Issues Authenticator

Plano de Testes

Versão <1.0.0>

Histórico de Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 23/10/2017 | 1.0 | Início do Plano de teste | Alessandro, Andréa e Bruno |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sumário

[1. Introdução 4](#_Toc22094979)

[1.1 Finalidade 4](#_Toc22094980)

[1.2 Escopo 4](#_Toc22094981)

[1.3 Definições, Acrônimos, e Abreviações 4](#_Toc22094982)

[1.4 Referências 4](#_Toc22094983)

[1.5 Visão geral 4](#_Toc22094984)

[2. Estratégia de Teste 4](#_Toc22094985)

[2.1 Teste <tipo de teste> 5](#_Toc22094986)

[2.1.1 Prazo para realização 5](#_Toc22094987)

[2.1.2 Recursos necessários 5](#_Toc22094988)

[2.1.3 Requisitos a serem testados 5](#_Toc22094989)

[2.1.4 Casos de Teste 5](#_Toc22094990)

[3. Resultados dos Testes 5](#_Toc22094991)

Plano de Testes

# Introdução

Neste documento estão descritos alguns casos de uso/teste para realização dos teste, bem como a estratégia de testes adotada, detalhamento dos tipos de testes para a execução dos testes, os recursos humanos necessários, assim como os produtos que serão gerados.

## Finalidade

A finalidade deste documento é apresentar as definições dos níveis e estratégias de testes que deverão ser adotadas no sistema Issues Authenticator. Além de reunir as informações necessárias para o planejamento e ao controle de esforço de teste.

## Escopo

O objetivo deste documento é descrever o planejamento dos testes que serão realizados durante as *Sprints* no projeto *Issues Authenticator*, visando garantir a qualidade por meio da detecção de falhas no software. Todos os documentos produzidos pela equipe de desenvolvimento serão verificados.

O presente documento será desenvolvido de forma incremental e iterativa a medidas que os artefatos forem sendo disponibilizados pela equipe de desenvolvimento.

## Referências

|  |  |  |  |
| --- | --- | --- | --- |
| **Título** | **Versão** | **Data** | **Onde pode ser obtido** |
| Plano de Releases | 1.1 | 23/10/2017 | Github |
| Casos de Uso | 1.1 | 23/10/2017 | Github |
| Plano de Projeto | 1.1 | 23/10/2017 | Github |
| Lista de Requisitos | 1.1 | 23/10/2017 | Github |

## Visão geral

O restante deste documento está descrito em Seções. A Seção 2 descreve a estratégia de teste utilizada neste projeto. E se dará de forma incremental, ou seja, a medida que a equipe de desenvolvimento for finalizando as sprints, serão realizado os testes propostos.

# 

# Estratégia de Teste

A estratégia de teste utilizada nesta pesquisa é: teste caixa preta -> teste de sistema (aceitação) -> teste funcional. Lembrando que esta é apenas uma estratégia escolhida inicialmente, podendo ser modificada posteriormente.

## Teste Funcional

Para os testes funcionais, deverão ser executados testes de aceitação e sistema. Como ainda não foram criadas estruturas que permitam o planejamento de testes de integração e unitários, por enquanto não foram criados casos de teste para estes dois níveis.

### Prazo para realização

O cronograma da execução do teste está descrito na Tabela 1.

**Tabela 1. Cronograma da execução do teste.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nível de Teste** | **Quando Realizar** | **Data Inicio** | **Data Fim (Previsão)** |
| ***Testes unitários*** | Realizados ao final de cada Sprint. | 20/10/2017 | 01/12/2017 |
| ***Testes de integração*** | Realizados ao final de cada Sprint, a partir da Sprint 2. | 28/10/2017 | 01/12/2017 |
| ***Teste de sistema*** | Realizados ao final de cada Sprint, a partir da Sprint 2. | 28/10/2017 | 01/02/2017 |
| ***Teste de aceitação*** | A cada release. | 01/11/2017 | 03/12/2017 |

### Recursos necessários

Para realização dos testes, os recursos necessários são:

1. Servidor de Homologação.
2. Servidor de Produção.
3. Servidor de Teste.
4. Celular com Android nas versões 4.4, 5.0, 5.1, 6.0, 7.0.
5. Computador com 4GB de Memória RAM, Processador Intel Core i3 e HD de 500 GB.
6. Pessoas envolvidas: Equipe de teste e equipe de desenvolvimento.

### Requisitos a serem testados

Os seguintes requisitos funcionais devem ser testados.

|  |  |
| --- | --- |
| **Nº do Requisito Funcional** | **Descrição** |
| RF01 | O sistema deve ser capaz de efetuar o cadastro de um usuário em sua plataforma. |
| RF02 | O sistema deve detectar o horário em que cada usuário adentra ou deixa as dependências do laboratório. |
| RF03 | O sistema deve registrar o horário em que cada usuário adentra ou deixa as dependências do laboratório. |
| RF04 | O sistema deve, para cada usuário, criar um registro do histórico de presença​ ​nas​ ​salas​ ​do​ ​laboratório. |
| RF05 | O software deve manter um histórico de presença do usuário em determinado​ ​laboratório. |
| RF06 | O sistema deve permitir a adição e remoção de aparelhos atrelados a usuários. |
| RF07 | O sistema deve registrar as preferências de um usuário com relação à temperatura, iluminação e umidade. |
| RF08 | O sistema deve ser capaz de efetuar o registro das preferências do usuário​ ​no​ ​Issues​ ​Monitoring​ |
| RF09 | O Delegado deverá ser capaz de visualizar os endereços MAC dos dispositivos conectados à rede local do laboratório em que estão alocados |
| RF10 | O sistema deve comunicar ao Issues Monitoring entradas e saídas de usuários das dependências do laboratório. |
| RF11 | Chegadas e saídas de frequentadores do laboratório devem ser registradas automaticamente, sem intervenção do​ ​usuário. |
| RF12 | O sistema deve ser capaz de avisar o usuário via aplicativo caso seja necessário o ajuste de temperatura de uma determinada dependência do laboratório. |