

**Plano de Teste**

**Grupo 10: Techseller**

**Integrantes:**

Davi da Silva Patricio

Elivelton Sampaio

Kauan Martins

Luanderson Amparo Meira

Luiz Felipe Cardoso

**Instituição:** Centro Universitário Senac

São Paulo 2025

**Sumário**

Página 3 ....................................... Histórico de revisões

Página 3 ........................................ 1 Escopo do projeto de Software

Página 3 ........................................ 1.2 Objetivo do projeto

Página 4 ........................................ 1.3 Metodologia de desenvolvimento

Página 4 ........................................ 1.3.1 Processos de desenvolvimento

Página 4 ........................................ 1.3.2 Processos de teste

Página 5 ........................................ 2 Matriz de rastreabilidade de requisitos do sistema

Página 5 .................................. 2.1 Requisitos funcionais

Página 5 .................................. 2.2 Requisitos não funcionais

Página 6 .................................. 3 Estratégias de teste

Página 7 .................................. 4 Ambiente de teste

Página 7 .................................. 5 Matriz de Responsabilidade

Página 7 .................................. 6. Relação dos casos de teste

Página 8 .................................. 6.1 Descrição dos principais casos de teste

Página 10 ................................ 7 Riscos do projeto de teste de Software

Página 11 ................................ 8 Critérios de conclusão

Página 12 ................................ Referências

**Histórico de Revisões:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Autor** | **Descrição** |
| 1.0 | 05/05/2025 | Luanderson | implementações de teste no projeto criando o diretório de testes e aplicação de testes de caixa preta |
| 2.0 (final) | 17/05/2025 | Davi | Verão final do projeto com todos os testes do plado de teste e relatório dos testes |

1. **ESCOPO DO PROJETO DE SOFTWARE**
   1. **Escopo Geral**

Garantir que o sistemaTechSeller, desenvolvido em Java com Spring Boot, esteja funcional, seguro e usável, validando requisitos funcionais e não funcionais por meio de testes planejados, executados e documentados.

**1.2 Objetivo do Projeto**

O objetivo deste documento é definir a abordagem, os tipos de testes e os critérios adotados para verificar e validar os requisitos funcionais e não funcionais do sistema TechSeller.

Este plano contempla a execução de testes funcionais (caixa preta), testes unitários com JUnit, testes de integração via Insomnia, e testes de requisitos não funcionais, como desempenho (JMeter), usabilidade e segurança.

A intenção é assegurar que os principais módulos do sistema — como cadastro de cliente, login, carrinho de compras e finalização de pedidos — estejam funcionando corretamente, conforme os requisitos estabelecidos, e que o sistema esteja estável, responsivo e seguro para os usuários finais.1.2 Metodologia de Desenvolvimento

Foi utilizada a metodologia ágil com enfoque em sprints, integração contínua e revisões iterativas. O projeto é estruturado em Java com Spring Boot (MVC), persistência com Spring Data JPA e banco relacional. O controle de versão é feito com Git.

**1.3 Metodologia de Desenvolvimento**

O projeto TechSeller adotou a **metodologia ágil**, com foco em ciclos iterativos e entregas incrementais, permitindo o ajuste contínuo de funcionalidades e testes ao longo do desenvolvimento. A equipe se organizou em **sprints semanais**, nas quais os requisitos eram analisados, implementados e testados progressivamente.

**1.3.1 Processos de Desenvolvimento:**

**Framework Utilizado:** Java 20 com Spring Boot (arquitetura MVC), com persistência via Spring Data JPA.

**Banco de Dados:** PostgreSQL (produção) e H2DB (ambiente de teste local).

**Controle de Versão:** Git, com repositório centralizado e branches por funcionalidade.

**Integração Contínua (CI):** Compilação, build e execução de testes automatizados configurados a cada entrega.

**Gerenciamento de Tarefas:** Distribuição de papéis e responsabilidades conforme matriz de responsabilidade.

**1.3.2 Processos de Testes**

**Testes Unitários**: Escritos com JUnit 5, focados na lógica de negócio das classes de serviço (ex: UserService, ProdutoService). Executados localmente e integrados à pipeline de build.

**Testes de Integração**: Realizados com a ferramenta *Insomnia*, visando validar a comunicação entre o frontend e os endpoints RESTful da aplicação.

**Testes Funcionais (Caixa Preta)**: Conduzidos manualmente com base em cenários de uso reais, verificando a interface e os fluxos completos de navegação.

**Testes de Requisitos Não Funcionais**:

* *Desempenho*: Aplicado com JMeter simulando diferentes cargas de usuários simultâneos.
* *Usabilidade*: Avaliação heurística baseada em princípios de Nielsen.
* *Segurança*: Simulações de acesso indevido para verificar controles de autenticação e sessões.

