



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

CARLOS RANAN ARAUJO LIMA

**DESENVOLVIMENTO DE UM APLICATIVO DIRECIONADO A AUXILIAR OS
USUÁRIOS DO ÔNIBUS ESCOLAR UNIVERSITÁRIO DA UNIVERSIDADE
FEDERAL DO CEARÁ E INSTITUTO FEDERAL DO CEARÁ CAMPUS –
QUIXADÁ-CE**

QUIXADÁ

2020

CARLOS RANAN ARAUJO LIMA

DESENVOLVIMENTO DE UM APLICATIVO DIRECIONADO A AUXILIAR OS
USUÁRIOS DO ÔNIBUS ESCOLAR UNIVERSITÁRIO DA UNIVERSIDADE
FEDERAL DO CEARÁ E INSTITUTO FEDERAL DO CEARÁ CAMPUS – QUIXADÁ-
CE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Profa. Ma. Antônia Diana Braga Nogueira.

QUIXADÁ

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- L697d Lima, Carlos Ranan Araujo.
Desenvolvimento de um aplicativo direcionado a auxiliar os usuários do ônibus escolar universitário da Universidade federal do Ceará e Instituto federal do Ceará campus Quixadá-CE / Carlos Ranan Araujo Lima. – 2020.
65 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Ciência da Computação, Quixadá, 2020.
Orientação: Profª. Ma. Antônia Diana Braga Nogueira..
1. Software-Desenvolvimento. 2. Serviço baseado em localização. 3. Desenvolvimento de software móvel.
I. Título.

CDD 004

CARLOS RANAN ARAUJO LIMA

DESENVOLVIMENTO DE UM APLICATIVO DIRECIONADO A AUXILIAR OS
USUÁRIOS DO ÔNIBUS ESCOLAR UNIVERSITÁRIO DA UNIVERSIDADE
FEDERAL DO CEARÁ E INSTITUTO FEDERAL DO CEARÁ CAMPUS – QUIXADÁ-
CE

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da
Computação do Campus de Quixadá da
Universidade Federal do Ceará, como
requisito parcial à obtenção do grau de
bacharel em Ciência da Computação.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

Profa. Ma. Antônia Diana Braga Nogueira (Orientadora)
Universidade Federal do Ceará (UFC)

Profa. Dra. Paulyne Matthews Jucá
Universidade Federal do Ceará (UFC)

Prof. Me. Marcos Dantas Ortiz
Universidade Federal do Ceará (UFC)

A Deus,
Aos meus pais e minha irmã. Gercian,
Carlos e G ssica.

AGRADECIMENTOS

Agradeço primeiro a Deus por me manter na trilha certa durante este projeto de pesquisa com saúde e forças para chegar até o final.

À minha mãe, Maria Gercian, que sempre fez o possível e o impossível para me ajudar a conquistar meus objetivos sempre me incentivando a estudar. Nada que eu diga ou faça pode retribuir a dívida de amor que tenho por você.

Ao meu pai, Carlos, que mesmo tão longe esteve sempre perto para me apoiar nos momentos mais difíceis.

À minha irmã Géssica, que sempre me ajudou e esteve comigo em todos os momentos mesmo e quando não era possível para ela. Sou muito grato por tê-la como irmã!

Aos meus tios Ivan, Ildisvan, Gerciane e Gercilede que nos momentos mais difíceis sempre me aconselharam e me apoiaram a concluir minha graduação.

Ao meu primo, Danton, um guru da matemática, que me ajudou em muitas disciplinas durante minha graduação e nesse trabalho.

À minha atual namorada, Jhessica, a pessoa que merece os melhores agradecimentos possíveis por me ajudar e aconselhar nesse trabalho do início ao fim, mesmo quando não podia. Muito obrigado, Jhessica!

Sou grato à minha família pelo apoio que sempre me deram durante toda a minha vida em especial a Araujada. Obrigado a todos!

A todos os meus amigos, que de alguma forma me incentivaram.

À professora Diana, por sua orientação, ensinamentos e paciência comigo durante esse trabalho, obrigado, Diana!

Aos professores participantes da banca examinadora professor Marcos e professora Paulyne pelo tempo, pelas valiosas colaborações e sugestões. Muito obrigado!

A todos os meus professores da UFC pela excelência da qualidade técnica de cada um, não só no ensinamento profissional como também a ser uma boa pessoa.

“Palavras são, na minha nada humilde
opinião, nossa inesgotável fonte de magia.
Capazes de formar grandes sofrimentos e
também de remediá-los.”

(JK. Rowling)

RESUMO

Em 2007, a prefeitura de Quixadá fez um acordo, juntamente com os campi federais, assumindo a responsabilidade de transportar a custo zero os estudantes universitários até que fosse implementado linhas urbanas (G1, 2019). No ano de 2020, ainda não foram implementadas linhas urbanas de ônibus em Quixadá e existem três ônibus que fazem o percurso da Praça José de Barros para os campi UFC/IFCE, sendo um disponibilizado pela UFC e os outros dois pela Prefeitura Municipal de Quixadá - CE. A pequena quantidade da frota de ônibus disponibilizada gera incertezas para os alunos, que não sabem o horário correto e nem se haverá ônibus fazendo o trajeto. Neste trabalho, propõe-se a identificação, o planejamento e o desenvolvimento de uma ferramenta com finalidade de facilitar o deslocamento dos estudantes que utilizam o transporte universitário em Quixadá, usando os serviços LBS e uma adaptação do SCRUM com sprints semanais. O presente estudo foi dividido em duas etapas. A primeira foi a elaboração de um questionário e uma análise das ferramentas já existentes no mercado, que possuem funcionalidades semelhantes ao objetivo específico dessa pesquisa. A segunda foi o desenvolvimento e a execução das sprints do Scrum Solo que foram planejadas a partir dos resultados da primeira etapa. Assim, foi criado a aplicação PASTORABUS para acessar e disponibilizar a localização em tempo real do ônibus universitário. Nos testes e validação das funcionalidades, foi disponibilizado a um grupo de validação o arquivo apk da aplicação e a ata de validação para validar as funcionalidades. O aplicativo PASTORABUS pode ser muito útil para os estudantes dos campi UFC e IFCE de Quixadá devido à funcionalidade de mostrar a posição em tempo real do ônibus, permitindo um melhor aproveitamento do tempo, pois o estudante saberá a localização do ônibus. As informações de consultas poderão ser úteis para os novos estudantes dos campi para analisar quais os ônibus que fazem a rota como também as paradas que fazem parte do percurso.

Palavras-chave: Location Based Service (LBS). Scrum Solo. Android. Desenvolvimento móvel

ABSTRACT

In 2007, a city hall in Quixadá made an agreement, together with federal campuses, assuming the responsibility of transporting university students zero cost until urban lines were implemented (G1, 2019). In 2020, urban bus lines have not yet been implemented in Quixadá and there are three buses that run from Praça José de Barros to UFC / IFCE campuses, one made available by UFC and the other two by the city administration of Quixadá - CE. The small amount of available buses generates uncertainties for students, who do not know the correct time and whether there will be buses doing the route. In this work, we propose the identification, planning and development of a tool to improve the displacement of students who use college transport in Quixadá, using LBS services and an adaptation of SCRUM with weekly sprints. This study was divided into two stages. The first one was the elaboration of a questionnaire and an analysis of the tools already on the market, which have characteristics similar to the specific objective of this research. The second one was the development and execution of the Scrum solo sprints that were planned based on the results of the first stage. Thus, the PASTORABUS application was created to access and make available the real-time location of the university bus. In the tests and validation of the functionalities, the application's apk file and the validation minutes were made available to a validation group to validate the functionalities. The PASTORABUS app can be very useful for students on the UFC and IFCE Quixadá campuses due to the functionality of showing the bus position in real time, allowing a better use of time, as the student will not be in the queue for hours. Consultation information could prove useful for new campuses students to analyze which buses make the route as well as the bus stops that are part of the route.

Keywords: Location Based Service (LBS). Scrum Solo. Android. Development.

LISTA DE FIGURAS

Figura 1-Fluxo Do Scrum.....	20
Figura 2- Fluxo do processo Scrum Solo.	21
Figura 3- Atividade de requeriment do Scrum Solo	22
Figura 4 - Atividade de Sprint do Scrum Solo.....	23
Figura 5 -Atividade de deployment do Scrum Solo.	24
Figura 6- Atividade de Gestão do Scrum Solo	25
Figura 7- Análise das ferramentas já existentes no mercado.....	39
Figura 8 – Tela de permissão de compartilhamento PASTORABUS.....	44
Figura 9 - Conjunto de círculos com raio de 50 metros por todo o percurso.	45
Figura 10 - Tela de visualização dos dados das paradas.....	47
Figura 11- Modelo de armazenamento de dados das posições compartilhadas.	48
Figura 12 – Tela de consulta dos ônibus.....	50
Figura 13 – Tela de informações aplicação PASTORABUS.....	52
Figura 14 - Estrutura analítica do projeto PASTORABUS.	55
Figura 15- Diagrama de caso de uso aplicação PASTORABUS.....	59

LISTA DE GRÁFICOS

Gráfico 1 - Percentual de idade dos usuários do ônibus escolar acadêmico	32
Gráfico 2 - Percentual de acadêmicos dos campi UFC/IFCE	32
Gráfico 3 - Percentual de usuários com transporte próprio fazendo o percurso.....	33
Gráfico 4 - Os horários de maior fluxo do ônibus universitário.....	33
Gráfico 5 - A falta de qualidade do transporte universitário pode prejudicar no rendimento do aluno.	34
Gráfico 6- Lista de funcionalidades e o percentual de aprovação delas por cada usuário. ..	34

LISTA DE QUADROS

Quadro 1 - Comparação dos trabalhos relacionados e este trabalho.	18
Quadro 2 - Product backlog do aplicativo PASTORABUS.	40
Quadro 3 - Ferramentas utilizadas para o desenvolvimento da aplicação PASTORABUS .	41
Quadro 4 - Sprint backlog da aplicação PASTORABUS.	42
Quadro 5 - Ata de validação da aplicação PASTORABUS.	53
Quadro 6- Calendário da aplicação PASTORABUS.	56
Quadro 7- Calendário da execução das sprints da aplicação PASTORABUS.	57

LISTA DE ABREVIATURAS E SIGLAS

LBS	<i>Location-Based Services</i>
IFCE	Instituto Federal do Ceará
UFC	Universidade Federal do Ceará
FNDE	Fundo Nacional de Desenvolvimento
GPS	<i>Global Positioning System</i>
FLP	<i>Fused Location Provider</i>
JSON	<i>JavaScript Object Notation</i>
XML	<i>Extensible Markup Language</i>
SDK	<i>Software Development Kit</i>
PSP	<i>Personal/Software Process</i>

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Objetivos	15
<i>1.1.1 Objetivo geral</i>	16
<i>1.1.2 Objetivo específico</i>	16
1.2 Organização	16
2 TRABALHOS RELACIONADOS	16
2.1 Aplicação Bustracker – Oferecendo uma Melhor Experiência aos Usuários do Transporte Urbano, a Partir da Itilização de Informações de Rastreamento Veicular.	16
2.2 Ubibusroute: Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais.	17
2.3 Um Sistema Web de Consulta de Trajetos de Transporte Público.	17
2.4 Análise omparativa	18
3 FUNDAMENTAÇÃO TEÓRICA	19
3.1 Scrum Solo	19
<i>3.1.1 Atividade do processo [pagotto, et al (2016)].</i>	22
<i>3.1.2 Atores do processo [pagotto, et al (2016)].</i>	25
<i>3.1.3 Artefatos do processo [pagotto, et al (2016)].</i>	26
3.2 Location-Basead Services	27
<i>3.2.1 Modelo de Interação</i>	27
<i>4.2.2 Localização Simbólica</i>	28
<i>3.2.3 Posição x localização</i>	28
<i>3.2.4 Perfis de usuário</i>	29
4 PROCEDIMENTOS METODOLÓGICOS	29
4.1 Tipos do Estudo	29
4.2 Coletas de Dados	30
4.3 Análise de Dados	30
4.4 Desenvolvimento do Aplicativo PASTORABUS	30
4.5 Etapa de Testes	31
4.6 Melhorias e Correções de Bugs	31
5 DESENVOLVIMENTO DO APLICATIVO PASTORABUS	31
5.1 Questionário	31
5.2 Análise de Ferramentas	35
<i>5.2.1 Transit</i>	35
<i>5.2.2 Moovit</i>	35
<i>5.2.4 Cadê o Ônibus?</i>	36
<i>5.2.5 Cittamobi</i>	37
<i>5.2.6 Meu ônibus - fortaleza</i>	37
<i>5.2.7 Google maps go</i>	38
5.3 Execução das Atividades Utilizando o Scrum Solo	39
<i>5.3.1 Requeriment</i>	39
<i>5.3.2 Sprint</i>	41
<i>5.3.3 Deployment</i>	53
<i>5.3.4 Management</i>	54
5.4 Bugs e Melhorias	58
<i>5.4.1 Correção de bugs</i>	58

6 CONSIDERAÇÕES FINAIS.....	59
REFERÊNCIAS	62

1 INTRODUÇÃO

O crescimento exponencial das tecnologias envolvendo telefones celulares e os preços de dispositivos mais acessíveis para a população junto com a disseminação das redes móveis 3G e 4G fez com que surgisse um crescimento no mercado de aplicativos móveis (SIQUEIRA, 2012). Com a grande quantidade de celulares no nosso dia a dia, é essencial o desenvolvimento de aplicações que proporcionem uma maior comodidade para a população e, assim, o tempo que seria gasto para atividades de menor interesse pode ser gasto com atividades de maior interesse do usuário.

Muitas ferramentas existentes no mercado fazem uso de geolocalização para atingirem suas finalidades são as chamadas *Location-Based Services* (LBS). São ferramentas que utilizam da localização de algum usuário específico para agregar valor para alguma finalidade. Em uma pesquisa realizada pela empresa Kanta TNS (2018), o tipo de aplicação mais usado pelos usuários envolvendo o serviço LBS é de navegação (49%), e, em seguida, vem a procura por restaurantes (26%), encontrar pessoas de seu interesse (22%), serviços de transporte público (19%) e para anúncios e ofertas (13%).

Segundo Silva (2000), um dos principais motivos para as pessoas não optarem pelo o transporte público como forma principal de locomoção é a falta de informações sobre o horário que cada veículo irá passar no ponto de parada e que se essa demanda de informações fosse suprida, o tempo de espera nas paradas dos usuários de transporte público seria menor.

