UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

GIONATTA MARCON MOCELLIN
GIUSEPE PIETRO NIQUELE

UTILIZAÇÃO DE APLICATIVO MÓVEL E *CROWDSOURCING* PARA RASTREIO DE ÔNIBUS

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2014

GIONATTA MARCON MOCELLIN
GIUSEPE PIETRO NIQUELE

UTILIZAÇÃO DE APLICATIVO MÓVEL E *CROWDSOURCING* PARA RASTREIO DE ÔNIBUS

Proposta de Trabalho de Conclusão de Curso de Engenharia de Computação apresentado ao Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de "Engenheiro em Computação".

Orientadora: Prof^a. Dr^a. Ana Cristina Kochem Vendramin

Co-orientadora: Prof^a. Dr^a. Anelise Munaretto Fonseca

CURITIBA

2014

LISTA DE SIGLAS

ICT	*Information and Communication Technology*
GPS	*Global Positioning System*
URBS	Companhia de Urbanização e Saneamento de Curitiba
MBTS	Mobile Bus Tracking System
SMS	*Short Message Service*
JSP	JavaServer Pages
RF	Radio frequência
P2P	*Peer-to-Peer*
DTN	*Delay Tolerant Networks*
CVS	*Control Version System*
ADT	*Android Developer Tools*
SDK	*Software Development Kit*
API	*Application Programming Interface*
JDK	*Java Development Kit*

SUMÁRIO

1 INTRODUÇÃO

O conceito de *Smart Cities* (Cidades Inteligentes, em tradução livre) é um sustentáculo elementar para o desenvolvimento urbano sustentável. De certa forma, ele poderia contribuir na solução de diversos problemas críticos, originados a partir da urbanização de grandes cidades como congestionamentos, poluição do meio ambiente, e os limites dos recursos naturais. (PAN et al., 2013).

Um outro aspecto do conceito de cidades inteligentes a ser considerado, é a importância crescente das tecnologias digitais para um futuro sustentável. Assim sendo, destaca-se seis principais áreas, nas quais essas inovações digitais tem a competência para fazer a diferença: vida inteligente, governança inteligente, economia inteligente, ambiente inteligente, pessoas inteligentes e mobilidade inteligente (SCHUURMAN et al., 2012).

Com o uso da tecnologia da informação e comunicação (do inglês ICT, *Information and Communication Technology*,) a análise e mineração de dados de sensoriamento grande cidades é um passo importante para considerá-las uma cidade inteligente. Por exemplo, informações sobre a mobilidade dos veículos e pessoas tornaram-se temas em voga nos últimos tempos devido à prevalência do GPS (*Global Positioning System*) e outras tecnologias de localização. Esse conhecimento sobre a cidade e seus habitantes poderá trazer benefícios nas áreas de transporte, planejamento urbano, saúde pública, segurança pública e comércio (PAN et al., 2013).

Embora as ações de sensoriamento e mineração dos dados seja de imprescindível importância, algo deve ser feito para transformar esses dados em informações. Nesse sentido, uma abordagem interessante para a resolução desta questão consiste no conceito de *crowdsourcing* (SCHUURMAN et al., 2012).

Howe (2006) caracteriza *crowdsourcing* como o fenômeno no qual várias pessoas comuns dedicam seu tempo livre na busca de soluções para um problema de interesse coletivo. Isto posto, a massificação dos chamados *smartphones* e o advento de redes móveis e sem fio tais como o 3G e o WiFi e recentemente, o 4G, permitem que pessoas fiquem

conectadas a maior parte do tempo, em qualquer lugar. Dentre os inúmeros benefícios que os *smartphones* fornecem, a possibilidade de troca de informações entre usuários é uma delas, bem como a aplicação do conceito de *crowdsourcing*.

Considera-se o seguinte cenário: uma pessoa que acabou de chegar em um ponto de ônibus, deseja saber onde o mesmo se encontra. Questões como "será que o ônibus está chegando?", "será que vai demorar?" ou "será que o ônibus estragou?" são bastante comuns. Mas como descobrir suas respostas? Infelizmente, não é possível respondê-las apenas com a tabela de horários de chegada dos ônibus, disponível nos terminais e no *site* da URBS (Companhia de Urbanização e Saneamento de Curitiba).

Nesse sentido, é proposto nesse documento uma solução possível para esse quesito. A ideia básica é a construção de um aplicativo que alie conceitos, tanto de *Smart Cities* quanto de *crowdsourcing*, no qual informações em tempo real sobre o transporte coletivo, especificamente no que compete o estado atual de uma linha de ônibus, sejam disponibilizadas aos usuários.

