PLANO DE TESTES

Controle de patrimônio – ContPatri

Versão 0.1

**Goiânia, 10 de setembro de 2013**

**Histórico de revisões**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 10/09/2013 | 1.0 | Criação do Documento | Rafael Braga |
|  |  |  |  |

Sumário

[**1. Objetivos** 4](#_Toc367025961)

[**2. Propósito** 4](#_Toc367025962)

[**3. Itens de Teste** 5](#_Toc367025963)

[**4. Abordagem e Critérios** 5](#_Toc367025964)

[**5. Documentação** 8](#_Toc367025967)

[**6. Recursos** 9](#_Toc367025968)

[**7.** **Riscos** 10](#_Toc367025970)

,

## **Objetivos**

O Plano de Teste tem como objetivo o planejamento, especificação, documentação, recursos e programação de todas as atividades do teste. Este plano segue o padrão IEEE Std. 829-1998 for Software Test Documentation. Ele é referente ao projeto CONTPATRI, Controle de Patrimônio, e tem como finalidade identificar os itens que devem ser inspecionados pelos testes, descrever a abordagem de teste que será usada bem como identificar os recursos e riscos inerentes ao processo de teste.

## **Propósito**

Atualmente, na UFG, todos os bens permanentes adquiridos pela UFG – mobiliário e equipamentos diversos, por exemplo – passam por um processo de tombamento e recebem um número de patrimônio. Em cada unidade acadêmica existe o Agente Patrimonial que é responsável por, anualmente, informar o sistema da UFG, denominado SICOP (Sistema de Controle Patrimonial), a situação atual dos bens em sua unidade acadêmica.

Com o CONTPATRI espera-se obter mais agilidade e precisão no processo de coleta de dados dos bens patrimoniais, bem como a diminuição de carga de trabalho, assim como facilitar o acesso das informações e geração de relatórios.

Mediante a este cenário este documento tem por propósito o planejamento das ações relativas ao Teste de Software que cubram as funcionalidades do CONTPATRI.

## **Itens de Teste**

Apresentamos a seguir as funcionalidades que serão testadas bem como as que não farão parte do escopo do teste.

* 1. **Funcionalidades a serem testadas**

|  |  |
| --- | --- |
| **PROJETO** | **REQUISITOS FUNCIONAIS** |
| **CONTPATRI** | RF01 - Permitir o cadastro e o acesso dos agentes e gestores patrimoniais; |
| RF02 - Prover meios de interface para a coleta do número do patrimônio com suas respectivas informações; |
| RF03 - Possibilitar o informe dos bens permanentes ao SICOP através de telas específicas; |
| RF04 - Atualizar as informações de inventário oriundas do SICOP; |
| RF05 - Gerar relatórios de discrepâncias do inventário. |
| **PROJETO** | **REQUISITOS NÃO FUNCIONAIS** |
| **CONTPATRI** | RNF-01 - Toda a utilização deve requerer autenticação do usuário paras prevenir acesso não autorizado aos dados |

* 1. **Funcionalidades que não serão testadas**

|  |  |
| --- | --- |
| **PROJETO** | **REQUISITOS NÃO FUNCIONAIS** |
| **CONTPATRI** | RNF-02 - Toda mensagem de erro produzida deve ser clara e significativa, de modo a permitir que o usuário possa se recuperar do erro ou cancelar a operação. |

# **Abordagem e Critérios**

Neste capítulo serão definidas as abordagens adotadas para testar o produto de software bem como os critérios de entrada e saída. Será apresentada também a documentação que deve ser gerada pela equipe de testes.

# **Abordagem de testes**

Para efetuar os testes de validação do CONTPATRI serão usadas as seguintes abordagens:

* Teste Caixa-Branca: testes baseados na arquitetura interna do software onde é necessário que o profissional de testes conheça a tecnologia empregada pelo software, bem como um adequado conhecimento da arquitetura interna da solução. Serão executados como testes de unidades pelos desenvolvedores.
* Teste Caixa-Preta: utiliza técnicas para garantir que os requisitos do sistema são plenamente atendidos pelo software construído. Não é seu objetivo verificar como ocorrem internamente os processamentos no software, mas se o algoritmo inserido produz os resultados esperados.

## Apresenta-se a seguir os testes que serão efetuados para o SAD. Todos os planos e condições de testes serão baseados nas especificações funcionais e no documento de requisito

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase da Validação** | **Abordagem** | **Objetivos** | **Metodologia e Ferramentas** |
| Teste de Unidade | Caixa Branca (Estrutural) | - Testa a unidade  - Garantir máxima cobertura do código-fonte  - Garantir o processamento de diferentes caminhos | - Framework de testes unitários  - Framework de grau de cobertura do código |
| Teste Funcional | Caixa Preta (Funcional) | - Testa as funcionalidades do sistema  - Garantir que cada elemento do aplicativo atenda aos requisitos funcionais do negócio  - Garantir que o sistema está executando | - Casos de testes  - Scripts de testes |
| Teste de integração | - Caixa Preta  - Caixa Branca | - Testa a integração entre as partes do software  - Garantir a perfeita interface entre os diversos componentes existente na arquitetura tecnológica  - Garantir a perfeita colaboração entre componentes |  |

