|  |  |
| --- | --- |
| Tiago Quatrin  07/01/2016  English |  |
|  | Plano de Testes |
| Versão 1.0 | Especificação do Plano de Testes |

Histórico de Alterações

A cada alteração do documento, devem ser listados: a versão, a data da alteração, o responsável pela alteração, os itens que sofreram alteração e o motivo pelo qual os mesmos foram alterados.

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Elaborado por** | **Histórico de Alterações** |
| 1.0 | 07/01/2016 | Tiago Quatrin | Criação do Documento |
|  |  |  |  |
|  |  |  |  |

Índice

[1 Introdução 5](#_Toc439936450)

[1.1 Objetivos 5](#_Toc439936451)

[2 Escopo 5](#_Toc439936452)

[2.1 Estratégia de Teste 5](#_Toc439936453)

[2.2 Técnicas de Teste 5](#_Toc439936454)

[2.3 Níveis dos Testes 6](#_Toc439936455)

[3 Processo de Teste 7](#_Toc439936456)

[3.1 Papéis e Responsabilidades 7](#_Toc439936457)

[3.2 Ferramentas de Teste 7](#_Toc439936458)

[3.3 Configuração do Ambiente de Teste 8](#_Toc439936459)

[4 Resultado dos Testes 8](#_Toc439936460)

Índice de Tabelas

[Tabela 3.1 – Papéis e Responsabilidades 7](#_Toc342638578)

[Tabela 3.2 – Configuração do Ambiente de Teste 8](#_Toc342638579)

# Introdução

Este documento tem o objetivo de proporcionar a documentação do planejamento de como e do que será testado no projeto denominado Twitter, assim como os recursos utilizados para realização dos testes.

## Objetivos

A seguir são descritos os principais objetivos do plano de testes:

* Identificar informações do projeto e os componentes de software que devem ser testados;
* Identificar os recursos necessários;
* Recomendar e descrever as estratégias de testes a serem utilizadas;
* Controlar o ambiente de teste.

# Escopo

Os testes planejados e descritos neste documento se referem a validação da funcionalidade, denominada: Realizar um post do usuário, do projeto Twitter, descrita no documento <documento de especificação>.

Os itens que fazem parte do escopo dos testes são:

* + Login no sistema
  + Realização do post (tweet)
  + Verificação do post
  + Exclusão do post
  + Logout do sistema

## Estratégia de Teste

A estrégia de teste a ser utilizada é:

* **Teste Caixa Preta**: este tipo de teste analisa o item a ser testado sob o aspecto funcional. O foco dos testes de caixa preta está nas entradas, saídas esperadas e saídas obtidas, independentemente da implementação do item a ser testado.

## Técnicas de Teste

As técnicas de teste que serão utilizadas são:

* **Teste Funcional**: executa as funções associadas aos casos de uso garantindo que: a) todos os cenários (fluxos alternativos e secundários) sejam percorridos; b) as entradas fornecidas (válidas e inválidas) resultam em comportamentos definidos na especificação dos casos de uso. Tem como objetivo garantir a corretude das funcionalidades descritas a partir dos casos de uso, de modo que a aplicação se comporte conforme os fluxos de eventos e diagramas elaborados;
* **Teste de Interface**: testa o comportamento dos componentes. Se existirem componentes gráficos com comportamentos e layouts previamente definidos. Testa a navegabilidade das interfaces. Tem como objetivo assegurar que o comportamento, requisitos, projeto gráfico e navegacional definidos para as interfaces sejam atendidos;
* **Teste de Segurança e Controle de Acesso**: assegura que o modelo de permissão de acesso ao sistema e suas respectivas funcionalidades atende às especificações. **Configuração**: assegura que as configurações especificadas atendem às necessidades para o perfeito funcionamento do sistema, e que não há nenhum conflito com outros aplicativos executando na máquina.