A cidade de Quixadá no Estado do Ceará é conhecida como um polo estudantil com a presença de diversas universidades, dentre elas o Instituto Federal do Ceará (IFCE) e a Universidade Federal do Ceará (UFC - Campus Quixadá). Ambas as instituições são localizadas a aproximadamente 4,9 km do centro da cidade (Praça José de Barros), o que torna necessário o uso de transporte universitário. Em 2007, a prefeitura de Quixadá fez um acordo, juntamente com os campi federais, assumindo a responsabilidade de transportar a custo zero os estudantes universitários até que fosse implementado linhas urbanas (G1, 2019).

Em 2013, a então presidenta Dilma Rousseff (PT), promulgou uma ementa da lei 12.816/13, que autoriza os gestores municipais a utilizarem o Fundo Nacional de Desenvolvimento (FNDE) para regulamentar o transporte escolar para uso universitário. Mesmo após o acordo feito pela Prefeitura Municipal de Quixadá e o direito ao transporte

escolar universitário previsto em constituição, a realidade vivida é outra, representando prejuízos em vários aspectos para os alunos dos campi citados.

No ano de 2020, ainda não foram implementadas linhas urbanas de ônibus em Quixadá e existem três ônibus que fazem o percurso da Praça José de Barros para os campi UFC/IFCE, sendo um disponibilizado pela UFC e os outros dois pela Prefeitura Municipal de Quixadá - CE. A pequena quantidade da frota de ônibus disponibilizada gera incertezas para os alunos, que não sabem o horário correto e nem se haverá ônibus fazendo o trajeto. Para França e Mello (2016), as redes de transportes são fundamentais na sociedade atual por representarem uma influência direta na qualidade de vida e na eficiência dos transportes coletivos.

Os aplicativos Moovit¹, Transit² e Citymapper³ são exemplos de sistemas já disponibilizados no mercado móvel que surge para atender essa necessidade do usuário. Segundo o centro de pesquisas relacionado aos transportes, Mobility Lab (2018), esses aplicativos são fundamentais para obter as melhores informações de mobilidade urbana. Entretanto, essas ferramentas abrangem uma grande quantidade de localidades e inúmeros meios de transporte, dessa forma, elas não são interessantes para um local específico que possua poucos ônibus fazendo suas rotas.

Este trabalho visa utilizar esses conceitos para desenvolver um aplicativo com finalidade de facilitar o deslocamento dos estudantes que utilizam o transporte universitário em Quixadá. A solução irá utilizar os serviços LBS para acessar a localização do usuário, que está dentro do transporte e, desse modo, informar ao usuário que aguarda a posição do ônibus e os horários previstos para determinado ônibus passar em uma parada do trajeto. Para o desenvolvimento, a solução irá utilizar uma adaptação do SCRUM com sprints semanais. Ao final de 10 *sprints*, teremos a primeira *release* do produto.

1.1 Objetivos

Nesta seção, serão apresentados o objetivo geral e os objetivos específicos deste trabalho.

¹ <https://play.google.com/store/apps/details?id=com.tranzmate>

² <https://play.google.com/store/apps/details?id=com.thetransitapp.droid>

³ <https://play.google.com/store/apps/details?id=com.citymapper.app.release>

1.1.2 Objetivo geral

Apresentar o desenvolvimento de um aplicativo direcionado a auxiliar o usuário do ônibus escolar universitário (UFC e IFCE) em Quixadá, fornecendo informações, como localização do ônibus e tempo de espera.

1.1.2 Objetivo específico

- Aplicar uma adaptação do Scrum para equipes de um só indivíduo no desenvolvimento do software, descrevendo os resultados de cada etapa, artefatos desenvolvidos e decisões tomadas durante todo o projeto.
- Desenvolver um aplicativo, utilizando os serviços LBS para implementar a função que irá utilizar o aluno que está dentro do ônibus como um GPS para obter a localização em tempo real do transporte universitário.
- Aplicar uma avaliação no software baseado em uma entrevista que será produzida para validar a qualidade do sistema desenvolvido.

1.2 Organização

Este trabalho está organizado em 6 tópicos: na seção 2, estão descritos os trabalhos relacionados; na seção 3, é apresentada a Fundamentação Teórica; na seção 4, os procedimentos metodológicos são descritos; na seção 5, mostra-se o desenvolvimento do aplicativo PASTORABUS e, finalmente, na seção 6 conclui-se o trabalho.

2 TRABALHOS RELACIONADOS

Nesta seção, serão descritos os estudos que possuem relação com o trabalho proposto, destacando as semelhanças e diferenças com esse trabalho.

2.1 Aplicação Bustracker – Oferecendo uma Melhor Experiência aos Usuários do Transporte Urbano, a Partir da Itilização de Informações de Rastreamento Veicular.

Em um estudo realizado por Siqueira (2012), é definido um modelo de armazenamento de dados para informações de localização de ônibus geradas por uma infraestrutura de

rastreamento dos veículos. Tendo como ponto inicial o modelo de armazenamento, é definida a aplicação *Bus Tracker*, a qual funciona no modelo cliente-servidor, e tem como funcionalidade permitir que o usuário, a partir de algum dispositivo móvel, tenha acesso à localização do transporte urbano.

2.2 Ubibusroute: Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais.

No trabalho desenvolvido por Lima et al. (2012), é proposto um sistema de recomendação de rotas aos usuários de transporte público, projetado na arquitetura cliente-servidor, que utiliza dados dinâmicos, obtidos de redes sociais, que possuem relação com o contexto do sistema. A recomendação das rotas é baseada em pontos de origem, destino e tipo de busca, fornecidos pelos usuários.

Do lado do cliente, funciona a parte do aplicativo móvel que vai enviar requisição de dados estáticos e um pedido de rota ao servidor, recebendo como resultado a rota e os dados para percorrê-la. Já do lado do servidor, vai existir uma subdivisão do mesmo em três componentes principais: Extrator de Informações Contextuais (responsável por colher informações das redes sociais e apura a situação do trânsito naquele momento); Identificador de Rotas (que identificar todas as rotas possíveis); e Indicador de Rotas (responsável por selecionar a melhor rota de acordo com a preferência do usuário em relação ao tempo, gastos ou distância).

O aplicativo móvel foi desenvolvido sobre a plataforma Android, em Java e usando alguns recursos da API do *Google Maps*. O aplicativo revela a possibilidade de recomendar rotas de ônibus aos usuários, a partir de informações dinâmicas de redes sociais, mas afirma que ainda existe uma demanda de melhorias no sistema que posteriormente serão abordadas em pesquisas futuras.

2.3 Um Sistema Web de Consulta de Trajetos de Transporte Público.

Freitas e Jaques (2011) propõem um sistema web para consulta e calcular rotas em trechos de ônibus e a pé para usuários do transporte público, permitindo a possibilidade de estimar os tempos de baldeação e, com isso, planejar sua viagem. Dada uma localização inicial (rua e número) e um destino, o sistema pressupõe as linhas de ônibus que o usuário deve utilizar para chegar ao seu objetivo.

Para buscar o melhor percurso, o sistema utiliza das técnicas de inteligência artificial e com algoritmo de busca heurística A*. A aplicação acarreta em percursos que compreendem o momento em que se está utilizando linhas de ônibus e o momento em que se está utilizando em trechos a pé. O *feedback* dos usuários do aplicativo foi positivo. Além disso, todos consentiram que as funcionalidades implementadas no sistema, atendem a proposta do trabalho.

2.4 Análise Comparativa

O Quadro 1 apresenta uma comparação dos trabalhos relacionados e este trabalho.

Quadro 1 - Comparação dos trabalhos relacionados e este trabalho.

	Siqueira (2012)	Lima et al. (2012)	Freitas e Jaques (2011)	Trabalho proposto
Usa serviços LBS	Faz uso do serviço LBS	Não identificado	Não identificado	Faz uso do serviço LBS
Fornecem dados da localização do ônibus	Faz uso do serviço LBS	Não identificado	Não identificado	Faz uso do serviço LBS
Recomenda Rotas	Não identificado	Faz uso do serviço LBS	Faz uso do serviço LBS	Não possui
Define uma infraestrutura para coletar informações referentes à localização do ônibus	Não identificado	Não identificado -	Não identificado	Faz uso do serviço LBS
Metodologia de desenvolvimento	PROJETO UNIFICADO	Não identificado	Não identificado	Scrum Solo

Fonte: elaborado pelo autor.

Em Siqueira (2012), a aplicação não foca em uma infraestrutura de rastreamento veicular. No caso, não é definida nem uma forma de obtenção dos dados para serem utilizados no rastreamento do veículo. Como forma de suprir a ausência desses dados, é feito uma simulação em que um *script* roda em conjunto com o servidor, gerando localizações falsas em um pequeno intervalo de tempo para simular a localização do ônibus. Em um segundo momento, é instalado um rastreador em um dispositivo móvel que tem a função de a cada 10 segundos recuperarem a localização do aparelho e enviar para o modelo de armazenamento. Nesse ponto esse trabalho se diferencia da aplicação *Bus Tracker*, pois a proposta é definir uma

infraestrutura de rastreamento veicular que irá utilizar o aluno que está dentro do ônibus como um GPS para obter a localização do transporte universitário, enquanto que o Bus Tracker não foca sua aplicação em uma infraestrutura de rastreamento veicular.

A interseção desse trabalho com o de Lima et al. (2012) ocorre pelo fato de ambos proporem melhorias no transporte urbano, mas com sugestões diferentes. O presente estudo busca a construção de um sistema para fornecer informações, como localização do ônibus e tempo de espera nas paradas. Eles se diferem também na sua localidade de aplicação, enquanto Lima et al. (2012) propõem uma aplicação de uso geral para a sociedade, o referido trabalho sugere uma aplicação voltada para os estudantes dos campi UFC/IFCE.

O sistema web proposto por Freitas e Jaques (2011) propõe uma alternativa para melhorar o transporte público, desse modo, assim como esse este trabalho, traz alternativas que podem vir a simplificar seu dia a dia. Porém, o sistema possui como principal funcionalidade o cálculo de rotas até a parada e entre paradas, enquanto o outro será uma aplicação móvel que tem como principal funcionalidade fornecer informações, como localização do ônibus e tempo de espera.

3 FUNDAMENTAÇÃO TEÓRICA

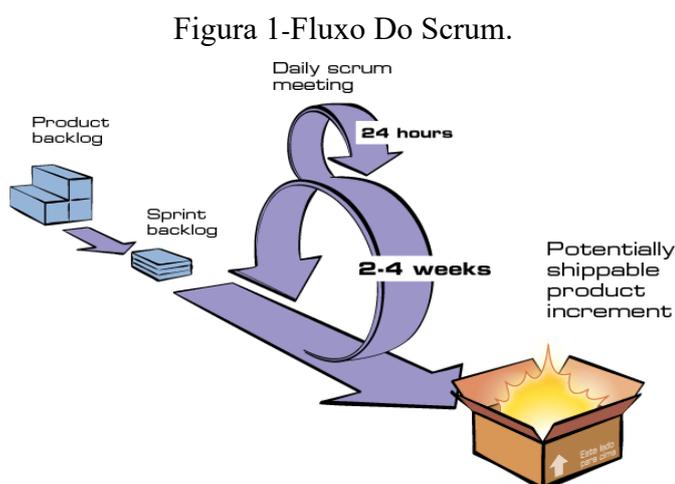
Nesta seção, serão apresentados os principais conceitos necessários para o desenvolvimento deste trabalho.

3.1 *Scrum Solo*

Segundo dados do Ministério do Planejamento do Brasil, em 2015, os desenvolvedores solos de *software* representam aproximadamente 10% da massa produtiva no âmbito de *software*. Para Pagotto et al (2016), esses profissionais carecem de um processo de produção para desenvolvimento de seus trabalhos de conclusão de curso.

O *scrum solo* é um *framework* de gerenciamento de projetos, criado em 2012 pelos desenvolvedores Tiago Pagotto, José Augusto Fabri, Alexandre L'Érario e José Antônio Gonçalves, que une as boas práticas delineadas pelo *Personal/Software Process* (PSP) e pelo *Scrum*. O processo proposto por eles vem para suprir a ausência de uma técnica de desenvolvimento para desenvolvedores solos.

Pagotto et al (2016) define o *scrum* como um *framework* ágil para gerenciamento e planejamento de projetos o qual é composto em *sprints* (geralmente mensais). As *sprints* são ciclos curtos de tempo que são definidos de acordo com a prioridade do *product owner* (proprietário do produto). Com o decorrer de cada *sprint*, o *scrum team* (analista e programadores) faz o possível para ter o conjunto de funcionalidades definidas, para aquela *sprint*, testadas e codificadas (K. SCHWABER, 2004, cap. 1-2). A Figura 1 descreve o fluxo do *Scrum*.



Fonte: PAGOTTO, et al (2016).

Na primeira reunião de planejamento de cada *sprint* (*daily scrum meetings*), onde se encontra o *product owner* e *scrum team*, é definido quais itens do *product backlog*, que são os requisitos ou funcionalidades a serem desenvolvidas no *software*, entrarão no *sprint backlog*, lista de requisitos ou funcionalidades que serão implementadas no *sprint* (K. SCHWABER, 2004, cap. 1-2).

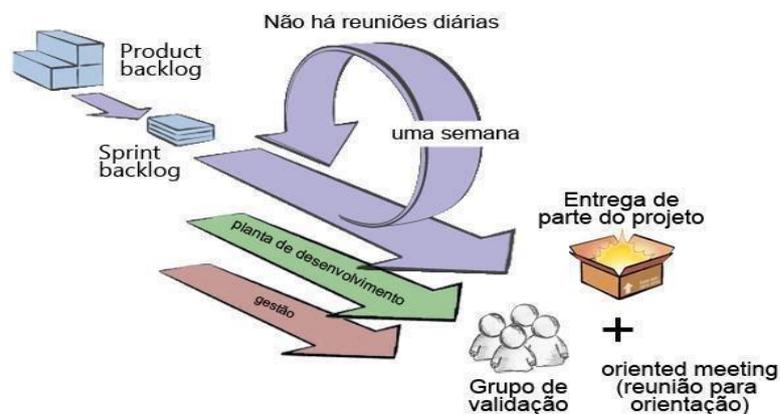
Ao final de cada *sprint*, é feita outra reunião na qual é executado uma revisão e demonstração das funcionalidades produzidas durante a *sprint*, com isso, o *product owner* dá seu *feedback* sobre as demonstrações, o que pode levar a alterações nas funcionalidades entregues (K. SCHWABER, 2004, cap. 1-2).

Pagotto et al (2016) define o PSP como um processo de melhoria individual dos desenvolvedores que fiscaliza, administra e aprimora suas competências para produzir *software* de melhor qualidade. Assim, o intuito do PSP é auxiliar o desenvolvedor a melhorar a sua forma de trabalho, entendendo primeiramente das suas próprias competências e sabendo onde e como melhorá-las.

O pensamento por trás do PSP é que a habilidade individual dos desenvolvedores de uma organização influencia na competência da mesma para construir softwares de determinado tamanho e grau de complexidade. Segundo Sutherland et al (2007), o PSP se baseia no princípio do conhecimento, avaliação e melhoria contínua do processo individual.

