

Nome do Projeto: Issues Authenticator	Data da Solicitação: 10/09/2017
Responsável: Equipe Cascavel	
Solicitante: Guilherme Horta Travassos	Clientes: Fábio Farzat, Guilherme Travassos, Hilmer Neri, Talita Ribeiro

Versões e Revisões deste documento			
Data	Comentário	Autor	Versão
17/09/2017	Especificação dos requisitos de um subsistema para controle de presença de usuários que atuará em conjunto com o sistema já estabelecido, o Issues Monitoring.	Equipe Cascavel	1.0
09/10/2017	Evolução do plano de projeto.	Equipe Cascavel	1.1
15/10/2017	Evolução do plano de projeto.	Equipe Cascavel	1.2
17/10/2017	Correções no plano de projeto.	Equipe Cascavel	1.3
01/11/2017	Correções no plano de projeto.	Equipe Cascavel	1.4
26/11/2017	Correções no plano de projeto.	Equipe Cascavel	1.5
30/11/2017	Correções no plano de projeto.	Equipe Cascavel	1.6

Issues Authenticator

Plano de Projeto

1 Visão

A Equipe Cascavel é composta por estudantes da disciplina Engenharia de Software, ministrada pelo professor Guilherme Travassos.

Atualmente o sistema Issues Monitoring monitora as condições de ambiente dos laboratórios, cadastra os usuários autorizados do laboratório, monitora os usuários presentes em cada laboratório e envia alertas acerca das condições ambientais dos laboratórios. A autenticação dos usuários no laboratório é feita por um sistema de biometria, o MyDenox. Ele é um sistema de autenticação terceirizado que foi incorporado ao esquema de autenticação do Issues Monitoring. Esse sistema exige hardware para a realização da autenticação mas não há dispositivos de hardware suficientes para todos os laboratórios, o que seria custoso.

Assim, a Equipe Cascavel propõe uma solução para otimizar o processo de autenticação nos laboratórios e reduzir a dependência de hardwares externos. O Issues Authenticator deve ser um aplicativo Android de autenticação autônoma dos usuários do LENS. Com uma interface simples, ele deve permitir que o usuário controle suas preferências ambientais e visualize as mensagens de alerta recebidas do Issues Monitoring.

1.1 Escopo do Projeto

O software a ser desenvolvido deve ser capaz de se comunicar com o Issues Monitoring. Deve efetuar o login no sistema para usuários já cadastrados no Issues Monitoring, e avisar o usuário não cadastrado da necessidade de cadastro prévio. Ainda, o software deve ser capaz de informar ao Issues Monitoring as preferências do usuário acerca da temperatura e da luminosidade para o ambiente do laboratório. O software também precisa ser capaz de exibir as mensagens enviadas pelo Issues Monitoring.

1.2 Escopo Não Incluído no Projeto

O software Issues Authenticator não intenciona resolver como o Issues Monitoring irá decidir quais preferências ambientais deverão ser ativadas em um ambiente com mais de um usuário presente. O software Issues Authenticator não intenciona resolver como será feito o cadastro de usuários no sistema Issues Monitoring.

1.3 Glossário

Termo	Descrição
CT	Centro de Tecnologia
CT-2	Centro de Gestão Tecnológica
LENS	Laboratório de Engenharia de Software
Preferências ambientais	Temperatura e umidade consideradas ideais pelo usuário do sistema para o laboratório

2 CRONOGRAMA

O estabelecimento de um cronograma visa a apresentação do projeto de software e sua respectiva documentação nos prazos estabelecidos pelos stakeholders. Pela natureza do projeto e o fato de haver três *releases* já marcadas, o modelo mais apropriado para o ciclo de vida do produto é o modelo incremental, visto que em cada *release* é necessário realizar testes que podem indicar possíveis defeitos que devem ser corrigidos antes da entrega.

Os papéis não são estáticos e é esperado que cada indivíduo contribua em diversas áreas do projeto.