Alguns aplicativos para dispositivos móveis foram desenvolvidos com o objetivo de prover, ao usuário, informações referentes ao transporte público. Um dos aplicativos mais recentes é o chamado "Busão Curitibano", lançado em 2013[1]. Os desenvolvedores – e também os usuários do aplicativo – encontraram algumas resistências por parte da URBS, pois o aplicativo usava a base de dados do transporte coletivo. Devido aos inúmeros acessos, acabava derrubando o servidor da URBS e tornando o serviço de acompanhamento de ônibus indisponível.

Outro aplicativo semelhante foi desenvolvido por Sujatha et al. (2014). Os autores propuseram um sistema chamado MBTS (*Mobile Bus Tracking System*) que ajuda qualquer pessoa a obter informações de um determinado ônibus sem fazer ligações ou perturbar passageiros com mensagens SMS (*Short Message Service*). Um banco de dados é utilizado para armazenar informações sobre os ônibus (por exemplo, as coordenadas do veículo), as quais estarão disponíveis para acesso através de um aplicativo Android..

1.1 JUSTIFICATIVA

A utilização de uma arquitetura centralizada, estilo cliente–servidor é um ponto comum nos dois trabalhos relacionados. Em ambos, nota-se que a arquitetura encontra-se em uma relação n:1, ou seja, vários clientes para um único servidor. Alguns dos problemas

[1]Para mais informações sobre o aplicativo, consultar notícia da Gazeta do Povo, disponível em: http://www.gazetadopovo.com.br/vidaecidadania/conteudo.phtml?id=1377575

que podem surgir nesse tipo de arquitetura é que o servidor pode ficar sobrecarregado e eventualmente ficar indisponível devido ao grande número de usuários utilizando o serviço, como foi o caso do "Busão Curitibano". Além disso, nesse caso, o servidor utilizado não é de propriedade dos desenvolvedores, mas sim da Urbs. Caso a companhia deseje interromper o serviço de fornecimento de dados, o "Busão Curitibano" perde sua utilidade. Isso não constitui um problema para o MBTS, pois o mesmo concentra as informações em um servidor próprio, mantido pelos próprios desenvolvedores.

Sendo assim, a principal motivação para o desenvolvimento deste projeto é implementar um aplicativo de acompanhamento de ônibus para dispositivos móveis – *smartphones* – que utilize uma rede descentralizada. Nela, os próprios usuários são os nós, que compartilharão informações entre si. Para tal, utilizar como referência o aplicativo móvel "Waze"[2], que implementa uma espécie de comunidade para mapeamento do trânsito em tempo real. Através do "Waze", cerca de 40 milhões de usuários compartilham informações, no mundo todo (TECHTUDO, 2014).

Além disso, usufruir do recurso de GPS, já utilizado nos ônibus de Curitiba, para fornecer aos usuários a posição dos veículos.

1.2 OBJETIVO GERAL

Desenvolver um aplicativo para *smartphones* que rodam sobre o sistema operacional Android, para acompanhamento de ônibus em Curitiba, unindo idéias de aplicativos já existentes como o "Waze" e o "Busão Curitibano", e o conceito de *crowdsourcing*.

1.3 OBJETIVOS ESPECÍFICOS

- Desenvolver um aplicativo, para *smartphones*, para acompanhamento de ônibus da rede de transporte de Curitiba;
- Estudar e implementar no aplicativo o conceito de *crowdsourcing*, para permitir que usuários colaborem entre si com informações pertinentes, através de *feedbacks* pré-definidos no aplicativo;
- Desenvolver o aplicativo para que o mesmo seja executado sobre uma plataforma que não exija custos de desenvolvimento;

[2]Uma análise do aplicativo pode ser encontrada em http://www.techtudo.com.br/tudo-sobre/waze.html

- Implementar uma arquitetura de rede descentralizada, a fim de evitar problemas como sobrecarga de servidor.

1.4 ESTRUTURA E ORGANIZAÇÃO

O presente documento encontra-se organizado da seguinte forma. O capítulo 1 apresenta a introdução, com a formulação do problema, justificativa do projeto, objetivo geral e específicos do mesmo. O capítulo 2 apresenta o levantamento bibliográfico, importante para fundamentar o projeto e fornecer um embasamento. A metodologia, que descreve as etapas de desenvolvimento pelas quais o projeto percorrerá, é apresentada no capítulo 3. Os recursos para desenvolvimento do projeto e a viabilidade do mesmo, são apresentados nos capítulos 4 e 5, respectivamente. Por fim, conclusões pertinentes ao desenvolvimento do projeto, tais como dificuldades esperadas e horizontes de projeto, são apresentados no capítulo 6.

2 LEVANTAMENTO BIBLIOGRÁFICO

Este capítulo apresenta uma breve introdução à Smart Cities, Crowdsourcing, contexto atual do desenvolvimento de programação voltada a aplicativos de transporte público e demais conceitos, pormenores, ao desenvolvimento do trabalho.