Definido tanto o PSP como *Scrum* é possível aprofundar no *Scrum* solo e estabelecer suas diferenças em relação ao *Scrum*, mostrando todas suas atividades, atores e artefatos. Tanto o *product backlog* quanto o *sprint backlog* ocorrem iguais em ambos os *frameworks*. De forma equivalente ao *Scrum*, o desenvolvedor deve entregar um protótipo do *software* com novas funcionalidades e, podem existir, quando necessário, reuniões de orientação entre o grupo de validação (clientes e usuários finais) e o desenvolvedor. No *Scrum* solo, é proposta a redução da duração das *sprints* há uma semana e que não seja necessária uma reunião ao final de cada *sprint* como no *Scrum* (PAGOTTO et al, 2016). A Figura 2 descreve o fluxo do processo *Scrum* solo.

Figura 2- Fluxo do processo Scrum Solo.



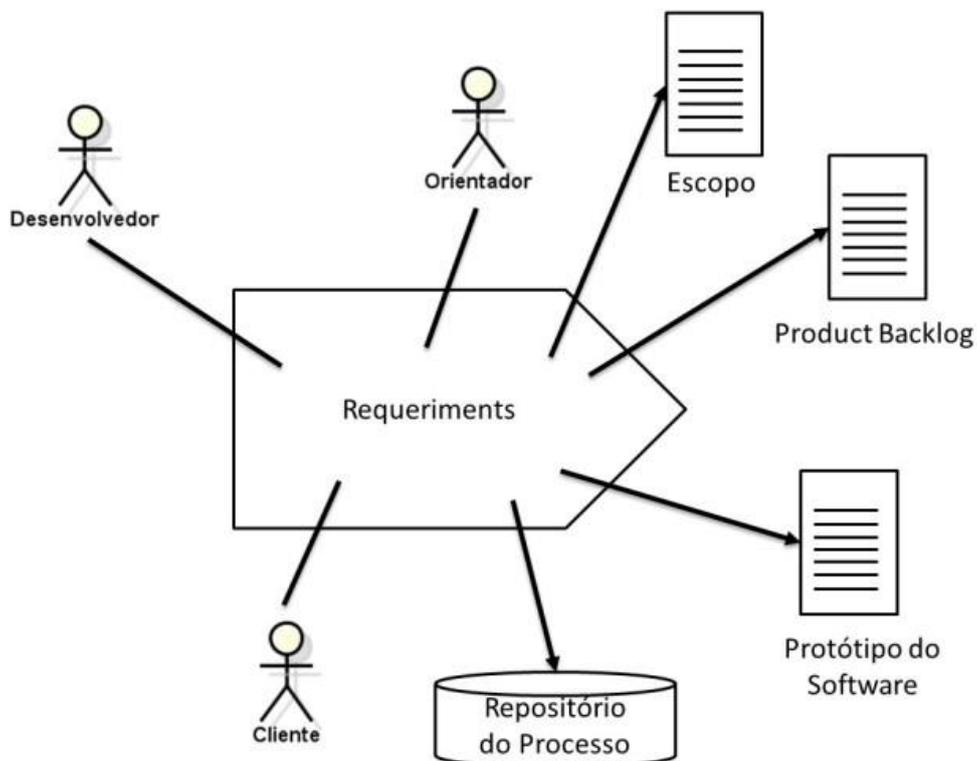
Fonte: PAGOTTO, et al (2016).

A seguir, será descrita a especificação do Scrum Solo relacionando as atividades, atores e artefatos:

3.1.1 Atividade do processo [pagotto, et al (2016)].

- *Requeriment*: objetiva definir o escopo do produto, caracterizar o cliente e definir a *product backlog*.
- Entrada: informações coletadas com o cliente e orientador.
- Saída: escopo, *product backlog* e protótipo de software.

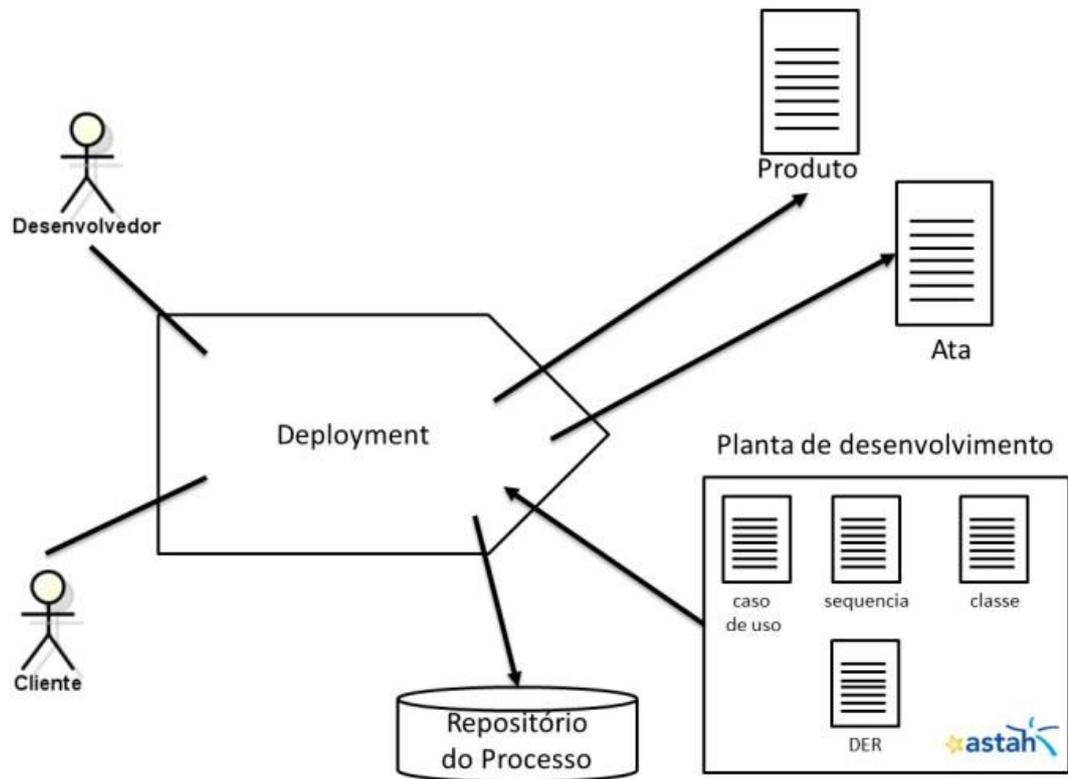
Figura 3- Atividade de requeriment do Scrum Solo



Fonte: PAGOTTO, et al (2016).

- *Sprint*: objetiva desenvolver o conjunto de itens selecionados a partir da *product backlog* em duração máxima de uma semana.
- Entrada: *product backlog* e protótipo de software.
- Saída: *sprint backlog*, produto, ata e planta de desenvolvimento.

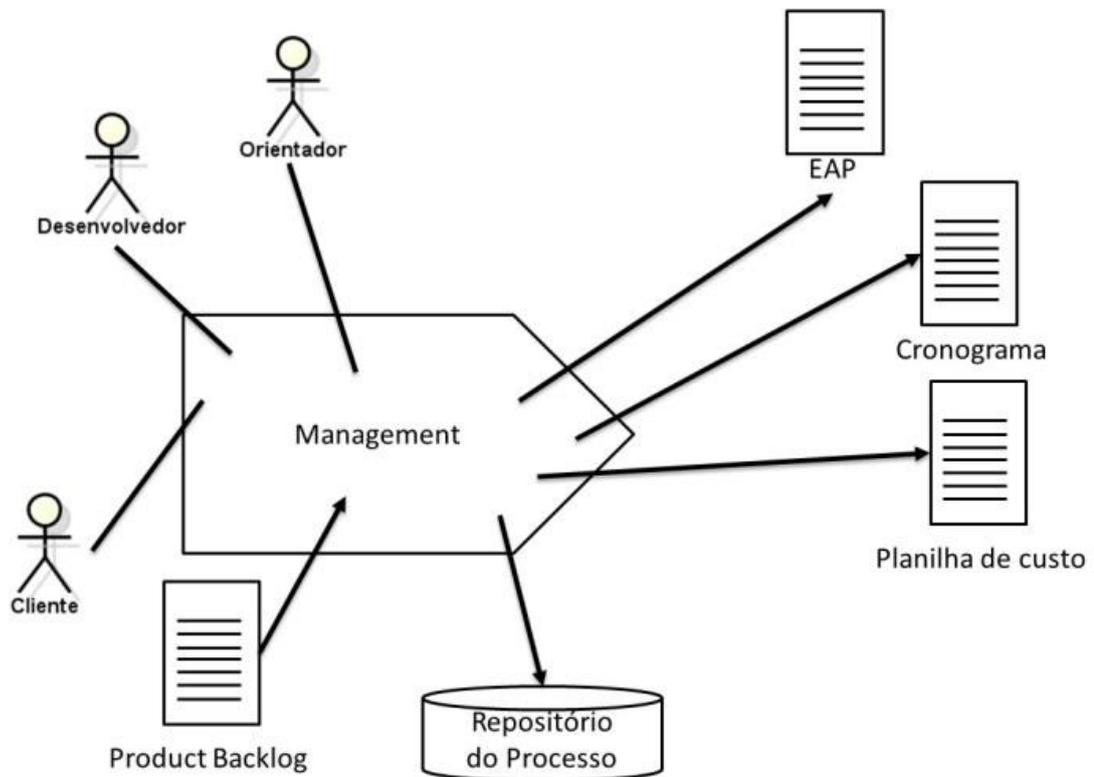
Figura 5 -Atividade de deployment do Scrum Solo.



Fonte: PAGOTTO, et al (2016).

- *Management*: objetiva planejar, monitorar e controlar o desenvolvimento do produto.
- Entrada: *product backlog*.
- Saída: estrutura analítica do projeto (EAP), cronograma, planilha de custo e planilha de controle.

Figura 6- Atividade de Gestão do Scrum Solo



Fonte: PAGOTTO, et al (2016).

3.1.2 Atores do processo [pagotto, et al (2016)].

- *Product owner*: usuário chave do sistema ou pessoa com conhecimento das necessidades dos usuários. Ele irá interagir diretamente com o desenvolvimento do produto e pode ser caracterizado como uma única pessoa: gerente, contador, secretária, ou, como um grupo de usuários, no caso de múltiplos usuários interessados em algum produto.
- Desenvolvedor individual: responsável por executar o processo e construir o produto.
- Orientador: pessoa de vasto conhecimento na área de desenvolvimento do produto, que pode ser requisitada quando forem geradas dúvidas pela parte do desenvolvedor individual, ou para aconselhar no decorrer da conclusão do escopo do produto.
- Grupo de Validação: usuários que farão parte da etapa de validação do produto e que serão os possíveis usuários do sistema sendo os responsáveis diretos pela aprovação ou não do produto.

3.1.3 Artefatos do processo [pagotto, et al (2016)].

- Escopo: especifica o escopo do produto e a questões intrínsecas ao mapeamento dos problemas do product owner. Descreve os principais pontos do software, o perfil do cliente e os itens do *product backlog* (requisitos funcionais).
- Protótipo de Software: acumula as telas da aplicação e modificação de dados do software, além das interfaces dos relatórios.
- *Product backlog*: lista de requisitos funcionais que devem ser implementadas no software. O requisito funcional deve ser restrito, a descrição, a data da inserção e a data de seleção para a *sprint backlog*.
- Repositório do processo: repositório em nuvem onde serão armazenadas todas as produções feitas durante a construção da aplicação.
- *Sprint backlog*: guarda os requisitos funcionais que serão desenvolvidos para determinado sprint. O *sprint backlog* armazena o código do requisito funcional e o link para acesso a planta de especificação daquele requisito data de inserção da funcionalidade no *sprint backlog*, tempo de construção (orçado ou previsto e realizado ou real), data de validação e, por fim, se a funcionalidade precisou de um retrabalho para sua conclusão.
- Produto ou parte dele em funcionamento: a variante do produto que o cliente pode testar para analisar o retorno do seu investimento.
- Ata: documento que objetiva armazenar a validação dos requisitos. É utilizada no *sprint* e na entrega da funcionalidade. Na entrega, é validado pelo cliente enquanto que no *sprint* é validada pelo orientador.
- Planta de desenvolvimento: armazena todos os artefatos que foram gerados durante a execução do projeto. Para essa etapa é recomendável o uso dos diagramas de: casos de uso, sequência, classes e entidade e relacionamento.
- Estrutura Analítica do Projeto (EAP): estrutura do projeto que organiza o mesmo em atividades e as subdivide em outras atividades menores, o que torna o projeto mais fácil de ser gerenciado e também auxilia na elaboração do cronograma e da planilha de custo.
- Cronograma: organiza as atividades de forma sequencial e define o tempo de execução de cada uma. Para o *Scrum* solo é sugerido que se utilize o cronograma no formato de para a produção do cronograma.

- Planilha de custo: mapeia o custo efetivo gerado durante a execução do projeto. Por ela pode ser feita uma análise do orçamento realizado inicialmente e os custos e tempos reais gastos no projeto.

3.2 Location-Based Services

O GPS é um sistema de posicionamento de dimensão mundial em tempo real que informa a posição, a partir de rede de satélites, de um ponto qualquer na superfície terrestre (TECHTUDO, 2014). Com a liberação do GPS pelo Departamento de Defesa dos Estados Unidos, para uso civil na década de 80 (*Global Positioning System*) surgiram diversas aplicações que usavam a localização de alguma pessoa para lhe prover serviços personalizados. Embora a obtenção das coordenadas espaciais de um ponto possa ser considerada, atualmente, uma tarefa fácil, há muito tempo se procurava uma maneira de localização terrestre que substituísse as fontes de orientação, pouco precisas proporcionadas pela orientação do Sol, das estrelas e dos planetas, predominante durante séculos (MONICO, 2008).

Segundo Schiller e Voisard (2004), *Location-based services* (LBS) podem ser definidos como quaisquer serviços com valor agregado oferecidos em um ambiente sem fio que exploram a informação de localização de um terminal móvel. O método que esses serviços usam para ter acesso a localização de determinado dispositivo podem ser divididos em duas categorias que são: centradas na rede (*network centric*) e as centradas no dispositivo (*device centric*).

Para os serviços centrados na rede, a própria rede é quem é responsável por informar a localização de forma automática para o LBS. Um exemplo seria o uso de estações-base para, a partir dela, calcular a posição relativa do equipamento móvel em relação a essa estação. Para os serviços centrados no dispositivo, o próprio aparelho, ou algum dispositivo atrelado a ele, é quem é o responsável por informar sua localização, um exemplo seria o próprio GPS.