Data	Entregáveis
18/09/2017	Plano de Projeto Inicial incluindo a descrição inicial dos requisitos funcionais e não funcionais de Tecnologia.
27/09/2017	Especificação dos requisitos (cenários) e refinamento do Plano de Projeto (priorização de requisitos, atividades de garantia da qualidade e plano de releases).
18/10/2017	Revisão e evolução dos requisitos, Modelos de projeto iniciais (diagramas, arquitetura, layouts de interface humano computador, etc.) e proposta de integração com o Issues Monitoring. Refinamento do plano de releases. Indicação do ambiente de desenvolvimento no GitHub.
18/10/2017 (1 semana de duração)	SPRINT 1.1: Implementação dos requisitos funcionais RF01, RF02, RF09, RF12, RFIM02 e RFIM04. Verificação da compatibilidade da solução com os requisitos não funcionais RNF01, RNF02, RNF08 e RNFIM01.
25/10/2017 (1 semana de duração)	SPRINT 1.2: Implementação dos requisitos funcionais RF05, RF06, RF11, RFIM03. Verificação da compatibilidade da solução com os requisitos não funcionais RNF01, RNF11.
01/11/2017	RELEASE 1: Conforme planejado pela equipe e acompanhado de informação sobre os requisitos tratados, modelos de projeto, código fonte, laudos de teste funcional e manual de instalação e do usuário.
08/11/2017 (1 semana de duração)	SPRINT 2: Implementação dos requisitos funcionais RF03, RF04, RF05, RF06, RF08. Verificação da compatibilidade da solução com os requisitos não funcionais RNF04, RNF09, RNF10 e RFIM03. <i>(alguns requisitos foram realocados para sprint 2)</i>
15/11/2017	RELEASE 2: Conforme planejado pela equipe e acompanhado de informação sobre os requisitos tratados, modelos de projeto, código fonte, laudos de teste funcional e manual de instalação e do usuário.
22/11/2017 (1 semana de duração)	SPRINT 3.1: Implementação dos requisitos funcionais RF07, RFIM01. Verificação da compatibilidade da solução com os requisitos não funcionais RNF03 e RNFIM02.
29/11/2017 (aproximadamente 1 semana de duração)	SPRINT 3.2: Ajustes e verificação de conformidade com RNF02, RNF05, RNF06, RNF07 e RNF08.
03/12/2017	RELEASE 3: Entrega do produto final, acompanhado de informação sobre todos os requisitos tratados, modelos de projeto, código fonte, laudos de teste funcional e manual de instalação e do usuário.
04/12/2017	Entregar até as 10:00 h (antes da aula) o material a ser usado na apresentação do produto para os stakeholders (em template a ser disponibilizado para os times).

3 ORGANIZAÇÃO DA EQUIPE

3.1 Envolvidos no Projeto

Nome	Papel
FELIPE ASSIS	Desenvolvedor
GABRIEL MORGADO	Gerente de Projeto
LUIS EDUARDO PESSOA	Desenvolvedor
PEDRO REIS	Desenvolvedor
VITOR TEIXEIRA	Product Owner

- Gerente de Projeto: atribuição principal consiste em gerenciar a equipe além monitorar e revisar o projeto e seus respectivos artefatos.
- Product Owner: atribuições semelhantes ao gerente de projeto mas com um foco maior no desenvolvimento do produto em detrimento dos artefatos.
- Desenvolvedor: atribuição central consiste em codificar o problema com base nos requisitos.

4 DESCRIÇÃO DO SISTEMA PROPOSTO

O Issues Authenticator deve ser um aplicativo de fácil uso e rápido tempo de execução dentro da plataforma Android, com uma interface simples que permita o usuário ter acesso a suas preferências em uma única tela após um login efetuado com sucesso. O Issues Authenticator permite que o ambiente se antecipe ao usuário dado que suas preferências já estão salvas no banco de dados para serem usadas pelo sistema. Os requisitos esperados no sistema estão detalhadamente descritos no arquivo de requisitos.

5 PADRÕES DE PROJETO, TÉCNICAS E FERRAMENTAS PROPOSTAS

O projeto será desenvolvido pelos cinco membros da Equipe Cascavel, trabalhando independentemente, cada um em seu escritório particular, com o auxílio da ferramenta de controle de versão GitHub. Espera-se, ao utilizar essa ferramenta, o desenvolvimento organizado e controlado do sistema. Quanto aos ambientes de desenvolvimento utilizados para codificação, cada componente da equipe terá a liberdade de utilizar a IDE de sua preferência.

- O aplicativo será desenvolvido utilizando *React Native* com algumas dependências importantes, sendo elas:
 - *Redux*: responsável por controlar a máquina de estados referente ao fluxo de dados da aplicação. Como muitas das interações no aplicativo são através de “formulários”, se torna uma dependência importante para lidar com o caráter reativo do *React*.
 - *React Redux*: para integrar a máquina de estados do fluxo de dados com a aplicação *React*.
 - *Redux Thunk*: *middleware* que facilita as requisições assíncronas referentes à máquina de estados do fluxo de dados.

