

# **Issues Authenticator**

## **Plano de Testes**

**Versão <1.1.0>**

Issues Authenticator	
Plano de Testes	Versão <1.1.0>1

## Histórico de Revisão

Data	Versão	Descrição	Autor
23/10/2017	1.0	Início do Plano de teste	Alessandro, Andréa e Bruno
14/11/2017	1.1	Atualização	Lucas Barcellos

Issues Authenticator	
Plano de Testes	Versão <1.1.0>1

## Sumário

1.	Introdução	4	
1.1	Finalidade	4	
1.2	Escopo	4	
1.3	Definições, Acrônimos, e Abreviações		<b>Erro! Indicador não definido.</b>
1.4	Referências	4	
1.5	Visão geral	4	
2.	Estratégia de Teste	4	
2.1	Teste <tipo de teste>	4	
2.1.1	Prazo para realização	5	
2.1.2	Recursos necessários	5	
2.1.3	Requisitos a serem testados	5	
2.1.4	Casos de Teste		<b>Erro! Indicador não definido.</b>
3.	Resultados dos Testes		<b>Erro! Indicador não definido.</b>

Issues Authenticator	
Plano de Testes	Versão <1.1.0>1

# Plano de Testes

## 1. Introdução

Neste documento estão descritos alguns casos de uso/teste para realização dos testes, 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.

### 1.1 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.

### 1.2 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.

### 1.3 Referências

Título	Versão	Data	Onde pode ser obtido
Plano de Releases	1.2	09/11/2017	Github
Casos de Uso	1.2	09/11/2017	Github
Plano de Projeto	1.2	28/10/2017	Github
Lista de Requisitos	1.2	09/11/2017	Github

### 1.4 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.

## 2. 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.

### 2.1 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.

Issues Authenticator	
Plano de Testes	Versão <1.1.0>1

### 2.1.1 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 Início	Data Fim (Previsão)
<i>Testes unitários</i>	Realizados ao final de cada Sprint.	20/10/2017	01/12/2017
<i>Testes de integração</i>	Realizados ao final de cada Sprint, a partir da Sprint 2.	28/10/2017	01/12/2017
<i>Teste de sistema</i>	Realizados ao final de cada Sprint, a partir da Sprint 2.	28/10/2017	01/12/2017
<i>Teste de aceitação</i>	A cada release.	01/11/2017	03/12/2017

### 2.1.2 Recursos necessários

Para realização dos testes, os recursos necessários são:

- i. Servidor de Homologação.
- ii. Servidor de Produção.
- iii. Servidor de Teste.
- iv. Celular com Android nas versões 4.4, 5.0, 5.1, 6.0, 7.0.
- v. Computador com 4GB de Memória RAM, Processador Intel Core i3 e HD de 500 GB.
- vi. Pessoas envolvidas: Equipe de teste e equipe de desenvolvimento.

### 2.1.3 Requisitos a serem testados

Os seguintes requisitos funcionais devem ser testados.

Nº do Requisito Funcional	Descrição
RF01	O sistema deve permitir a adição e remoção de aparelhos atrelados a usuários.
RF02	O sistema deve detectar o horário em que cada usuário adentra ou deixa as dependências do laboratório.
RF03	O software deve manter um histórico de presença do usuário em determinado laboratório.
RF04	O sistema deve registrar as preferências de um usuário com relação à temperatura, iluminação e umidade.
RF05	O sistema deve ser capaz de efetuar o registro das preferências do usuário no Issues Monitoring.
RF06	O Delegado deverá ser capaz de visualizar os endereços MAC dos dispositivos conectados à rede local do laboratório em que estão alocados
RF07	O sistema deve comunicar ao Issues Monitoring entradas e saídas de usuários das dependências do laboratório.
RF08	Chegadas e saídas de frequentadores do laboratório devem ser registradas automaticamente, sem intervenção do usuário.
RF09	O sistema deve ser capaz de verificar que o usuário está cadastrado na base de dados do Issues Monitoring.