**2. MATRIZ DE RASTREABILIDADE DE REQUISITOS DE SISTEMA**

**2.1 Requisitos Funcionais (RF)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Nome** | **Descrição resumida** |  |  |  |  |  |
| RF02 | Atualizar item do carrinho | POST / atualizar |  |  |  |  |  |
| RF03 | Remover Item do Carrinho | POST / remover |  |  |  |  |  |
| RF04 | Calcular frete | POST / frete |  |  |  |  |  |
| RF05 | Finalizar pedido | GET / finalizar |  |  |  |  |  |
| RF06 | Confirmar pedido | POST / confirmar-pedido |  |  |  |  |  |
| RF08 | Cadastrar cliente | POST / cadastrar |  |  |  |  |  |
| RF20 | Login de usuário | GET / login |  |  |  |  |  |

**2.2 Requisitos Não Funcionais (RNF)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Categoria | Característica | Descrição | **Ferramenta de teste utilizada** |
| RNF01 | Desempenho | Tempo de resposta | O sistema deve responder em menos de 2 segundos nas requisições principais. | |  | | --- | | Apache JMeter |  |  | | --- | |  | |
| RNF1.1 | Desempenho | Escalabilidade | O sistema deve manter a estabilidade sob carga de até 100 usuários simultâneos. | |  | | --- | | Apache JMeter |  |  | | --- | |  | |
| RNF1.2 | Desempenho | Vazão | Vazão média de 16,8 requisições/segundo com 100 usuários. | |  | | --- | | Apache JMeter |  |  | | --- | |  | |
| RNF02 | Usabilidade | Interface | A interface deve ser intuitiva, responsiva e compatível com desktop e mobile. | |  | | --- | | Avaliação heurística informal |  |  | | --- | |  | |
| RNF2.1 | Usabilidade | Navegabilidade | Botões identificáveis, fluxo claro de navegação. | |  | | --- | | Testes manuais (Desktop) |  |  | | --- | |  | |
| RNF03 | Segurança | Autenticação | Acesso a recursos protegidos requer login com senha criptografada. | |  | | --- | | Testes manuais de sessão |  |  | | --- | |  | |
| RNF3.1 | Segurança | Controle de acesso | Tentativas de acesso sem login devem redirecionar à tela de login. | Testes de redirecionamento |

**3. ESTRATÉGIA DE TESTE**

**Níveis de Teste**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Nome do teste** | **Técnica** | **Requisitos** |
| 01 | Teste Unitário | JUnit | RF02 - RF06- RF08 -RF20 |
| 02 | Teste de Integração | Imnsonia | RF04 - RF05 |
| 03 | Teste de sistema | Manual | RF02- RF06 - RF08 - RF20 |
| 04 | Teste de aceitação | Caixa preta | RF02 - RF03 -RF04 - RF05-RF06 - RF08 - RF20 |

**Tipos de Testes**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tipo** | **Técnica** | **Descrição** |
| 01 | Desempenho | JMeter | Testar tempo de resposta de carregamento |
| 02 | Usabilidade | Avaliação | Interface deve ser responsiva e simples |
| 03 | Segurança | Manual | Testes de autenticação e sessão |

**4. AMBIENTE DE TESTE**

|  |  |
| --- | --- |
| **Recursos** | **Descrição** |
| Computador Cliente | Windows 10/11, navegador Chrome |
| Servidor de aplicação | Spring Boot local |
| Banco de dados | PostgreSQL local |
| Ferramenta de teste | JUnit, JMeter |
| Ferramenta de Integração | Imnsonia |

**5. MATRIZ DE RESPONSABILIDADE**

|  |  |  |  |
| --- | --- | --- | --- |
| **fase** | **Papel** | **Responsável** | **Atividade** |
| Planejamento | líder de testes | Luanderson Meira | Coordenar o plano e revisões |
| Especificação | Analista de testes | Elivelton Sampaio | Criar casos de testes e prepara dados |
| Execução | Testador | Davi Patricio, Kauan Martins | Executa os testes |
| Resultados | Avaliador de resultados | Luiz Felipe | Documenta e interpreta os resultados |

**6. RELAÇÃO DOS CASOS DE TESTES**

|  |  |  |
| --- | --- | --- |
| **CT** | **Cenário** | **Descrição** |
| CT01 | Cadastro de cliente | |  |  |  | | --- | --- | --- | | |  | | --- | | Valida o cadastro com dados válidos e redirecionamento para login. Inclui teste de integração via Insomnia. |  |  | | --- | |  | |  |  | | --- | |  | |
| CT02 | Login | |  | | --- | | Verifica login com credenciais corretas e tratamento de erro com credenciais inválidas. Inclui teste de integração via Insomnia. |  |  | | --- | |  | |
| CT03 | Carrinho | |  | | --- | | Valida adição e atualização de produto no carrinho |  |  | | --- | |  | |
| CT04 | Remover item | |  | | --- | | Verifica remoção e ajuste de subtotal do carrinho |  |  | | --- | |  | |
| CT05 | Calcular frete | |  | | --- | | Valida cálculo para diferentes CEPs e tratamento de erro |  |  | | --- | |  | |
| CT06 | Finalizar pedido | |  | | --- | | Valida redirecionamento e exibição de endereço/pagamento |  |  | | --- | |  | |
| CT07 | Confirmar pedido | Verifica conclusão e registro do pedido no sistema |