* 1. **Critérios**

Ao definir os critérios de teste, definem-se medidas claras para avaliar e conduzir os testes de software. Segue a tabela com os critérios definidos para esse plano de teste:

|  |  |
| --- | --- |
| **Objetivo do Teste:** | Garantir a funcionalidade apropriada da aplicação, entrada de dados, processamento e recuperação. |
| **Técnica:** | Executar cada caso de uso, fluxo de caso de uso ou função usando dados válidos e inválidos, para verificar as seguintes situações:   * Os resultados esperados ocorrem quando um dado válido é usado. * As mensagens apropriadas de erro e aviso são apresentadas quando um dado inválido é utilizado. * Cada regra de negócio é aplicada de maneira apropriada. |
| **Critério de Início:** | * Especificação Funcional e de Requisitos disponíveis e aprovados * Plano de testes disponível e aprovado * Casos de testes disponíveis e aprovados * Requisito implementado; * Ambiente de teste preparado; |
| **Critério de completeza e sucesso** | Cobertura do teste:   * Todas as funcionalidades foram exercitadas e as saídas apresentadas foram de acordo com as esperadas; * Todas as funções, inclusão, alteração e exclusão, foram refletidas no Banco de Dados. |
| **Critério de Suspensão e Retomada** | Suspensão   * Problema de login no sistema. * Banco de Dados corrompido, um defeito que impede a continuação e execução dos testes. * Sem acesso ao sistema; * Servidor de teste paralisado/desligado ou mau funcionamento.   Retomada   * Correção ao acesso do sistema replicado no ambiente de teste. * Banco de dados restaurado e ativo. * Servidor de teste ativo; |
| **Critério de Finalização:** | Todos os testes planejados foram executados.  Todos os defeitos identificados foram endereçados para a ferramenta de gerência de defeitos e resolvidos.  Com relação ao módulo testado:   * Todos os itens estarem aprovados;   Com relação ao sistema:   * Todos os módulos estarem aprovados; |

# **Documentação**

Os documentos que serão criados na documentação de teste serão as seguintes:

• Plano de Teste;

• Casos de teste;

• Procedimentos de teste;

• Relatórios de testes;

• Dados de teste usados nesse projeto;

• Programa e componentes;

# **Recursos**

* 1. **Recursos Humanos**

Abaixo segue os papeis e responsabilidades da equipe de teste.

|  |  |
| --- | --- |
| **Papel** | **Responsabilidade** |
| Gestor de Teste | * Fazer o planejamento dos testes e do esforço de teste, garantindo que os requisitos de qualidade são devidamente cumpridos com a maior eficiência possível. * Distribuir os recursos (humanos e materiais) com vista a melhorar a eficiência do processo de testes. |
| Analista de Teste | * Definir as funcionalidades a testar, garantindo o cumprimento dos requisitos definidos. * Projetar os casos de teste e os resultados esperados, avaliando a qualidade do produto. |
| Testador | * Executar os testes previamente definidos * Registrar os resultados. |
| Desenvolvedores | * Programar os testes unitários e de integração. |

## 

## Matriz de Recursos X Habilidade

Abaixo segue o apanhado dos colaboradores e sua habilidades

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Profissional** | **Habilidades Necessárias** | | | | |
| **Planejamento** | **Caso de Teste** | **Gestão de Defeitos** | **TestLink** | **Teste unitário** |
| Gestor de Teste | **V** | **V** | **V** | **V** |  |
| Analista de Teste |  | **V** | **V** | **V** |  |
| Tester |  | **V** | **V** | **V** |  |
| Desenvolvedor |  |  |  |  | **V** |

* 1. **Ambiente**

Abaixo são especificadas as características do ambiente necessário para execução dos testes, ou seja, a infraestrutura de hardware e software.

A infraestrutura de hardware pode ser verificada no Plano de Configuração.

|  |  |
| --- | --- |
| **Ferramenta** | **Descrição** |
| [Testlink](http://test.3waynet.com.br/testlink) | Ferramenta de Gerenciamento de Processos de Testes: especificação de casos de teste e execução |
| Redmine | Ferramenta para gerenciamento de defeitos |
| Selenium 2 | Framework para testes de aplicações web |
| JUnit | Framework com suporte para automação de testes unitários |
| Firebug | Ferramenta acoplada ao Mozilla Firefox para inspecionar/editar objetos no DOM. |

# **Riscos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **IDRisco** | **Causa** | **Risco** | **Plano de Mitigação** | **Plano de Contingência** | **Disparar Plano de Contingência** |
| **RS001** | Equipe nova e inexperiente. | Atraso na entrega do executável – código fonte | - Analisar o andamento do projeto frequentemente.  - Analisar o Realizado x Planejado.  Realizar treinamento com a equipe de desenvolvimento. | **-** Alertar o Gerente de Projetos para tomar as ações cabíveis. | As atividades de desenvolvimento não forem entregues no prazo.  Consequência:  Atraso no início das atividades de teste.  Atraso na entrega do projeto. |
| **RS002** | Solicitações de mudança nos requisitos no decorrer da codificação | Atraso na entrega do executável – código fonte | Analisa deve comunicar todos os envolvidos no projeto sobre a alteração dos requisitos.  O(s) analista(s) devem replicar as alterações das solicitações de mudança na documentação e em seguida commitar o documento atualizado.  Alterar os Casos de Testes de acordo com a(s) solicitação (ões) de mudança | O Gerente de projeto deve solicitar a alteração, o commit e a comunicação a todos os envolvidos sobre a(s) alteração (ões) realizada (s).  Alterar os casos de testes, que foram impactados com a alteração, de acordo com a documentação atualizada. | As atividades de teste não iniciarem conforme o planejado.  As solicitações de mudanças não são replicadas na documentação  Consequência:  Atraso no início das atividades de teste.  Atraso na entrega das atividades de testes  Teste realizado com base em documentação desatualizada.  Atraso na entrega do produto. |