2.1 SMART CITIES

O conceito de Cidades Inteligentes e o respectivo aumento no seu uso, pode ser reconhecido como o resultado do alto crescimento das tecnologias digitais na intenção de garantir um futuro sustentável. Embora o conceito seja empregado normalmente com o objetivo de elaborar estratégias que objetivem melhorar a qualidade de vida dos cidadãos, criando um futuro sustentável, o conceito por si só continua sendo usado em diferentes contextos e assim permanece um tanto ambíguo. Dessa forma faz-se necessário estabelecer um critério muito importante que o diferencia das cidades-conceito, o aspecto colaborativo entre os diversos interessados, mais comumente os cidadãos (SCHUURMAN et al., 2012).

Um outro enfoque, mais empresarial, apresentado por Walravens (2012), menciona que o conceito também pode ser empregado para caracterizar um grupo de organizações que intentam por revolucionar algum aspecto em uma região, entre os quais estão parques empresariais, o nível de escolaridade de uma população, o uso de tecnologias em contextos urbanos, o aumento da eficácia de governos e especialmente aquelas com foco em ITC. Walravens argumenta também que de maneira geral, o uso de serviços de telefonia móvel adquirem uma importância inevitável, já que é um sub aspecto de ITC e dessa forma grande importância na construção de uma Cidade Inteligente (WALRAVENS, 2012).

Em vista disso, sistemas operacionais desenvolvidos para *smartphones* inspiram desenvolvedores a conceber aplicativos e serviços que aperfeiçoam a vida nas cidades de diversas formas diferentes, como por exemplo, facilitando acesso à informação sobre o transporte publico. Além disto, à medida que os *smartphones* se tornam mais acessíveis

e populares, cria-se o desejo de que se tornem eminentes ferramentas na busca de uma cidade mais inteligente (WALRAVENS, 2012).

2.2 CROWDSOURCING

Outro aspecto importante a ser mencionado é a caracterização de *crowdsourcing*, no que tange inteligência coletiva. À grosso modo, entende-se inteligência coletiva como quando indivíduos de um determinado grupo, que possui interesse por determinado assunto, combinam seus conhecimentos a fim de encontrar a solução para um determinado problema. Através da interação social, o conhecimento individual é compartilhado, corrigido, processado, enriquecido e avaliado. Normalmente, os resultados colhidos são melhores que os obtidos por um único indivíduo. Talvez a aplicação mais difundida desse conceito seja a Wikipedia (SCHUURMAN et al., 2012).

Por conseguinte, a ponte que se estabelece entre *crowdsourcing* e *smart cities* acompanha o advento dos smartfones. Para Kanhere (2011), as melhorias no poder de processamento, sensoriamento e capacidades de armazenamento permite que telefones celulares sejam comparados a dispositivos de computação. Esse fenômeno abre margem ao surgimento de um novo paradigma, denominado pela literatura de sensoriamento participativo, cuja ideia principal gira em torno de capacitar um cidadão comum a coletar e compartilhar dados de sensoriamento do ambiente em que está inserido, a partir de seu telefone celular (KANHERE, 2011).

Ainda segundo Kanhere, sensoriamento participativo possui quatro vantagens principais sobre redes de sensores tradicionais (já que esta necessita de uma gama considerável de dispositivos sem fio, principalmente em áreas urbanizadas), sendo elas: custo de implementação baixo, já que usa telefones celulares e Wi-Fi; uma ampla mobilidade e cobertura proporcionada pelas operadoras de telefonia; economias de escala proporcionadas pelo uso de celulares; a facilidade assegurada pelas lojas de aplicativos e também as inúmeras formas de desenvolvimento de software disponíveis para sistemas operacionais de dispositivos móveis (KANHERE, 2011).

Com base em todos esses conceitos, estabelece-se, a seguir, um breve estado da arte registrado na literatura, no que se refere à área de desenvolvimento de aplicativos para plataformas móveis dentro da ideia de *crowdsourcing* e *smart cities*.

2.3 ESTADO DA ARTE

Conforme já relacionado na Introdução, Sujatha et al. (2014) propuseram um sistema chamado MBTS para que qualquer pessoa obtenha informações de um determinado ônibus através de um aplicativo móvel.

Três são os atores do sistema: os passageiros, os chamados "coordenadores" do ônibus e as pessoas que estão aguardando em um ponto. Todos os atores necessitam de um *smartphone* com sistema operacional Android conectado à Internet. Continuamente, os passageiros e os "coordenadores" alimentam um banco de dados com informações sobre o ônibus. Essas informações podem ser as coordenadas do ônibus, obtidas através do GPS integrado ao *smartphone*, ou até mesmo a "situação" do ônibus – se ele está lotado ou vazio, se houve algum acidente, se o ônibus está atrasado, entre outros. Essas informações ficam armazenadas em um banco de dados para serem acessadas, posteriormente, por alguém que esteja utilizando o aplicativo, principalmente usuários que estão aguardando um ônibus em algum ponto.

