4	Pesquisar produtos
id UC5	Listar produtos

id UC6	Visualizar detalhes do produto
id UC7	Logout

#### Requisitos não-funcionais: [🔗](#)

Identificador do requisito	Nome do requisito
id req1	Usabilidade
id req2	Segurança
id req3	Testabilidade
id req4	Manutenibilidade
id req5	Desempenho

#### 4. Estratégia de Teste

A estratégia de teste incluirá:

- Testes unitários realizados pelos desenvolvedores para verificar as funcionalidades individualmente.
- Testes de integração para garantir que as diferentes partes do sistema funcionem em conjunto.
- Testes de sistema para verificar o funcionamento do sistema como um todo.
- Testes de performance para avaliar o tempo de resposta da aplicação.
- Testes de carga para verificar como a aplicação se comporta sob diferentes volumes de acesso simultâneo.
- Testes de regressão para verificar se alterações realizadas não prejudicam o software.
- Teste de aceitação pelo usuário para validar a usabilidade.

#### 5. Casos de Teste

Serão criados casos de teste para cada funcionalidade do sistema, abrangendo cenários positivos e negativos, bem como casos de teste de limite e estresse.

Caso de Teste 1	Cadastrar usuário			
Objetivo	Verificar se é possível criar um novo usuário			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema ( )	Unidade (x)	Aceitação ( )
Responsável(is)	Equipe de testes			

Caso de Teste 2	Login			
Objetivo	Verificar se é possível logar na aplicação			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema ( )	Unidade (x)	Aceitação ( )

Responsável(is)	Equipe de testes
-----------------	------------------

Caso de Teste 3	Adicionar produto ao carrinho de compras			
Objetivo	Verificar se o usuário consegue adicionar um produto válido ao carrinho com sucesso			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema (x )	Unidade ( )	Aceitação ( )
Responsável(is)	Equipe de testes			

Caso de Teste 4	Pesquisar produtos			
Objetivo	Verificar se é possível pesquisar e listar os produtos.			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema ( )	Unidade (x)	Aceitação ( )
Responsável(is)	Equipe de testes			

Caso de Teste 5	Listar produtos			
Objetivo	Listar todos os produtos cadastrados na loja,			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema (x)	Unidade ( )	Aceitação ( )
Responsável(is)	Equipe de testes			

Caso de Teste 6	Visualizar detalhes do produto			
Objetivo	Garantir que os detalhes de um produto específico sejam exibidos corretamente			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema (x)	Unidade ( )	Aceitação ( )
Responsável(is)	Equipe de testes			

Caso de Teste 7	Logout			
Objetivo	Verificar se é possível o usuário sair da sessão.			
Técnica:	(x) manual		(x) automática	
Estágio do teste	Integração ( )	Sistema ( )	Unidade (x)	Aceitação ( )
Responsável(is)	Equipe de testes			

## 6. Ambiente de Teste

O teste será realizado em um ambiente de teste dedicado, replicando o ambiente de produção o mais próximo possível.

## 7. Recursos

- Equipe de Teste: 5 testadores.
- Ambiente de Teste: Servidor dedicado com configuração similar ao ambiente de produção.
- Dados de Teste: Conjunto de dados de teste representativos serão criados para simular cenários reais.
- Conexão com a internet.
- Ferramentas de teste.

## 17 Cronograma

O teste será realizado ao longo de 2 semanas, conforme o seguinte cronograma:

Tipo de teste	Duração	data de início	data de término
Planejar teste		dd/mm/aaaa	dd/mm/aaaa
Projetar teste		dd/mm/aaaa	dd/mm/aaaa
Implementar teste		dd/mm/aaaa	dd/mm/aaaa
Executar teste		dd/mm/aaaa	dd/mm/aaaa
Avaliar teste		dd/mm/aaaa	dd/mm/aaaa

## 9. Critérios de Aceitação

O sistema será considerado aceito quando:

- Todos os casos de teste forem executados e passarem com sucesso.
- Todos os defeitos críticos e importantes forem corrigidos e verificados.
- A **interface for validada por usuários reais** no teste de aceitação.

## 10. Riscos

- Atrasos no desenvolvimento podem afetar o cronograma de teste.
- Possíveis problemas de compatibilidade com navegadores e dispositivos.

## 11. Responsabilidades

- A equipe de desenvolvimento será responsável por corrigir defeitos encontrados durante o teste.
- A equipe de teste será responsável por executar os casos de teste e relatar os resultados.

## 12. Comunicação

Relatórios de teste serão gerados regularmente e compartilhados com a equipe de desenvolvimento e gerência para acompanhamento e tomada de decisões.

### ✅ 13. Aprovação

Este plano de teste será revisado e aprovado pela equipe de desenvolvimento antes do início do teste. Quaisquer alterações subsequentes serão comunicadas e aprovadas pelas partes interessadas relevantes.

### 💡 14. Considerações Finais

Este plano de teste é um guia para o processo de teste da aplicação Severest. O objetivo é garantir que o software atenda aos requisitos e seja lançado com alta qualidade e confiabilidade.