## Níveis dos Testes

As etapas dos testes serão realizadas de acordo com o estágio do desenvolvimento do projeto:

* **Teste Unitário**: são os únicos testes realizados por desenvolvedores, e não por testadores. Eles têm o objetivo de validar individualmente itens menores (classes ou métodos básicos) da implementação de um caso de uso. Tem como objetivo garantir que o item a ser testado cumpre a função para o qual ele foi projetado;
* **Teste Integrado**:este tipo de teste verifica a compatibilidade da interface entre os componentes da aplicação, constituindo uma técnica sistemática para a construção da estrutura do programa. Tem como objetivo garantir, em uma maior escala, que os casos de uso implementados (e testados por testes de unidade) estejam interagindo entre si conforme definido;
* **Teste de Sistema**: este tipo de teste verifica se a aplicação está funcionando como um todo. A integração da aplicação com o ambiente operacional similar ao de produção (hardware, pessoas e outras aplicações) é testada. Tem como objetivo verificar uma combinação de requisitos funcionais instanciados sob diferentes condições;
* **Teste de Aceitação**:este tipo de teste verifica se a solução obedece aos critérios definidos. Isso geralmente traduz-se na verificação da satisfação do cliente e do usuário com a solução como um todo. Tem como objetivo garantir que a solução satisfaz as expectativas de cliente e usuário**.**

As etapas que fazem parte do escopo dos testes são: Teste Integrado e Teste de Sistema.

# Processo de Teste

A seguir são descritos os papéis e configurações de ambiente necessárias para a realização dos testes.

## Papéis e Responsabilidades

Os papéis e responsabilidades do são descritos na tabela 1.2.:

Tabela 3.1 – Papéis e Responsabilidades

|  |  |  |
| --- | --- | --- |
| Papel | Responsável | Responsabilidades |
| Coordenador de Teste | Tiago Quatrin | O Coordenador de Teste tem responsabilidade de:   * Acompanhar atividades da equipe de teste; |
| Analista de Teste | Tiago Quatrin | O Analista de Teste tem responsabilidade de:   * Elaborar Plano de Testes; * Elaborar Projeto de Testes (identificar, especificar os casos de testes, procedimentos e massa de dados de testes); * Executar testes manuais; * Reportar erros detectados. |
| Testador | Tiago Quatrin | O Testador tem responsabilidade de:   * Executar testes manuais; * Reportar erros. |

## Ferramentas de Teste

Serão utilizadas as seguintes ferramentas durante o processo de teste:

* Testlink: utilizada para criar os casos de testes, e suas respectivas execuções;
* Selenium IDE e Selenium Webdriver: usadas com a finalidade de criar scripts de testes automatizados;
* Netbeans IDE: ferramenta para criar, alterar e rodar os scripts automatizados, criados através do Selenium Webdriver;
* Mantis: bugtracker utilizado para controle e reporte de erros encontrados durante o processo de teste.

## Configuração do Ambiente de Teste

Os detalhes da configuração do ambiente de teste são descritos na tabela 1.1.:

Tabela 3.2 – Configuração do Ambiente de Teste

|  |  |  |  |
| --- | --- | --- | --- |
| Ambiente: | | | ( ) Intranet  ( X ) Internet  ( ) Extranet  ( ) Client Server |
| Itens | | Ambiente | |
| Teste | | |
| URL de Acesso | | https://twitter.com | | |
| Softwares Básicos | | Navegador de internet - ver item Navegador (Browser) | | |
| **Software de Apoio** | Navegador (Browser) | Navegador oficial: Mozilla Firefox com Plugin Selenium IDE instalado | | |
| Linguagem de Programação ou Ferramenta para Desenvolvimento | Netbeans IDE com suporte à JAVA, Junit e Selenium Webdriver | | |

# Resultado dos Testes

A fase de testes é dada por encerrada quando todos os requisitos previstos para a versão tiverem sido validados, seguindo os casos de testes planejados e os mesmos não apresentarem mais bugs com criticidade. O acompanhamento dos bugs deverá ser realizado pela ferramenta de bugtracker estabelecida, neste caso o Mantis.