

*Registro de Mudanças*

| **Versão** | **Data de Mudança** | **Por** | **Descrição** |
| --- | --- | --- | --- |
| número da versão | Data de Mudança | Nome da pessoa que fez as mudanças | Descrição das mudanças feitas |
|  |  |  |  |
|  |  |  |  |

**1** **INTRODUÇÃO 2**

1.1 Escopo 2

*1.1.1* *No escopo 2*

*1.1.2* *Fora do escopo 2*

1.2 Objetivos de Qualidade 2

1.3 Papéis e Responsabilidades 2

**2** **METODOLOGIA DE TESTE 3**

2.1 Visão Geral 3

2.2 Fases de Teste 3

2.3 Triagem de Erros 3

2.4 Critérios de Suspensão e Requisitos de Retomada 3

2.5 Completude do Teste 3

2.6 Atividades do projeto, estimativas e cronograma 4

**3** **ENTREGÁVEIS DE TESTE 4**

**4** **NECESSIDADES DE RECURSOS E AMBIENTE 4**

4.1 Ferramentas de Teste 4

4.2 Ambiente de Teste 4

**5** **TERMOS / ACRÔNIMOS 5**

1. **Introdução**

O Plano de Teste é projetado para descrever o escopo, abordagem, recursos e cronograma de todas as

atividades de teste do projeto Biblioteca Comunitária.

O plano identifica os itens a serem testados, os recursos a serem testados, os tipos de teste a serem

realizados, o pessoal responsável pelo teste, os recursos e o cronograma necessários para concluir o

teste e os riscos associados ao plano.

* 1. **Escopo**
     1. No escopo

O escopo dos testes para este projeto se concentrará na validação dos seguintes módulos e funcionalidades, através de testes manuais e unitários:

Testes Funcionais/Manuais:  
  
QT-001: Apenas o título  
QT-002: Apenas o título

QT-003: Apenas o título  
QT-004: Apenas o título  
  
Testes Unitários:

Descrever teste unitario 1  
Descrever teste unitario 2

Exemplo: Teste da classe/método responsável pela **lógica de negócio de realizar uma devolução**, focando na atualização de status do livro

* + 1. Fora do escopo

**Testes de Segurança:** Não serão realizados testes para identificar vulnerabilidades de segurança na aplicação.

* 1. **Objetivos de Qualidade**
* Garantir que o software sob teste (SUT) esteja em conformidade com os requisitos funcionais de Empréstimo, Devolução e Gestão de Unidades.
* Identificar e corrigir defeitos/problemas críticos relacionados à persistência de dados (leitura e escrita nos CSVs) antes de entrar em operação.
* Garantir uma cobertura de código razoável nas classes de lógica de negócio.
* Verificar que as exceções (ex: FileNotFoundException, NullPointerException) são tratadas de forma elegante, sem travar a aplicação

* 1. **Papéis e Responsabilidades**

Neste projeto, a equipe de teste é composta pelos próprios desenvolvedores. Os papéis foram distribuídos de forma colaborativa, onde todos os membros atuam como Analistas de Teste e Qualidade.  
  
**Papel**: Analista de Teste / Desenvolvedor de Testes  
  
**Responsável(eis)**: Thiago Melo, Bruno Gabriel, Eduardo Barros, Luigi Bongiolo e Thiago Pereira  
  
**Principais Responsabilidades:** Planejar e projetar os casos de teste (manuais e unitários). - Desenvolver os scripts de teste unitário utilizando JUnit. - Executar os casos de teste manuais. - Registrar e documentar os defeitos encontrados. - Preencher e manter este Plano de Teste.

1. **Metodologia de Teste** 
   1. **Visão Geral**

Mencionar o motivo da adoção de uma determinada metodologia de teste método para o projeto. A metodologia de teste selecionada para o projeto pode ser

* Cascata
* Iterativo
* Ágil
* RUP

* 1. **Fases de Teste**

**As fases de teste definem os tipos de teste a serem executados no software sob teste (SUT**). As fases de testes dependem do escopo do projeto, restrições de tempo e orçamento.

* 1. **Triagem de Erros**

O objetivo da triagem é

* Definir o tipo de resolução para cada erro
* Priorizar os erros e determinar um cronograma para todos os “Erros a serem corrigidos”.

* 1. **Critérios de Suspensão e Requisitos de Retomada**
* **Critérios de Suspensão:** A execução dos testes será suspensa se um defeito de prioridade **Crítica** for encontrado que bloqueie a execução de múltiplos outros casos de teste.
* **Requisitos de Retomada:** A execução dos testes será retomada assim que o defeito bloqueante for corrigido pela equipe e uma nova versão estável do software for disponibilizada.
  1. **Completude do Teste**

Os testes da "Entrega 1" serão considerados completos quando os seguintes critérios forem atendidos:

* 100% dos casos de teste manuais definidos no escopo foram executados.
* 100% dos testes unitários planejados foram implementados e estão passando com sucesso (status "verde").
* Nenhum defeito de prioridade **Crítica** ou **Alto** permanece sem correção.

* 1. **Atividades do projeto, estimativas e cronograma**

Descreva as atividades de teste a serem realizadas no projeto, suas estimativas e cronograma

1. **Entregáveis de Teste**

Aqui estão alguns exemplos de resultados

| * Plano de teste * Casos de teste * Matriz de rastreabilidade de requisitos * Relatório de erros * Estratégias de Teste * Métricas de Teste * Assinatura do cliente |
| --- |

1. **Necessidades de Recursos e Ambiente**

* 1. **Ferramentas de Teste**

**JUnit 5:** Framework para desenvolvimento e execução de testes unitários em Java.

**TestLink:** Ferramenta para gerenciamento e documentação dos casos de teste manuais.

**Git / GitHub:** Sistema de controle de versão para o código-fonte e todos os artefatos de teste.

**IDE de Desenvolvimento Java:** IntelliJ IDEA para codificação e execução dos testes

* 1. **Ambiente de Teste**

Mencione os requisitos mínimos **de hardware**que serão usados para testar o software.

Os seguintes softwaressão necessários, além de softwares específicos do cliente.

* Windows 10 e superior / MacOS
* JDK 21
* MS Exchange, etc.

1. **Termos / Acrônimos**

Faça uma menção a quaisquer termos ou acrônimos usados ​​no projeto

| **TERMO / ACRÔNIMO** | **DEFINIÇÃO** |
| --- | --- |
| API | *Application Program Interface* |
| SUT | *Software Under Test* |