3.2.1 Modelo de Interação

Conforme Chen et al (2003), existem dois modelos que fazem o contato entre usuário e os serviços LBS, um iniciado pelo próprio serviço, ou modelo *push*, e o outro pelo cliente, modelo *pull*. Para o iniciado pelo próprio serviço, o servidor é o responsável por enviar as informações. Essas informações são filtradas segundo as preferências definidas no perfil do usuário, e então enviadas de forma automática, dependendo da localização do aparelho do usuário. Supondo que ele use algum serviço LBS para informações de trânsito e de localização

de restaurantes. E esteja dirigindo em uma rodovia onde mais a frente tem um congestionamento. É papel do próprio serviço identificar e avisar ao motorista. Já no modelo *pull*, o cliente faz uma requisição ao servidor.

No modelo *push*, são consumidos muitos recursos de rede por conta da atualização constante da localização dos dispositivos móveis, o que é um problema para a aplicação. Além disso, o aumento de perfis de usuários usando essa aplicação causa um crescimento no número de dados relacionados à preferência de cada usuário, que está usando a aplicação, o que pode ser difícil de gerenciar.

4.2.2 Localização Simbólica

Uma localização simbólica é a reprodução coordenada em um LBS, como nomes, endereços e números. Para regiões simbólicas que possuem alguma relação, como um prédio e cada um de seus cômodos surge o problema de como deve ser representado esse relacionamento.

Gonçalves (2005) propõe uma solução de usar uma hierarquia de áreas. Desse modo, quando uma área pai receber alguma mensagem postada nela, as áreas filhas também receberão essa mensagem. O problema surge quando pensarmos que o último filho dessa hierarquia vai receber uma grande quantidade de dados, dados esses, que mal gerenciados poderão acarretar algum problema para a aplicação. Como forma de contornar esse problema, pode-se optar por usar uma hierarquia plana, de modo que cada cômodo tenha seu próprio nome. Entretanto, essa estratégia pode causar alguns problemas relacionados à grande quantidade de localidades que deverão ter nomes exclusivos.

3.2.3 Posição x localização

É comum a não distinção desses termos e para entendermos os conceitos gerais sobre LBS é de suma importância diferenciá-los. Localização está relacionada ao conceito de região simbólica. Frequentemente, a posição de um dispositivo móvel é informada por tecnologias de localização, para isso, é necessário um mapeamento para região simbólica a partir das coordenadas informadas por essas tecnologias de localização. Já a posição está relacionada com as coordenadas geográficas de um ponto da superfície terrestre.

3.2.4 Perfis de usuário

É necessário, nas aplicações LBS, armazenar atributos que retratam informações sobre o usuário. Para isso, Gonçalves (2005) propõe o uso de entidades, denominada perfis de usuários, que possuem como finalidade reunir essas informações. Essas informações podem ser tanto estáticas como dinâmicas. As estáticas são aquelas em que não receberam alterações no decorrer da aplicação, como nome, idade etc. Nas informações dinâmicas, a variação e alterações dos dados são constantes, como localização, status de alcançabilidade etc.

Conforme Leung et al (2003), além das informações sobre os perfis dos usuários, outros tipos de dados podem ser comumente encontrados nas aplicações para LBS. As informações geográficas estáticas são informações, no contexto geográfico, a qual sua modificação não é frequentemente alterada, como localização de pontes, estradas, prédios etc. Conteúdo dinâmico é um tipo de informação que depende da aplicação, essas informações são dinâmicas e mudam constantemente. Por exemplo, condições do tempo ou do tráfego em uma estrada. Informação da posição dos aparelhos móveis é informada pelas tecnologias de localização e são coordenadas físicas do dispositivo. É uma informação dinâmica que, dependendo da precisão da tecnologia, pode ser frequentemente atualizada.

4 PROCEDIMENTOS METODOLÓGICOS

Nesta seção, serão apresentados os passos necessários para a execução deste trabalho.

4.1 Tipos do Estudo

A primeira parte do trabalho trata-se de um projeto de pesquisa aplicada e apresenta uma abordagem quantitativa. Caracterizou-se como sendo um estudo transversal e descritivo, de acordo com Gil (2008), as pesquisas descritivas possuem como objetivo a descrição das características de uma população, fenômeno ou de uma experiência. A segunda etapa do trabalho foi voltada para o desenvolvimento do aplicativo PASTORABUS onde também foram executados os testes e correções de bugs da aplicação.

4.2 Coletas de Dados

A coleta de dados consistiu em duas etapas. A primeira referente à elaboração de um questionário de diferentes tópicos, tendo 11 questões com objetivo de identificar as principais necessidades dos usuários (ver Seção 5.1). O questionário foi aplicado através da plataforma gratuita Google Forms.

Na segunda etapa houve a coleta de informações sobre ferramentas já existentes no mercado, que possuem funcionalidades semelhantes à aplicação PASTORABUS. Com a conclusão dessa etapa foi feita uma análise sobre essas ferramentas.

4.3 Análise de Dados

Os dados do questionário (ver Seção 5.1) foram observados e analisados para auxiliar no desenvolvimento da ferramenta. A análise das ferramentas já existentes no mercado que possuem funcionalidades semelhantes à aplicação proposta serve para se ter uma maior compreensão de como o mercado de aplicativos está organizado em relação ao transporte público. Levando em conta sua avaliação e sua semelhança com o aplicativo proposto, sete plataformas foram escolhidas com objetivos de execução semelhantes para a análise. Em seguida, foi produzido um relatório detalhado contendo informações a respeito das plataformas analisadas (ver Seção 5.2).

Os dados do questionário foram interpretados em formas de gráficos fornecidos pela plataforma Google Forms e posteriormente analisados. Enquanto, na análise das ferramentas, foi desenvolvida uma análise escrita seguida de uma tabela com suas principais funcionalidades.

4.4 Desenvolvimento do Aplicativo PASTORABUS

Nessa etapa, foram executadas as *sprints* e, dessa forma, o desenvolvimento das funcionalidades propostas. Foram usados os serviços LBS para as funcionalidades de localização em tempo real do ônibus e cálculo estimado entre paradas.

É importante ressaltar, que o aplicativo foi desenvolvido para plataforma Android e que, na camada de *framework*, onde pode-se encontrar uma variedade de APIs, existe a *LocationAPI*, fornecida pela *Google Play Services*, que em conjunto com a biblioteca externa do Google Maps, foi utilizada para fornecer a localização e o mapa do aplicativo.

4.5 Etapa de Testes

Após a conclusão das *sprints* de desenvolvimento, foi dado início a etapa de teste do software. Para isso, fornecemos o produto para uso por um grupo de validação, que compreendem um conjunto dos alunos dos campi UFC/IFCE.

Foi disponibilizado a ata de validação para que os participantes dos testes pudessem validar as funcionalidades. Para as funcionalidades que não foram aprovadas, foi feito um retrabalho na etapa de correção de bugs.

4.6 Melhorias e Correções de Bugs

Com os dados do questionário, a respeito do *feedback* do grupo de validação, pode-se identificar possíveis melhorias e erros no produto disponibilizado. Dessa forma, foi dado início à etapa de melhorias e correções de *bugs* do sistema. Para então disponibilizar a primeira *realase* do aplicativo para uso dos alunos dos campi UFC/IFCE.

5 DESENVOLVIMENTO DO APLICATIVO PASTORABUS

Nesta seção, serão apresentados os passos para o desenvolvimento do aplicativo PASTORABUS. O código fonte do aplicativo está em: github.com/rananaraujo/PASTORABUS.

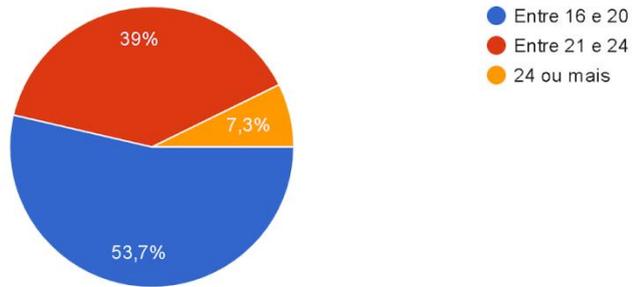
5.1 Questionário

Um questionário de 11 questões foi elaborado com o objetivo de identificar as principais necessidades dos usuários. O questionário foi aplicado através da plataforma gratuita Google Forms. No total, 41 estudantes responderam as 11 questões, onde 69,3% são acadêmicos da UFC e 31,7% do IFCE, ambos do campus Quixadá – CE. 53,7% e possuem idade entre 16 e 20 anos, 21 e 24 anos representa 39% dos entrevistados e o menor percentual para pessoas com mais de 24 anos com 7,3%. O Gráfico 1 mostra o percentual de idade dos usuários do ônibus escolar acadêmico e o Gráfico 2 mostra o percentual de acadêmicos dos campi UFC/IFCE.

Gráfico 1 - Percentual de idade dos usuários do ônibus escolar acadêmico

Qual a sua idade?

41 respostas

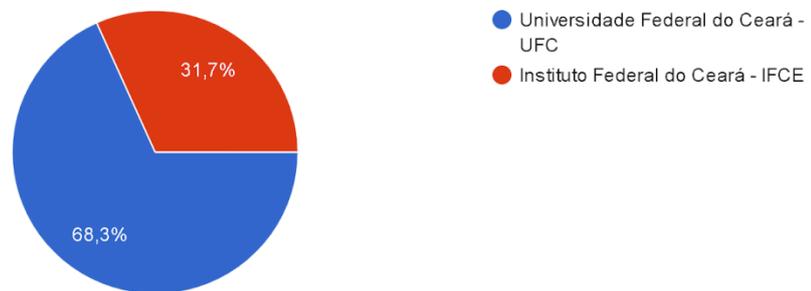


Fonte: elaborado pelo autor

Gráfico 2 - Percentual de acadêmicos dos campi UFC/IFCE

Onde estuda?

41 respostas



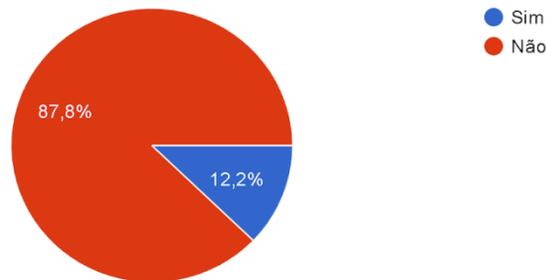
Fonte: elaborado pelo autor

Através do questionário, é possível destacar que apenas 12,2% do alunato não dependem do transporte universitário como meio de acesso às instituições. Enquanto 87,8% dos alunos dependem diretamente do uso do transporte. O Gráfico 3 mostra o percentual de usuários com transporte próprio fazendo o percurso centro Quixadá - campus UFC/IFCE.

Gráfico 3 - Percentual de usuários com transporte próprio fazendo o percurso.

Possui algum meio transporte próprio para fazer o percurso "centro Quixadá - campus UFC/IFCE"?

41 respostas



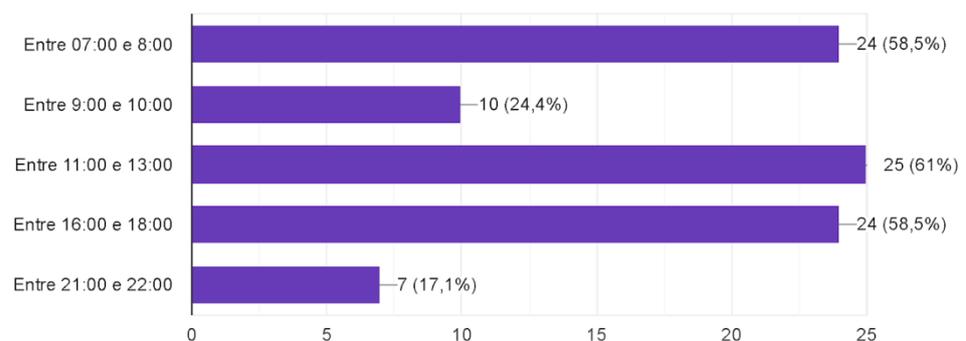
Fonte: elaborado pelo autor

Os horários de maior fluxo são às 7h/8h e 16h/18h, ambos com 58,5% e entre as 11h e 13h com 61%. Em uma das questões levantadas é possível notar como o transporte universitário pode influenciar no rendimento do aluno. Onde mais de 60% dos pesquisados responderam o que já foram prejudicados com faltas ou atrasos em aulas por conta do transporte universitário. O Gráfico 4 mostra os horários de maior fluxo do ônibus universitário e o Gráfico 5 mostra como a falta de qualidade do transporte universitário pode prejudicar no rendimento do aluno.

Gráfico 4 - Os horários de maior fluxo do ônibus universitário.

Quais horários que mais utiliza o ônibus? (Escolha até duas opções)

41 respostas

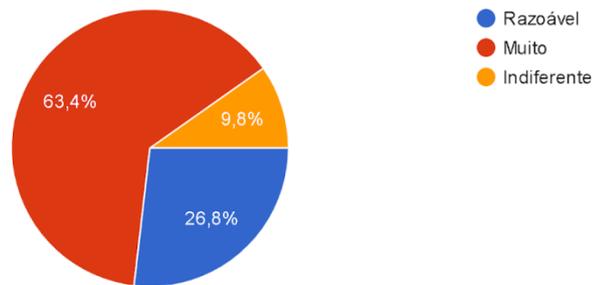


Fonte: elaborado pelo autor

Gráfico 5 - A falta de qualidade do transporte universitário pode prejudicar no rendimento do aluno.

Já foi prejudicado com faltas e/ou atrasos em aulas por conta do transporte universitário (centro Quixadá - Campus UFC/IFCE)?

41 respostas



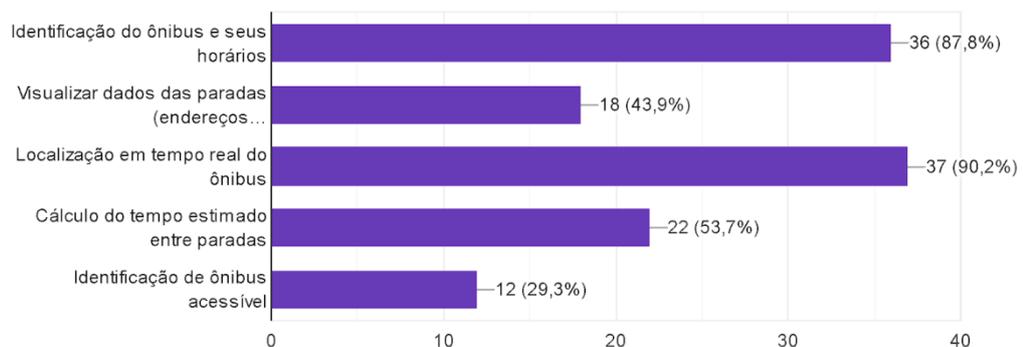
Fonte: elaborado pelo autor

Algumas funcionalidades foram listadas para o usuário identificar quais delas ele gostaria que a ferramenta disponibilizasse. A que mais atraiu os usuários refere-se à localização em tempo real do ônibus com 90,2%, seguida pela identificação do ônibus e seus horários com 87,8%. Diante do exposto, as funcionalidades se mostraram de suma importância para serem implementadas no sistema proposto. O Gráfico 6 mostra a lista de funcionalidades e o percentual de aprovação delas por cada usuário.