Plano de Projeto

- *React Navigation*: responsável por controlar a máquina de estados referente à navegação entre as telas do aplicativo.
- *React Native Android Location*: módulo para acesso à geolocalização do usuário.
- *React Native Android Wifi*: módulo para acesso aos dados de WiFi do usuário.
- *React Native Background Task*: módulo para registro de tarefas que rodam no plano de fundo do dispositivo.
- *Geodist*: para calcular a distância entre dois pontos no espaço a partir de latitude e longitude.
- A parte do *Issues Monitoring* a ser modificada está na linguagem *Python*.
- Será utilizado o linter *ESLint* para identificar pequenos defeitos no código.
- Também foi definido o *style guide Standard JS* que foi configurado no linter *ESLint* junto com a extensão *Prettier ESLint* para padronizar a formatação do código;

6 PLANO DE GARANTIA DE QUALIDADE

Considerando a conformidade com os requisitos como métrica principal para medir a qualidade do software, a qualidade será mantida através de testes para assegurar que os requisitos são atendidos.

7 PLANO DE DOCUMENTAÇÃO

Dentre os artefatos construídos constam:

- Plano de Projeto;
- Lista de Requisitos;
- Documento de Caso de Uso;

Dentre os artefatos a serem construídos constam:

- Manual de Instalação e Configuração do Sistema (release 1);
- Manual de Instalação e Configuração do Sistema (release 2);
- Manual de Instalação e Configuração do Sistema (release 3)
- Laudo de Testes (release 1);
- Laudo de Testes (release 2);
- Laudo de Testes (release 3);
- Documento de Modelos de Projeto;

8 PLANO DE TESTES

Os níveis de testes a serem realizados são os seguintes:

- Testes funcionais
- Testes de desempenho
- Testes de tolerância a falhas

- Testes de instalação e configuração

9 PLANO DE ENTREGA

A estimativa de entrega, atendendo ao prazo especificado pelos stakeholders, é 3 de dezembro de 2017. Esta é a data da entrega do produto final, acompanhado de informação sobre todos os requisitos tratados, modelos de projeto, código fonte, laudos de testes e manual de instalação.

10 PLANO DE MANUTENÇÃO

O plano de manutenção do sistema será definido em conjunto com os stakeholders em etapa posterior. A ideia geral é que seja feito o menor número possível de modificações no Issue Monitoring e que a documentação do Issues Authenticator esteja elucidativa e coerente a fim de facilitar a manutenção.

11 PLANO DE GERENCIAMENTO DE RISCOS

1. Cumprir prazo estabelecido:

Um dos maiores desafios do projeto é conseguir entregar o desenvolver o software dentro do prazo estabelecido. Existem diversos fatores que afetam o andamento do projeto como disponibilidade de tempo dos membros da equipe e domínio do problema e tecnologias utilizadas. Para reduzir o impacto destes fatores foi adotado o método de *sprints* que está organizado no Github. Assim, é possível ver quais funcionalidades precisam ser implementadas até a próxima *release*. Além disso, a troca de informação entre os membros da equipe é essencial.

2. Hospedar o Issues Monitoring:

É necessário hospedar a versão do Issues Monitoring modificada para funcionar com o aplicativo Issues Authenticator. Uma opção é usar o serviço da Digital Ocean.

12 INDICADORES DE SUCESSO

Indicador de Desempenho: Implementar os requisitos descritos no cronograma dentro do prazo de cada *sprint* ou *release*. Significa que o esforço estimado foi próximo do esforço efetivo.

Indicador de Impacto: Cliente satisfeito com o produto final. Somente pode ser avaliado no fim do projeto.

Indicadores Operacionais: Analisar as *milestones* definidas no *Github* e avaliar a progressão do projeto. Verificar o que ainda precisa ser feito e o tempo restante para ter uma visão do quanto o projeto tem progredido.

13 CRITÉRIO DE ACEITE

O desenvolvimento do projeto está dividido entre três *releases*, cada uma com um conjunto de condições que devem ser satisfeitas. A aceitação do *software* está estritamente relacionada a estas condições.

- Software em conformidade com os requisitos.
 - Identificar entrada e saída de usuário
 - Configurar preferências ambientais
- Software bem documentado.
- Projeto concluído no prazo estabelecido.

Aprovadores: Fábio Farzat, Guilherme Travassos, Hilmer Neri e Talita Ribeiro