**6.1 Descrição dos principais cenários de teste:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Cenário de Teste** | **Objetivo** | **Requisitos Funcionais** |
| CT01 | |  | | --- | | Cadastro de Cliente |  |  | | --- | |  | | |  | | --- | | Verificar se o sistema permite o cadastro com dados válidos e redireciona para o login com sucesso. |  |  | | --- | |  | | RF08 |
| CT02 | Login correto e incorreto | |  | | --- | | Validar login com credenciais corretas e a exibição de mensagens de erro com dados inválidos. |  |  | | --- | |  | | RF20 |
| CT03 | Atualizar item no carrinho | |  | | --- | | Garantir que o sistema atualiza corretamente a quantidade e o subtotal dos produtos no carrinho. |  |  | | --- | |  | | RF02 |
| CT04 | Remover item do carrinho | |  | | --- | | Testar a remoção de produtos do carrinho e o ajuste do subtotal. |  |  | | --- | |  | | RF03 |
| CT05 | Calcular frete com CEP | |  | | --- | | Verificar o cálculo correto do frete para diferentes CEPs e o tratamento de CEP inválido. |  |  | | --- | |  | | RF04 |
| CT06 | Finalizar pedido | |  | | --- | | Testar o redirecionamento para a tela de finalização com resumo de pedido, endereço e pagamento. |  |  | | --- | |  | | RF05 |
| CT07 | Confirmar pedido | |  | | --- | | Validar o registro final do pedido e redirecionamento para tela de confirmação com status correto. |  |  | | --- | |  | | RF06 |

**7. RISCOS DO PROJETO DE TESTE DE SOFTWARE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Risco** | **Descrição** | **Grau** | **Resposta planejada** |
| R01 | Falha no ambiente | |  | | --- | | Indisponibilidade do ambiente local para testes. |  |  | | --- | |  | | Médio | |  | | --- | | Uso de backup remoto para continuidade dos testes. |  |  | | --- | |  | |
| R02 | Mudança de requisito | |  | | --- | | Requisitos alterados após escrita ou execução dos testes. |  |  | | --- | |  | | Alto | |  | | --- | | Replanejamento e revisão dos testes afetados. |  |  | | --- | |  | |
| R03 | Baixa cobertura de exceções | |  | | --- | | Falta de testes para entradas inválidas ou fluxos alternativos. |  |  | | --- | |  | | Médio | |  | | --- | | Incluir cenários de exceção em sprints futuras. |  |  | | --- | |  | |
| R04 | Integração com APIs falhando | |  | | --- | | Dependência de APIs externas ou endpoints instáveis durante testes. |  |  | | --- | |  | | Médio | |  | | --- | | Testes isolados e mocks para simulações. |  |  | | --- | |  | |
| R05 | Limitações de tempo | |  | | --- | | Prazos curtos podem comprometer a execução completa dos testes. |  |  | | --- | |  | | Alto | Priorização de testes críticos e automatizados. |

**8. CRITÉRIOS DE CONCLUSÃO**

|  |  |
| --- | --- |
| **Critério** | **Descrição** |
| Execução | 100% dos casos de teste planejados (CT01 a CT07) devem ser executados pelo menos uma vez. |
| Resultado esperado | Todos os testes (funcionais, unitários, integração) devem ser concluídos sem erros críticos. |
| Conformidade | O sistema deve atender a todos os requisitos funcionais (RF) e não funcionais (RNF) especificados no plano de teste. |
| Evidências | Todas as evidências de execução (prints, logs, tabelas de decisão) devem estar documentadas e armazenadas no repositório oficial do projeto. |
| Estabilidade do Sistema | O sistema deve apresentar comportamento estável e responsivo durante os testes, inclusive sob carga simulada. |

**9. CRONOGRAMA DO PROJETO DE TESTE DE SOFTWARE**

|  |  |
| --- | --- |
| Semana | Atividade |
| 1 | Planejamento e levantamento de RF |
| 2 | Escrita dos casos de teste |
| 3 | Execução de testes unitários e API |
| 4 | Execução de testes manuais e ajuste |
| 5 | Consolidação e entrega dos documentos |

**REFERÊNCIAS**

* PRESSMAN, Roger S.; MAXIM, Bruce R. *Engenharia de Software: Uma Abordagem Profissional*. 8ª ed. McGraw-Hill, 2016.
* DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. *Introdução ao Teste de Software*. Elsevier, 2007.
* SOMMERVILLE, Ian. *Engenharia de Software*. 9ª ed. Pearson, 2011.