Gráfico 6- Lista de funcionalidades e o percentual de aprovação delas por cada usuário.

Dado a lista de funcionalidades abaixo o que você gostaria que o aplicativo possuísse?

41 respostas



Fonte: elaborado pelo autor

5.2 Análise de Ferramentas

A seguir, são analisadas ferramentas já existentes no mercado. Essas ferramentas foram escolhidas levando em conta sua avaliação e sua semelhança com o aplicativo proposto.

5.2.1 *Transit*

O aplicativo Transit⁴ foi desenvolvido pela empresa Québec Inc em Montreal, Quebec. O aplicativo está disponível em mais de 175 cidades nos continentes da América do Norte e Europa. Está disponível para as plataformas IOS e Android. Disponibilizando um sistema em tempo real, planejamento de rotas, navegação passo a passo, alarme de desvio de rotas, horários de saída e lembretes de chegada. O Transit é uma ferramenta com suporte para os seguintes meios de transporte: ônibus, metrô, trem, carros de rua, Uber, bike colaborativa e transporte de navegação fluvial.

Especificações técnicas:

- Versão 5.4.14.
- Tamanho: 12,36 MB.
- Lançamento em 29 de julho de 2013.
- Classificação livre.

5.2.2 *Moovit*

O aplicativo Moovit⁵ foi criado pela Tranzmate, uma start-up israelense. O aplicativo está disponível em mais de 88 países, entre eles o Brasil, e em 44 idiomas. O Moovit disponibiliza aos passageiros informações como chegadas em tempo real, alertas (engarrafamentos, obras, atrasos etc.), passo a passo de rotas, compartilhamento de informações entre usuários, além de rotas de bicicleta para países que possuem ciclovias. O Moovit é uma ferramenta para os seguintes meios de transportes: ônibus, metrô, trem, bicicletas e Uber. O aplicativo está disponível para as plataformas IOS, Android e Open the web app.

⁴ <https://play.google.com/store/apps/details?id=com.thetransitapp.droid>

⁵ <https://play.google.com/store/apps/details?id=com.tranzmate>

Especificações técnicas:

- Versão: 5.36.5.422.
- Tamanho: 11.01 MB.
- Lançamento em 28 de março de 2012.
- Classificação: livre.

5.2.3 *Citymapper*

O Citymapper⁶ está disponível para as plataformas IOS, Android e Open the web app. Desenvolvida pela Citymapper Limited, a ferramenta possui cobertura internacional. É um aplicativo destinado para ônibus, metrô, trens metropolitanos, bicicletas e Uber. Possui passo a passo de rotas, localização em tempo real do ônibus, alertas (engarrafamentos, obras, atrasos etc.), além de fornecer a localização do passageiro para sua rede de amigos.

Especificações técnicas:

- Versão: 8.11.
- Tamanho: 16,69 MB.
- Lançamento em 11 de abril de 2013.
- Classificação: livre.

5.2.4 *Cadê o Ônibus?*⁷

Este aplicativo atende as cidades de São Paulo e região, Teresina, Rio de Janeiro (Capital) e Curitiba. Com essa ferramenta, é possível pesquisar linhas de ônibus e observá-las. Os usuários podem visualizar o melhor trajeto e ter acesso ao mapa metropolitano da cidade, além de colaborar com outros usuários com informações sobre o ônibus. Foi desenvolvido pela empresa Nano IT Brasil e está disponível nas plataformas IOS, Android e Windows Phone.

Especificações técnicas:

⁶ <https://play.google.com/store/apps/details?id=com.citymapper.app.release>

⁷ <https://play.google.com/store/apps/details?id=solucoes.zeno.sinetram>

- Versão: 6.1.3.
- Tamanho: 18,69 MB.
- Lançamento em 14 de maio de 2012.
- Classificação: livre.

5.2.5 *Cittamobi*

O aplicativo CittaMobi⁸ informa a previsão de chegada do ônibus em tempo real e está disponível em mais de 70 cidades no território nacional. A ferramenta mostra a melhor rota de um ponto ao outro, exibe itinerários das linhas de ônibus e permite a comunicação entre os usuários. Também é possível a visualização de trechos a pé, trem, ônibus ou metrô. Nas cidades de São Paulo, Diadema, Ribeirão Preto, Guarulhos e Recife é possível fazer a recarga do cartão de transporte público.

Especificações técnicas:

- Versão: 6.17.10.
- Tamanho: 9,37 MB.
- Lançamento em 31 de janeiro de 2014.
- Classificação livre.

5.2.6 *Meu ônibus - fortaleza.*⁹

Desenvolvido pela M2M SOLUTIONS o Meu Ônibus - Fortaleza é um aplicativo que fornece previsões para que o ônibus chegue em determinado ponto, e uma lista de todas as paradas que estão registradas no aplicativo. A ferramenta também dispõe da opção para recarregar o bilhete único de transporte. Está disponível unicamente para a cidade de Fortaleza e para as plataformas IOS e Android.

⁸ <https://play.google.com/store/apps/details?id=br.com.cittabus>

⁹ <https://play.google.com/store/apps/details?id=br.com.m2m.meuonibus>

Especificações técnicas:

- Versão: 4.4.
- Tamanho: 9,6 MB.
- Lançamento em 14 de agosto de 2015.
- Classificação livre.

5.2.7 Google maps go.

Google Maps Go ¹⁰é uma versão mais otimizada do Google Maps. Disponível em mais de 70 idiomas, 200 países e 20.000 cidades, o aplicativo fornece informações de tráfego em tempo real, passo a passo de rotas e informações sobre o transporte público (Ônibus, metrô e trem). Além das informações sobre mobilidade urbana, a ferramenta conta com um grande arsenal de informações sobre pontos turísticos, restaurantes e empresas.

Informações técnicas:

- Versão: 98.
- Tamanho: 175 KB.
- Lançamento em 6 de fevereiro de 2018.
- Classificação: livre.

A seguir, a Figura 7 mostra uma análise das ferramentas já existentes no mercado, destacando quais funcionalidades elas possuem, seu tamanho para *download* e sua avaliação, segundo seus usuários.

¹⁰ <https://play.google.com/store/apps/details?id=com.google.android.apps.mapslite>

Figura 7- Análise das ferramentas já existentes no mercado.

FERRAMENTA	TAMANHO/ AVALIAÇÃO	IDENTIFICAÇÃO DO ÔNIBUS E SEU HORÁRIO	VISUALIZAR DADOS DAS PARADAS	LOCALIZAÇÃO EM TEMPO REAL DO ÔNIBUS	CÁLCULO DO TEMPO ESTIMADO ENTRE PARADAS	IDENTIFICAÇÃO DE ÔNIBUS ACESSÍVEL
	12,36 MB/4,8	x	x	x	x	-
	11,01 MB/4,5	x	x	x	x	-
	16,69 MB/4,5	x	x	x	x	-
	18,69 MB/2,5	x	x	-	-	x
	9,37 MB/3,7	x	x	x	x	-
	9,6 MB/1,9	x	x	-	x	-
	175 KB/4,2	x	x	x	x	-

Fonte: elaborado pelo autor

5.3 Execução das Atividades Utilizando o Scrum Solo

Nesta seção, é demonstrado o processo da execução das atividades do *Scrum Solo*, relacionando todos os artefatos de entrada do projeto com os artefatos de saída, gerados na execução do *Scrum Solo* com o desenvolvimento do aplicativo PASTORABUS. É importante ressaltar que o template da documentação utilizada no projeto foi disponibilizado previamente pelos autores do *scrum solo* (Pagotto, Fabri, L'Erario e Gonçalves).

5.3.1 Requirement

A primeira atividade do *Scrum Solo* foi a de *requirement*. Durante sua execução, o objetivo era definir o escopo do produto, caracterizar o cliente e definir o *product backlog* para que posteriormente esses documentos fossem usados nas outras atividades do processo.

O cliente foi caracterizado como os estudantes dos campi Instituto Federal do Ceará (IFCE) e a Universidade Federal do Ceará (UFC - Campus Quixadá). Em uma pesquisa previamente realizada, tornou-se possível identificar que a maioria dos estudantes faz uso do transporte universitário para a locomoção até a faculdade.

Com o aplicativo, será possível visualizar informações sobre os horários previstos para determinado ônibus passar em uma parada do trajeto, acessibilidade do ônibus, consulta das paradas, que compõem o percurso e a localização dos ônibus em tempo real para os usuários do sistema por meio do compartilhamento da localização. Dessa forma, será possível saber onde

o ônibus está e em quanto tempo o mesmo estará na parada. Será inicialmente disponível para o sistema Android e futuramente para IOS.

Definidos tanto o cliente da aplicação como o escopo do projeto foi possível definir o *product backlog*. O Quadro 2 apresenta a definição do *product backlog* do projeto.

Quadro 2 - Product backlog do aplicativo PASTORABUS.

ID	Descrição	Link para arquivo digital contendo a gravação (requisitos custodiais)	Data da inserção	Data de seleção para a Sprint
1	Mapear Rotas (Mapear paradas)	-	05/02/2020	16/03/2020
2	Confirmação de compartilhamento	-	05/02/2020	23/03/2020
3	Verificar localização compartilhada	-	05/02/2020	23/03/2020
4	Visualizar os dados das paradas	-	10/02/2020	20/04/2020
5	Compartilhamento em segundo plano	-	10/02/2020	06/04/2020
6	Cadastro de paradas	-	10/02/2020	20/04/2020
7	Identificar posição do ônibus	-	10/02/2020	27/04/2020
8	Mapear parada e posição do ônibus	-	10/02/2020	18/05/2020
9	Cadastro dos ônibus	-	10/02/2020	11/05/2020
10	Consultar ônibus	-	10/02/2020	11/05/2020
11	Informações sobre o APP	-	10/02/2020	01/06/2020

Fonte: elaborado pelo autor

Cada item do *product backlog* foi armazenado com um ID único para sua identificação, uma descrição sobre a funcionalidade a ser implementada, a sua data de inserção no documento do *product backlog* e sua data de seleção para a *sprint*. Para nenhuma dessas funcionalidades, tornou-se necessário armazenar o link para arquivo digital contendo a gravação (requisitos custodiais).

É importante ressaltar que alguns itens foram selecionados para sua respectiva *sprint* na mesma data de outros itens. Isso é feito baseado na dependência e semelhança das funcionalidades com alguma outra.

5.3.2 *Sprint*

Na segunda atividade do *Scrum Solo*, foi desenvolvido o conjunto de itens que foram criados a partir do *product backlog* e do protótipo do *software*, disponível no repositório da aplicação. O Quadro 3 mostra as ferramentas que foram usadas para o desenvolvimento da aplicação na atividade de *sprint*.

Quadro 3 - Ferramentas utilizadas para o desenvolvimento da aplicação PASTORABUS

Nome	Tipo	Versão	Descrição
Android Studio	IDE	3.4.1	Ferramenta para codificação
Git	Sistema de controle de versão	2.24.0	Ferramenta para versionamento e gerenciamento do código
Lockito GPS itinerary faker	APP	3.2.1	Programa para simular a rota do ônibus
LocationAPI	API	Não identificado	API para obter localização do ônibus
Google Maps API	API	Não identificado	Incorpora o mapa que será usado no APP

Fonte: elaborado pelo autor

É importante ressaltar, que por indicação dos autores do *Scrum Solo* (Pagotto, Fabri, L'Erario e Gonçalves), as sprints teriam prazo de conclusão de uma semana. Entretanto pela complexidade de alguns desses itens, algumas *sprints* do projeto PASTORABUS receberam duas semanas de prazo, que representariam 2 *sprints* do *Scrum Solo* para conclusão.

O Quadro 4 mostra como ficou organizado o *sprint backlog* da aplicação PASTORABUS.

Quadro 4 - Sprint backlog da aplicação PASTORABUS.

ID	Descrição	Data de inserção	Tempo de construção (em horas)		Data da finalização	Data de validação	Retrabalho (sim ou não)
			orçado	realizado			
1	Mapear Rotas (Mapear paradas)	16/03/2020	40	32	19/03/2020	17/07/2020	Sim
2	Confirmação de compartilhamento e verificar localização compartilhada	23/03/2020	80	80	03/04/2020	17/07/2020	Sim
3	Compartilhamento em segundo plano	06/04/2020	80	80	17/04/2020	13/07/2020	Não
4	Cadastro e visualização de dados da parada	20/04/2020	40	40	24/04/2020	17/07/2020	Sim
5	Identificar posição do ônibus	27/04/2020	80	64	06/05/2020	17/07/2020	Sim
6	Cadastro dos ônibus e Consulta de ônibus	11/05/2020	40	32	14/05/2020	17/07/2020	Sim
7	Mapear parada e posição do ônibus	18/05/2020	80	72	28/05/2020	13/07/2020	Não
8	Informações sobre o APP	01/06/2020	40	40	05/06/2020	17/07/2020	Sim
9	Etapa de testes	06/07/2020	40	40	10/07/2020	10/07/2020	Não
10	Correções de bugs	13/07/2020	40	40	17/07/2020	17/07/2020	Não

Fonte: elaborado pelo autor

O primeiro item da tabela é o ID, é indispensável evidenciar, que como algumas funcionalidades do *product backlog* foram desenvolvidas de formas conjuntas no *sprint backlog* o seu ID acabou sendo alterado. O segundo item é a descrição, nele é possível reparar a junção das funcionalidades do *product backlog* em uma única sprint.

A “Data de inserção” equivale ao dia em que a funcionalidade selecionada do *product backlog* para o *sprint backlog*. O tempo de construção da sprint ficou orçado em horas.

Primeiramente, foi estabelecido um prazo para a conclusão da *sprint*, baseado na sua complexidade de construção e depois o tempo real que foi gasto para aquela *sprint*.

Nessa atividade se gerou a ata da *sprint*, que corresponde, na tabela, a data de validação, ela é responsável por armazenar a data que a funcionalidade foi aprovada pelo orientador e consequentemente definir se teve retrabalho ou não naquela *sprint*.

Nas próximas seções, é relatado a execução de cada *sprint* realizada no desenvolvimento do aplicativo PASTORABUS, detalhando os conceitos utilizados para a sua produção e a experiência do desenvolvedor individual do *Scrum* Solo no desenvolvimento.

5.3.2.1 Primeira sprint: Mapear Rotas (Mapear Paradas)

Nesta Sprint, executou o mapeamento das rotas que compõem o percurso Praça José de Barros/IFCE e IFCE/Praça José de Barros. Assim, foram identificadas todas as coordenadas geográficas que compõem o percurso, a fim de obter informações que são úteis para as outras *sprints*.

Também, neste sprint, identificou-se a coordenada geográfica referente a cada parada. Essa informação será utilizada para calcular o tempo estimado entre a posição em tempo real do ônibus e a parada e, assim, fornecer uma informação mais precisa de quanto tempo o ônibus estará naquela parada.

Com uma análise mais detalhada da carga de trabalho do requisito, estimou-se um prazo de 40h, uma semana de trabalho, para a conclusão, tendo início no dia 16/03/2020 e término no dia 20/03/2020. Entretanto, 96 horas (4 dias) foram necessárias para sua conclusão. Dessa forma, foi concluída no dia 19/03/2020 e validada no dia 17/07/2020, na atividade do processo *Deployment* e foi necessário passar por retrabalho.

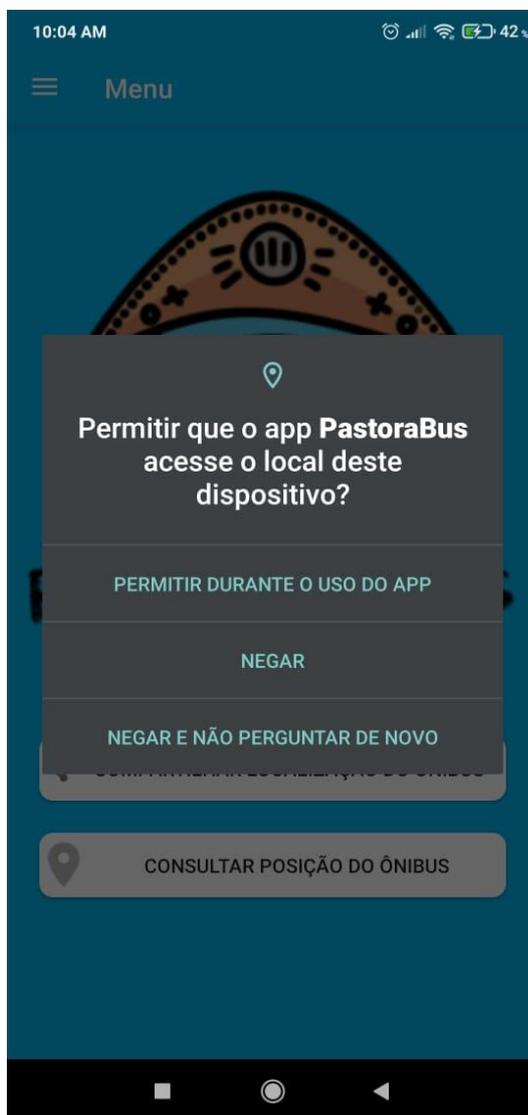
5.3.2.2 Segunda sprint: Confirmação de compartilhamento e verificar localização compartilhada

O Google Developers (2020) define uma sequência de passos necessários para que seja possível solicitar o acesso à localização do usuário. Essa solicitação é necessária como uma forma de proteger a privacidade do usuário que usam os aplicativos com serviços de localização.

Na aplicação PASTORABUS, essa solicitação foi utilizada para acessar a localização do usuário, onde foi criado um método para verificar se o aparelho do usuário possui as

permissões necessárias para compartilhar sua localização com o aplicativo. A Figura 8 mostra como ficou a tela de permissão de compartilhamento da aplicação PASTORABUS.

Figura 8 – Tela de permissão de compartilhamento PASTORABUS



Fonte: elaborado pelo autor

Para acessar a localização em tempo real do ônibus, foi utilizada a *LocationAPI*, fornecida pela *Google Play Services* junto com a biblioteca externa do Google Maps. A *LocationAPI* possibilitou acessar a localização de um usuário em um tempo constante enquanto o aplicativo está em primeiro plano. O problema surge quando se quer acessar a localização do usuário em segundo plano.

Além da necessidade de permissão de compartilhamento de localização com o aplicativo, é necessária uma forma de evitar que o usuário do PASTORABUS compartilhasse uma localização que está fora da rota, o que poderia comprometer com o resultado da posição em tempo real do ônibus.

Usando a rota mapeada anteriormente do percurso do ônibus, foi desenhado um conjunto de círculos com raio de 50 metros por todo o percurso. Esses círculos foram usados para identificar as localizações compartilhadas que estão dentro das rotas.

Ao identificar as localizações que estão dentro da rota a aplicação salva em um banco de dados real time *firebase* essa posição, passando a latitude e a longitude junto com horário, em milissegundos, de seu compartilhamento. As posições salvas posteriormente serão utilizadas para identificar a posição do ônibus. A Figura 9 mostra como ficou o mapa após o traçado dos círculos.

Figura 9 - Conjunto de círculos com raio de 50 metros por todo o percurso.



Fonte: elaborado pelo autor

Esta *sprint* possui uma carga maior de requisitos complexos o que demanda uma maior quantidade de tempo para sua implementação e, por esse motivo, estimou-se um prazo de 80 horas (2 *sprints*), que totalizaram duas semanas de trabalho para a sua conclusão tendo início no dia 23/03/2020 e término no dia 03/04/2020. Todo o tempo previsto foi usado para a

conclusão da *sprint*. Para essa Sprint, surgiu a necessidade de um retrabalho, que é definido na atividade do processo de Deployment.

5.3.2.3 Terceira sprint: Compartilhamento em segundo plano

Para acessar a localização do usuário em segundo plano, a aplicação PASTORABUS, em conjunto com a API *Fused Location Provider* (FLP), recebe a localização em segundo plano de duas formas distintas. Primeiro, o app é executado em segundo plano, a *LocationAPI* computará um novo local para ele apenas algumas vezes por hora. Isso acontece mesmo quando o aplicativo está solicitando atualizações de local de forma mais frequente. Segunda forma, usando o FLP na sua versão em lote, pode-se acessar a um histórico de localização de forma mais frequente após o aplicativo receber uma atualização em lote, o que também ocorre apenas algumas vezes por hora.

Essas duas formas de acesso se mostraram ineficientes para a aplicação PASTORABUS, pois o recebimento de localização apenas em algumas vezes por hora não é o ideal para uma aplicação que necessita dessa informação quase que em tempo real.

O Google Developers (2020) define que na tentativa de minimizar o uso da bateria os dispositivos que usam o Android 8.0 (API de nível 26) diminuem a frequência de atualização da localização em segundo plano e assim os aplicativos podem receber atualizações de local apenas algumas vezes por hora.

Essa terceira Sprint do projeto teve duração de 80 horas, duas semanas de trabalho, o mesmo tempo que foi planejado para sua conclusão. Esse prazo foi definido baseado na dificuldade de encontrar formas eficientes de compartilhamento de localização em tempo real em segundo plano.

5.3.2.4 Quarta sprint: Cadastro e visualização dos dados de paradas

Na etapa de cadastro de paradas, foram registrados todos os pontos de espera que compõem o percurso do ônibus universitário. As informações que foram salvas sobre essas posições são: fotos do local, informação sobre a sua posição (cruzamento ou local exato da parada), latitude e longitude referente ao local. A imagem 10 mostra como ficou a tela de visualização dos dados das paradas.

Figura 10 - Tela de visualização dos dados das paradas.



Fonte: elaborado pelo autor

As informações foram utilizadas para a visualização dos locais por parte do usuário. Isso possibilitou repassar para ele as paradas que fazem parte do percurso e deixá-lo definir qual ponto é mais adequada para a espera do transporte universitário.

A seleção de um ponto previamente determinado pelo usuário proporcionou calcular a rota do ônibus até o ponto de espera e desenhar o percurso no mapa e, desse modo, calcular o tempo estimado entre esses pontos.

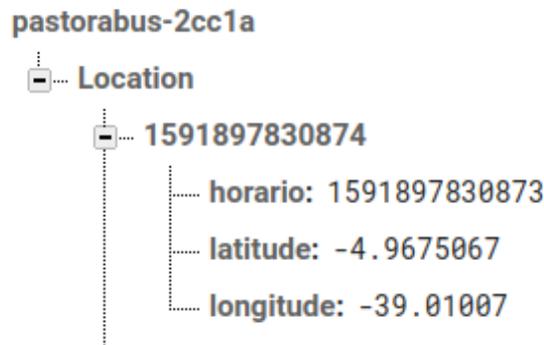
No total, cinco pontos de espera compõem o percurso do ônibus universitário. Como a quantidade de informações a serem armazenadas é pequena, juntamente com a quantidade reduzidas de paradas, planejado em 40 horas (uma semana / uma Sprint) para a conclusão dessa funcionalidade. Todo o tempo previsto foi usado para a conclusão da funcionalidade. Para esta

Sprint, foi necessário um retrabalho o que será detalhado na atividade do processo de Deployment.

5.3.2.5 Quinta sprint: Identificar posição do ônibus

Para identificar a posição do ônibus, é seguida uma sequência de passos lógicos usando os dados que foram armazenados pelo verificador de localização compartilhada no Banco de dados da aplicação. Lembrando que esses dados foram armazenados da seguinte forma como mostra a Figura 9.

Figura 11- Modelo de armazenamento de dados das posições compartilhadas.



Fonte: elaborado pelo autor

Todas as informações que estão armazenadas no banco de dados são baixadas pela aplicação e inseridas em uma lista A. Dentro dessa lista, são selecionadas as posições que foram compartilhadas no intervalo do horário atual até o horário atual menos 5 segundos e salvos em uma nova lista B.

Como uma forma de amenizar que um usuário que esteja dentro da rota compartilhe uma posição que não é a real posição do ônibus, foi feito o cálculo da mediana tanto para as latitudes como para a longitude que se encontram na lista B. O resultado desse cálculo é identificado como a posição do ônibus.

Caso nenhuma posição salva no banco de dados esteja contida no intervalo mencionado, é retornado uma mensagem para o usuário que o ônibus está fora do horário e é solicitando que ele verifique o itinerário do ônibus

Por fim, um método executado, em um tempo constante de 10 segundos, pela aplicação repete todo esse processo sempre atualizando a nova posição do ônibus.

Esta etapa foi selecionada como uma das mais complexas da aplicação, demandando um maior prazo para a sua conclusão. Assim, foram estabelecidas 80 horas, duas sprints equivalentes a duas semanas de trabalho, para a conclusão dessa funcionalidade. Desse prazo estabelecido, apenas 64 horas, 8 dias de trabalho, foram necessários para sua conclusão. Este sprint passou por um retrabalho, o qual será detalhado nas atividades do processo de Deployment.

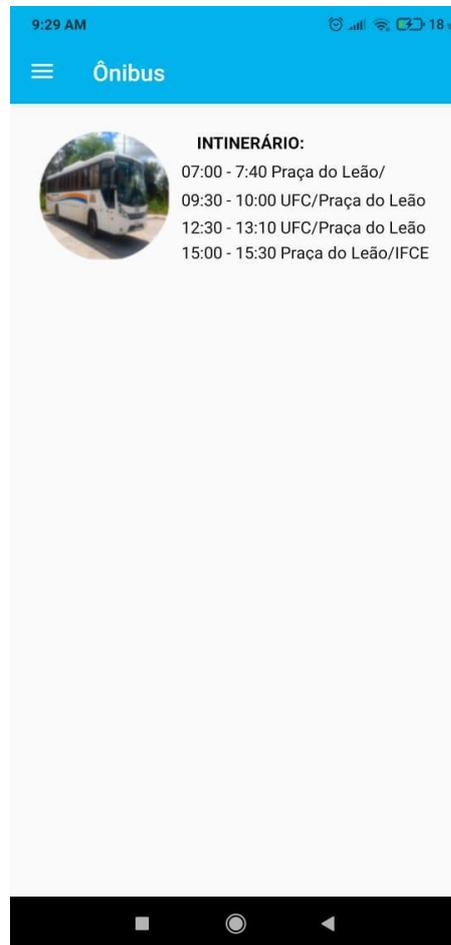
5.3.2.6 Sexta sprint: Cadastro e visualização do ônibus

Na etapa de cadastro do ônibus, foi catalogado o transporte universitário da UFC que faz o percurso Praça José de Barros/IFCE e IFCE/Praça José de Barros. As informações que foram armazenadas foram o itinerário, imagem e descrição do ônibus.

Essas informações foram utilizadas para a consulta do ônibus da UFC que faz o transporte dos alunos. Dessa forma, é possível visualizar o itinerário, cor e acessibilidade do transporte universitário.

Caso uma requisição de posição, em tempo real, seja feita em um horário fora do itinerário, o usuário será levado à tela de visualização do ônibus, para visualizar os horários em que as consultas de posições podem ser feitas. A Figura 12 mostra como ficou a tela de consulta dos ônibus.

Figura 12 – Tela de consulta dos ônibus



Fonte: elaborado pelo autor

Entre as datas de 16/03/2020 e a data de início da sprint, 04/05/2020, ocorreu a paralisação das aulas presenciais por conta da pandemia do coronavírus, o que impossibilitou o acesso à informação de todos os ônibus que fazem o percurso. Portanto, como um único ônibus será cadastrado foi determinado 40 horas, uma semana de trabalho equivalente a uma *sprint* para a conclusão dessa etapa. No entanto, foram necessárias 32 horas, 4 dias de trabalho para sua conclusão. Essa sprint passou por um retrabalho o qual será detalhado na atividade do processo de Deployment.

5.3.2.7 Sétima sprint: Mapear parada e posição do ônibus

Para mapear o percurso posição do ônibus/parada escolhida pelo usuário, foi usada a Direction API do Google. A ideia é utilizar da Direction API para fazer requisições html no formato:

```
https://maps.googleapis.com/maps/api/directions/outputFormat?parameters
```

O parâmetro *outputFormat* indica o tipo de arquivo, que a resposta da requisição, será baixado. Esses valores podem ser de dois tipos XML ou JSON. O parâmetro *parameters* será utilizado para representar a origem e o destino do percurso. A requisição feita pelo aplicativo PASTORABUS é feita no seguinte formato:

```
https://maps.googleapis.com/maps/api/directions/json?origin=latitude_posição_do_ônibus,longitude_posição_do_ônibus&destination=latitude_parada_escolhida_usuario,longitude_parada_escolhida_usuario
```

O resultado dessa requisição é um arquivo JSON contendo a melhor rota entre os pontos escolhidos. No arquivo, é possível acessar a *polyline* da melhor rota entre os dois pontos.

A *polyline* da rota é lida e desenhada no mapa da aplicação. Ela pode ser usada para calcular o tempo de espera de um usuário até que o ônibus esteja na sua parada escolhida. Usando sua largura e a velocidade média do ônibus.

O problema surge no percurso IFCE/Praça José de Barros. Como a requisição retorna a melhor rota entre dois pontos, ela passa a considerar caminhos intermediários, que não passam pela Praça José de Barros. Isso acarreta que a *polyline* desenhada não é realmente a da rota do ônibus.

Segundo o Google Developers (2020), o serviço de requisições utilizando a Direction API não foi projetado para responder em tempo real à entrada do usuário. Ele aconselha o uso de outra API chamada de Maps JavaScript API. Entretanto, essa API é paga o que inviabiliza seu uso na aplicação PASTORABUS.

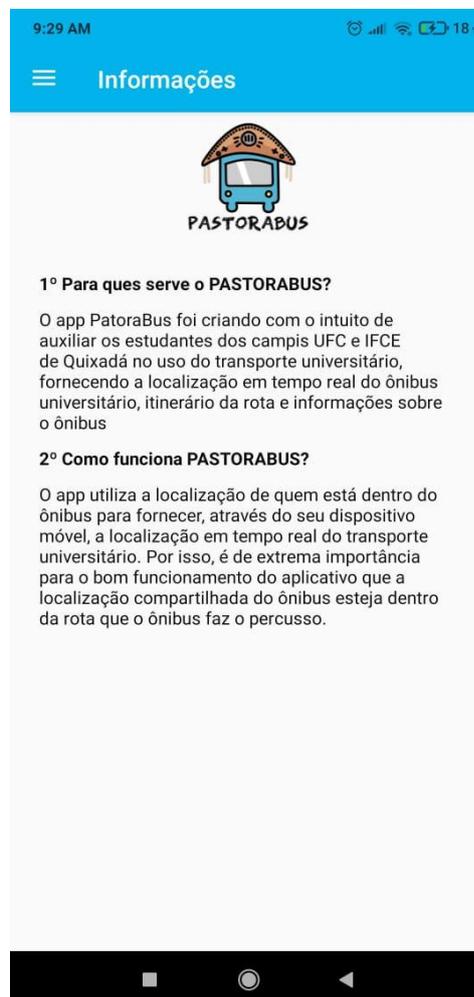
Nessa Sprint, foram estabelecidas 80 horas, duas sprints, o que equivale a duas semanas de trabalho para sua conclusão, mas foram gastas 72 horas para a sua conclusão. Esse prazo foi estabelecido pela dificuldade de encontrar meios que pudessem criar rotas entre dois pontos em tempo real.

5.3.2.8 Oitava sprint: Informações sobre o APP

Para informar o usuário sobre como o aplicativo PASTORABUS funciona, foi criada uma tela onde todas essas informações estivessem contidas de forma clara para o entendimento do usuário.

Como essa etapa era de fácil implementação, já que era apenas uma tela de informações sobre o aplicativo, foi estabelecido um prazo de 40 horas (uma sprint) para sua conclusão e foram gastas 40 horas para o término. Essa sprint passou por um retrabalho o qual será mais bem definido nas atividades do processo de Deployment.

Figura 13 – Tela de informações aplicação PASTORABUS.



Fonte: elaborado pelo autor

5.3.3 Deployment

Nesta etapa, seria disponibilizado o produto para uso do cliente para que o mesmo pudesse validar as funcionalidades do aplicativo, assim, gerando a ata de validação e a versão final do produto. Entretanto, com a paralisação das aulas presenciais, por conta da pandemia do novo coronavírus, desde a data de 16/03/2020, ocorreu que esta etapa sofreu alteração porque se tornou inviável o teste de campo com os alunos dos campi UFC e IFCE.

Nos testes e validação das funcionalidades, foi disponibilizado a um grupo de validação o arquivo apk da aplicação PASTORABUS e a ata de validação. O maior problema nesta etapa foi simular a localização dos participantes quando eles não estivessem na rota do ônibus no horário dos testes.

Para tanto, foi utilizado o aplicativo Lockito – Fake GPS itinerary¹¹, que permitiu que o celular dos participantes dos testes seguisse um itinerário falso, com total controle sobre a velocidade, altitude e precisão do sinal GPS. Dessa forma, os participantes simularam a rota que compõem o percurso Praça José de Barros para os campi UFC/IFCE.

Com a capacidade de simular a rota do ônibus, mesmo estando fora dela, foi disponibilizado para os participantes a ata de validação com o intuito deles avaliarem cada uma das funcionalidades do aplicativo. O Quadro 5 apresenta esse documento após a etapa de testes.

Quadro 5 - Ata de validação da aplicação PASTORABUS.

ID	Item da sprint backlog	Validado: Sim ou Não
1	Mapear Rotas (Mapear paradas)	Não
2	Confirmação de compartilhamento e verificar localização compartilhada	Não
3	Compartilhamento em segundo plano	Não
4	Cadastro e visualização de dados da parada	Não
5	Identificar posição do ônibus	Não
6	Cadastro dos ônibus e Consulta de ônibus	Não
7	Mapear parada e posição do ônibus	Não
8	Informações sobre o APP	Não

Fonte: elaborado pelo autor

¹¹ https://play.google.com/store/apps/details?id=fr.dvilleneuve.lockito&hl=pt_BR

Nos *sprints* 3 e 7, sabia-se, previamente, na atividade do processo *Sprint*, que essas funcionalidades foram implementadas, mas não funcionaram como deveriam para aplicação, dessa forma, elas não foram validadas e retiradas.

As *sprints* 1, 2 e 5 precisaram passar por um retrabalho, pelo fato de que a posição do ônibus não estava muito precisa, pois estava ficando um pouco fora da rota do ônibus.

Para as *sprints* 4, 6 e 8, foi identificado que seria necessário um retrabalho em cada uma delas. O menu de consultas, que continham as informações de parada ônibus e aplicativo, estava mal localizado na aplicação e os participantes da etapa de teste tiveram dificuldades em encontrá-las.

Vale ressaltar que toda a comunicação entre o grupo de validação e o desenvolvedor individual foi feita via Discord. Assim, foi possível repassar todas as informações necessárias para a conclusão dessa etapa.

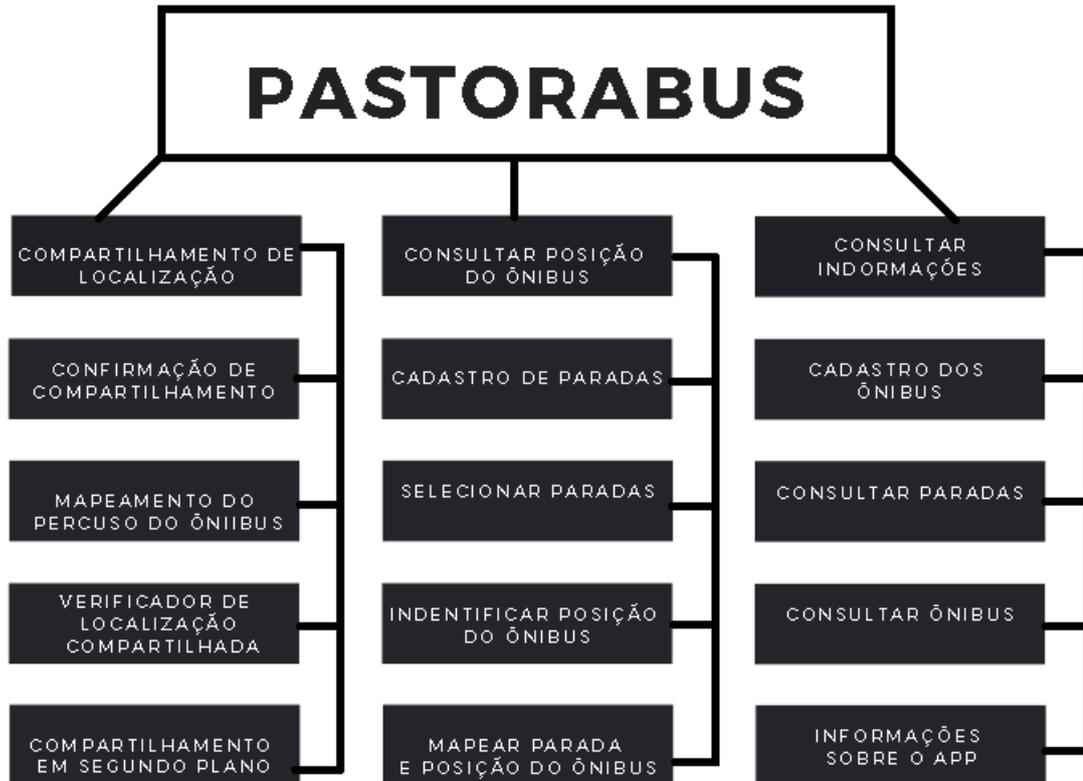
Essa atividade do processo foi selecionada como uma *Sprint* da aplicação PASTORABUS, sendo a nona *sprint* do *sprint backlog*. Assim ela teve início 06/07/2020 e término no dia 10/07/2020. Dessa forma, o tempo planejado de 40 horas, para essa *sprint* de testes, foi utilizado para a sua conclusão.

5.3.4 Management

Esta etapa tem o objetivo de planejar, monitorar e controlar o desenvolvimento do produto. É importante ressaltar que essa etapa teve duração do início até o fim do projeto PASTORABUS.

O primeiro documento gerado nessa etapa foi a Estrutura Analítica do Projeto (EAP). A EAP utilizada para ter uma maior compreensão sobre o escopo e as atividades que serão desenvolvidas no projeto. A Figura 10 mostra como ficou organizada a EAP da aplicação PASTORABUS.

Figura 14 - Estrutura analítica do projeto PASTORABUS.



Fonte: elaborado pelo autor

Nessa estrutura, é possível notar que no primeiro nível encontra-se o nome da aplicação, no caso PASTORABUS. No segundo nível, estão as macro funcionalidades: compartilhamento de localização, consulta da posição em tempo real do ônibus e consulta das informações sobre paradas, ônibus e app. No terceiro nível, estão os pacotes do trabalho de cada atividade.

Na produção do cronograma vale salientar que a etapa de *sprint* teve início no dia 16/03/2020 e término no dia 17/07/2020 e que as sprints tiveram prazo máximo de duas semanas. O Quadro 6 mostra o cronograma do projeto PASTORABUS.

Quadro 6- Calendário da aplicação PASTORABUS

Pacote de trabalho	Sprints (duas semanas)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mapear Rotas (Mapear paradas)	■													
Confirmação de compartilhamento e verificar localização compartilhada		■	■											
Compartilhamento em segundo plano				■	■									
Cadastro e visualização de dados da parada						■								
Identificar posição do ônibus							■	■						
Cadastro dos ônibus e Consulta de ônibus									■					
Mapear parada e posição do ônibus										■	■			
Informações sobre o APP												■		
Etapa de testes													■	
Correção de bugs														■

Fonte: elaborado pelo autor

Os autores do *scrum* solo Pagotto, Et Al (2016) aconselham, nesta atividade do processo, a criação da planilha de custo. Entretanto, como a aplicação tem como destinatário os estudantes dos campi UFC e IFCE, a ideia do aplicativo era eliminar qualquer custo financeiro para sua construção e usuário do aplicativo. No entanto, o tempo de desenvolvimento foi planejado e registrado o realizado. Para tanto, foi usado um quadro onde era marcado o início de uma sprint e sua finalização através de *post-it*. O quadro foi usado para marcar o real prazo de uma sprint. A Tabela 7 mostra como ficou a configuração desse quadro.

Quadro 7- Calendário da execução das sprints da aplicação PASTORABUS

Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
Início da execução da sprint Mapear Rota (Mapear paradas)	Execução da sprint	Execução da Sprint	Data finalização da sprint	X
Início da execução da sprint Confirmação de compartilhamento e verificar localização compartilhada	Execução da sprint	Execução da Sprint	Execução da sprint	Execução da sprint
Execução da sprint	Execução da sprint	Execução da Sprint	Execução da sprint	Data finalização da sprint
Início da execução da sprint Compartilhamento em segundo plano	Execução da sprint	Execução da Sprint	Execução da sprint	Execução da sprint
Execução da sprint	Execução da sprint	Execução da Sprint	Execução da sprint	Data finalização da sprint
Início da execução da sprint Cadastro e visualização dos dados de paradas.	Execução da sprint	Execução da Sprint	Data finalização da sprint	X
Início da execução da sprint identificar posição do ônibus	Execução da sprint	Execução da Sprint	Execução da sprint	Execução da sprint
Execução da sprint	Execução da sprint	Data finalização da Sprint	x	X
Início da execução da sprint Cadastro dos ônibus e Consulta de ônibus	Execução da sprint	Execução da Sprint	Data finalização da sprint	X
Início da execução da sprint mapear parada e posição do ônibus	Execução da sprint	Execução da Sprint	Execução da sprint	Execução da sprint
Execução da sprint	Execução da sprint	Execução da Sprint	Data finalização da sprint	X
Início da execução Informações sobre o APP	Execução da sprint	Execução da Sprint	Execução da sprint	Data finalização da sprint

Fonte: elaborado pelo autor

Pode-se destacar que a etapa de *Sprint* teve início no dia 16/03/2020 e término no dia 17/07/2020, no sprint de correção de *bugs*, mas tanto ela como a sprint de teste não foram registradas no quadro de atividades.

5.4 Bugs e Melhorias

Nesta seção, são descritos os bugs encontrados a partir da etapa de teste, feita na atividade do processo Deployment, e como esses bugs foram resolvidos. Esta etapa estava contida no sprint backlog da aplicação PASTORABUS.

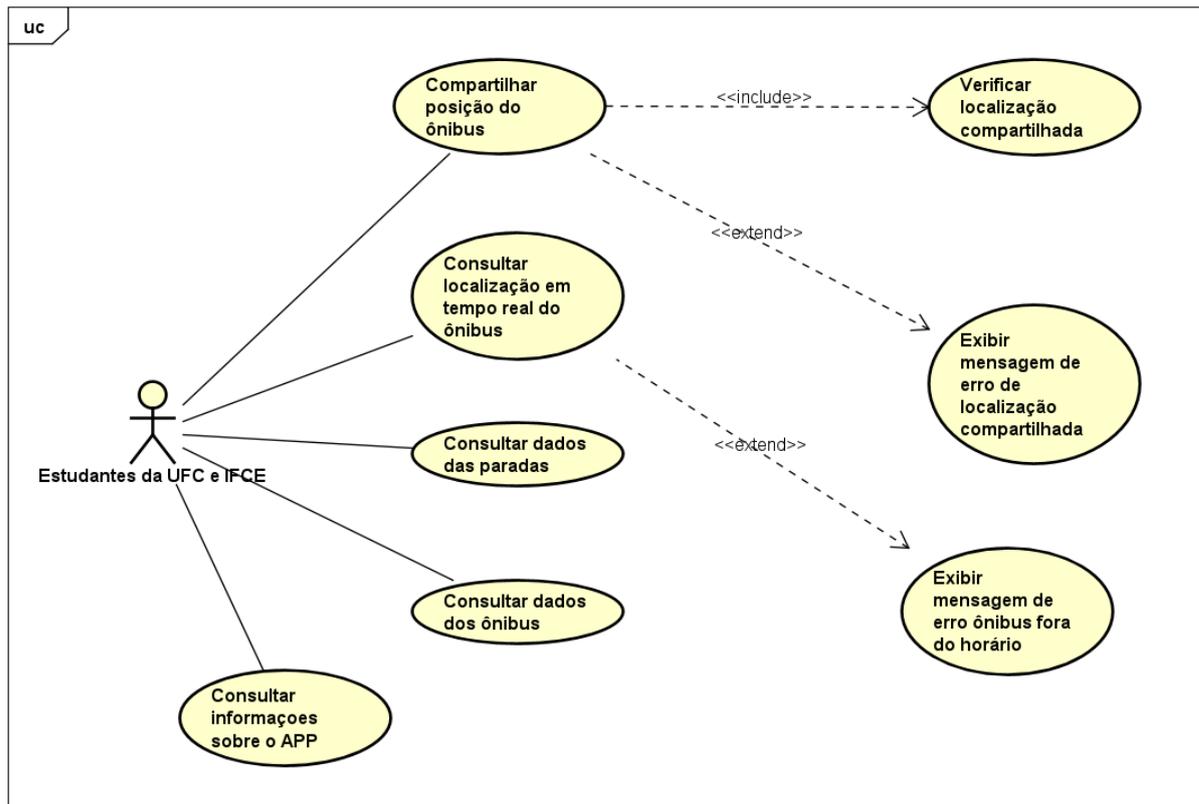
5.4.1 Correção de bugs

A primeira correção é feita a partir dos resultados dos testes nos *sprints* Mapear Rota (Mapear paradas), Confirmação de compartilhamento e verificar localização compartilhada e identificar posição do ônibus. Foi notado pelos participantes dos testes, que mesmo com os métodos usados para atingir a melhor localização em tempo real do ônibus, a localização continuava um pouco imprecisa e, que mesmo compartilhando sua posição dentro da rota, em alguns pontos o aplicativo solicitava ir para dentro da rota.

Como forma de solucionar o primeiro problema, o conjunto de círculos com raio de 50 metros por todo o percurso tiveram seus raios alterados para 30 m, dessa forma, diminuindo o conjunto de posições aceitas pelo aplicativo. Para o segundo problema, foi feito um novo mapeamento da rota do ônibus, adicionando mais círculos de raio de 30 metros nas posições que não tinham sido identificadas anteriormente.

Para as *sprints* Cadastro e visualização dos dados das paradas, Cadastro dos ônibus e consultas dos ônibus e Informações sobre o APP, percebeu-se que seria necessário um retrabalho em cada uma delas. Não seria mais necessário escolher previamente a parada em que o usuário iria esperar pelo transporte, por conta da remoção da sprint mapear parada posição do ônibus da aplicação. Dessa forma, o menu de consultas, que anteriormente só estava disponível quando a consulta da posição do ônibus fosse feita, foi trazido para a tela inicial da aplicação. A Figura 11 mostra, através do diagrama de caso de uso, as funcionalidades da aplicação PASTORABUS.

Figura 15- Diagrama de caso de uso aplicação PASTORABUS



Fonte: elaborado pelo autor

A etapa de correção de bugs foi selecionada como uma *Sprint* da aplicação PastoraBus, sendo a décima *sprint* do *sprint backlog*. Assim, ela teve início no dia 13/07/2020 e término no dia 17/07/2020, uma *sprint* foi gasta para sua conclusão. O tempo planejado de 40 horas foi utilizado para a finalização.

6 CONSIDERAÇÕES FINAIS

O referido trabalho apresentou a problemática de como facilitar o deslocamento dos estudantes que utilizam o transporte universitário em Quixadá. Este trabalho utilizou os serviços LBS e uma adaptação do Scrum para o desenvolvimento do aplicativo, consequentemente a solução da problemática exposta.

Este trabalho tem como propósito apresentar o desenvolvimento de uma ferramenta que pudesse auxiliar os estudantes dos campi UFC e IFCE em Quixadá. Para isso, foi necessário o uso dos serviços LBS que juntamente com um conjunto de APIs, disponibilizadas pelo Google, tornou possível a conclusão deste trabalho. Entretanto algumas funcionalidades previstas

anteriormente foram implementadas, mas não se mostram eficientes para o uso em tempo real.

Para o desenvolvimento da ferramenta PASTORABUS foi aplicado uma adaptação do Scrum para equipes de um só indivíduo, no desenvolvimento do software, chamada de Scrum solo. Ela se mostrou muito eficiente no sentido do desenvolvimento individual, mas algumas ressalvas devem ser feitas. Primeiramente, alguns dos artefatos disponibilizados por seus autores são muito simples e escassos de informações, que poderiam ser úteis no desenvolvimento da ferramenta. Outra ressalva é que não é disponibilizado nem um artefato para gerenciar as atividades, fazendo com que não tenha nem uma forma de acompanhar a implementação das *sprints*.

Como forma de amenizar a segunda problemática, utilizou-se um quadro onde, com a ajuda de post-it, era marcado o início de uma sprint e sua finalização. Desse modo, foi muito mais fácil gerenciar quando uma sprint acabava e quando ela iniciava e computar o real tempo gasto naquela sprint.

Infelizmente, por não ter os testes de campo, houve um comprometimento sobre a efetividade ou não do aplicativo PASTORABUS no auxílio da locomoção dos estudantes para a faculdade. Vale ressaltar que os testes com o grupo de validação do aplicativo atingiram seu propósito de monitorar e disponibilizar a localização em tempo real do ônibus e de disponibilizar informações sobre os ônibus que compõem o percurso como também as paradas.

O aplicativo PASTORABUS pode ser muito útil para os estudantes dos campi UFC e IFCE de Quixadá. Pois, com a funcionalidade de mostrar a posição em tempo real do ônibus, o horário do estudante pode ser mais bem gerenciado, já que ele não vai ficar horas na fila de espera. As informações de consultas poderão ser úteis para os novos estudantes dos campi para analisar quais os ônibus que fazem a rota como também as paradas que fazem parte do percurso.

O desenvolvimento desse aplicativo foi de grande relevância, como aluno da UFC campus Quixadá, vivenciou de perto a dificuldade que os alunos têm no uso do transporte universitário com a pequena quantidade da frota de ônibus disponibilizada e as incertezas sobre o horário correto e se haverá ônibus fazendo o trajeto. A criação de um aplicativo que possa ajudá-los com o transporte até a faculdade pode ser um meio facilitador para os acadêmicos.

Como trabalho futuro, foi identificado a dificuldade de meios de executar o compartilhamento em segundo plano para aplicações em tempo real. Uma possível solução para este problema seria o uso da gamificação. A gamificação é habitualmente descrita como a utilização de jogos em contextos que não são de jogos (DETERDING, 2011). Segundo BUNCHBALL INC (2010), a gamificação pode ser mais bem descrita como uma forma mais consistente e estratégica, que utiliza de jogos e seus elementos, para aplicações, que não tem

um vencedor ou conclusão, como uma forma de influenciar e causar mudanças no comportamento de indivíduos e grupos.

Dessa forma, a gamificação seria um meio excelente de influenciar os estudantes dos campi UFC e IFCE a usarem a aplicação PASTORABUS. A ideia seria utilizar de elementos de jogos para torna o aplicativo mais atrativo ao usuário, mas que só estaria visível para o usuário que está dentro da rota do ônibus e enquanto ele faz uso desses elementos compartilharia sua localização ajudando, assim, na real funcionalidade da aplicação PASTORABUS.

Outra solução para esse problema, pode ser a criação de uma versão do PASTORABUS para o motorista do ônibus, desse modo a localização do seu aparelho poderia ser usada quando o ônibus estivesse no horário de fazer o percurso e não tivesse nem uma localização sendo compartilhada naquele momento.

Para o problema da sprint 7 “mapear parada e posição em tempo real do ônibus”, pode-se fazer uso da Maps JavaScript API¹². Entretanto, essa API é paga. Outro problema é com os cadastros dos outros ônibus, que fazem a rota, pois não tem uma forma de identificar de qual ônibus o usuário está compartilhando sua localização. Como forma de solucionar esse problema, pode-se pedir para o usuário, antes de compartilhar a sua localização, selecionar de qual ônibus ele vai fazer o compartilhamento.

¹² <https://developer.android.com/guide/platform>

REFERÊNCIAS

BRASIL. **Artigo 5 da Lei nº 12.816** de 05 de jun de 2013.

BUNCHBALL INC. **Gamification 101**: an introduction to the use of game dynamics to influence behavior. [S.I], 2010. Disponível em:https://www.biworldwide.com/gamification/bunchball-nitro/?utm_source=bunchball.com&utm_medium=referral&utm_campaign=domain_redirect. Acesso em: 18 jul. 2020.

Cadê o Ônibus? 2012. Disponível em: <https://play.google.com/store/apps/details?id=solucoes.zeno.sinetram>. Acesso em 5 de out de 2019.

CENTRAL GLOBO. **Ônibus superlotado faz transporte de universitários em Quixadá-CE**. G1. Disponível em: <http://g1.globo.com/ceara/videos/t/todos-os-videos/v/onibus-superlotado-faz-o-transporte-de-universitarios-em-quixada/7529925/>. Acesso em: 26 de set de 2019.

Chen, X.; Chen, Y.; Rao, F. **An Efficient Spatial Publish/Subscribe System for Intelligent Location-Based Services**. [S.I], Proceedings of the 2nd international workshop on Distributed Event-Based Systems, San Diego, California, 2003.

CITYMAPPER, 2013. Disponível em: <https://play.google.com/store/apps/details?id=com.citymapper.app.release>. Acesso em: 5 de out de 2019.

CITTAMOBI, 2014. Disponível em: <https://play.google.com/store/apps/details?id=br.com.cittabus>.

DETERDING, S. **Gamification**: designing for motivation. Interactions. [S.I] v.19, n. 4, p. 14-17, jul./ago., 2012. Disponível em: <https://dl.acm.org/doi/10.1145/2212877.2212883>. Acesso em: 28 jul. 2020.

EXAMPLES OF LOCATION-BASED SERVICES APPS, 2018. Disponível em:https://medium.com/@the_manifest/7-examples-of-location-based-services-apps82b8be3bdcac. Acesso em: 15 de set de 2019.

Françoso, M.; Mello, N. **Influência dos aplicativos de smartphones para transporte urbano de trânsito**. 7º Congresso Luso Brasileiro para o Planejamento Urbano, Regional, Integrado e Sustentável Contrastes, Contradições e Complexidades. Brasil. Maceió, 2016.

Freitas, F.; Morais, R. F; Jaques, P. **Um Sistema Web de consulta de trajeto de transporte público**. PIPCA – Universidade do Vale do Rio dos Sinos (UNISINOS). Rio de Janeiro. 2011.

Gonçalves, Kleider Miranda. **Um Framework para Comunicação Baseada em Localização**. Dissertação de Mestrado Dissertação. Rio de Janeiro - RJ, 14 de abril de 2005. Disponível em: <https://docplayer.com.br/31677809-Kleider-miranda-goncalves-um-framework-para-comunicacao-baseada-em-localizacao.html>. Acesso em: 20 de set de 2019.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5. Ed. São Paulo: Atlas, 2008.

Google Maps Go, 2018. Disponível em: <https://play.google.com/store/apps/details?id=com.google.android.apps.mapslite>. Acesso em: 5 de outubro de 2019.

Google Forms – Pesquisas e formulários fáceis de criar para todos. Disponível em: https://www.google.com/forms/about/?utm_source=gaboutpage&utm_medium=formslink&utm_campaign=gabout. Acesso em: 26 de set de 2019

J. A. Fabri, A. L'Erario e T. Pagotto, **SCRUM SOLO** Processo de software para desenvolvimento individual, [S.I]. Cornélio Procópio, Brasil. 2016.

J.F.G. Monico. **Posicionamento Pelo GNSS: Descrição, Fundamentos e Aplicações**. [S.I], UNESP, São Paulo, 2008.

J. Sutherland, A. Viktorov, J. Blount e N. Puntikov. **Distributed SCRUM: Agile Project Management with Outsourced Development Teams**. [S.I], Hawaii International Conference on Software Systems (HICSS'40), 2007.

K. Schwaber. **Agile Project Development with SCRUM**. [S.I], capítulos 1-2, Microsoft Press, 2004.

Lima, V.; Magalhães, F.; Tito, A.; Santos, R.; Ristar, A.; Santos, L.; Vieira, V.; Salgado, A. **UbibusRoute: Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais**. [S.I], Recife – PE – Brasil.2011.

Lockito – Fake GPS itinerary. Disponível em: https://play.google.com/store/apps/details?id=fr.dvilleneuve.lockito&hl=pt_BR. Acesso em: 04 de Mar de 2020.

MOBILITY LAB. **Here are the three best public transit apps**. Disponível em: <https://mobilitylab.org/2018/08/31/here-are-the-three-best-public-transit-apps>. Acesso em: 15 de out de 2019.

Moovit, 2012. Disponível em: <https://play.google.com/store/apps/details?id=com.tranzmate>. Acesso em: em 5 de outubro de 2019.

Meu Ônibus - Fortaleza. 2015. Disponível em: <https://play.google.com/store/apps/details?id=br.com.m2m.meuonibus>. Acesso em: 5 de out de 2019.

Leung, H.; Burcea, I.; Jacobsen, H. Modeling **Location-based Services with Subject Spaces**. [S.I.], Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, 2003.

Siqueira, L. Gourlate. **Aplicação Bus Tracker: Oferecendo Uma Melhor Experiência Aos Usuários Do Transporte Público, A Partir Da Utilização De Informações De Rastreamento Veicular**. [S.I.], Trabalho de Graduação – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2012.

TRANSIT, 2013. Disponível em: <https://play.google.com/store/apps/details?id=com.thetransitapp.droid>. Acesso em: 5 de out de 2019.

Schiller, J. Voisard, A. **Location-Based Services, Morgan Kaufmann Series in Data Management Systems, Elsevier Inc.** [S.I.], 2004.

Silva, Danyela Morais. **Sistemas Inteligentes no Transporte Público por Ônibus**. [S.I.], Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

TECHTUDO. **Saiba o que é GPS e como funciona,** 2014. Disponível em <https://www.techtudo.com.br/artigos/noticia/2011/12/como-funciona-o-gps.html>. Acesso em: 20 de set de 2019.

TECHOPEDIA. **Definition - What does Location Based Services (LBS) mean?.** Disponível em: <https://www.techopedia.com/definition/12888/location-based-services-lbs>. Acesso em: 26 de set de 2019.