A arquitetura do MBTS pode ser observada na Figura 1. Basicamente, consiste de duas aplicações, uma Web e outra Android. A primeira permite o registro de usuários e ônibus e a segunda é utilizada para rastreio. A aplicação Web, desenvolvida em JSP (*JavaServer Pages*), é voltada para o administrador do sistema e também funciona como um "middleware" para que o usuário, através da aplicação Android, se conecte ao banco de dados e armazene ou solicite informações. Na figura, o usuário pode representar um passageiro, um coordenador ou uma pessoa que esteja aguardando em um ponto de ônibus (SUJATHA et al., 2014).

Uma outra pesquisa neste área de acompanhamento de ônibus, envolve a utilização de RF (Radio frequência). Foi desenvolvida por Paradells et al. (2014).

Os autores propõe a utilização de uma rede de dados composta por transmissores e receptores RF. Os transmissores situam-se nos ônibus, os quais transmitem informações para os receptores, que situam-se nos pontos de ônibus que, basicamente, são os nós da rede. Assim que um ônibus se aproxima de um ponto de ônibus - nó - inicia a transmissão de informações relevantes, captadas por sensores atrelados ao veículo. Essas informações podem ser usadas/acessadas por um usuário que está aguardando no referido ponto de ônibus.

Uma limitação do sistema baseado em RF, e percebida pelos próprios autores, diz respeito à distância entre o ônibus e os nós da rede (neste caso os pontos de ônibus).

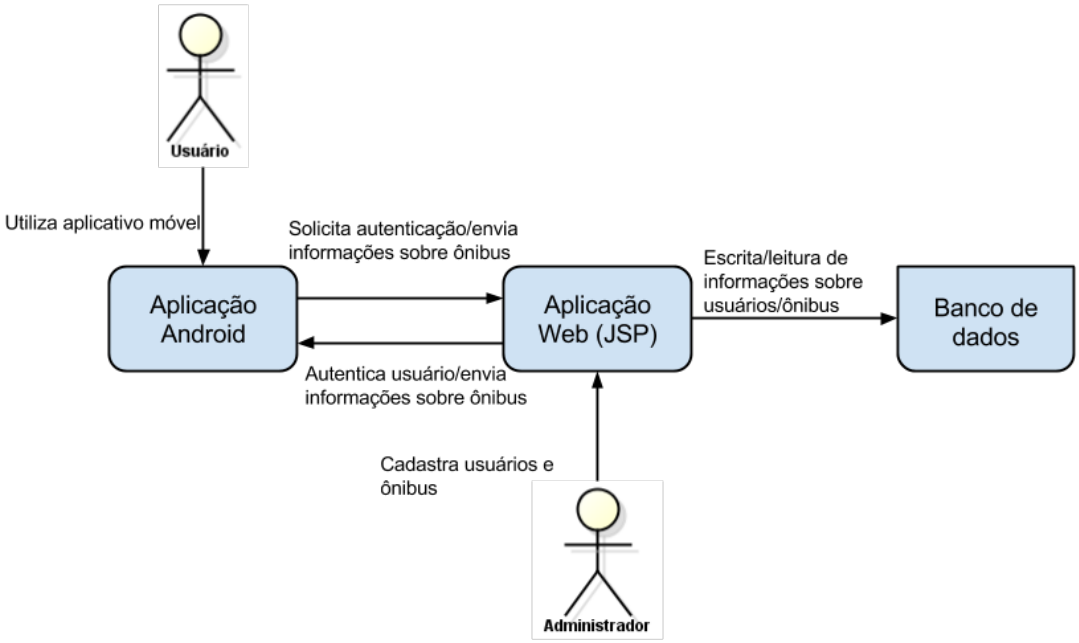

Figura 1: Arquitetura do MBTS.
Fonte: Adaptado de (SUJATHA et al., 2014).

Uma vez que sensores baseados em radio frequência possuem limitações de distância e na transmissão dos dados, acabam limitando a velocidade desenvolvida pelo veículo, o que torna o projeto inviável.

Ainda, as informações só podem ser coletadas por usuários que já estão próximos ao veículo (ou em pontos de ônibus próximos ao veículo). Isso acaba tirando o propósito de acompanhamento ou rastreio de ônibus, pois o mesmo já está próximo do ponto, e pode ser vislumbrado por um usuário sem o auxílio de um sistema específico para tal.

Alves, Martinez e Viegas (2012) apresentaram um sistema – chamado de *Trip-planner* – para planejar rotas em tempo real, para pessoas que utilizam transporte público em Lisboa. O sistema consegue informar quais são as melhores rotas e o tempo estimado de viagem para um determinado destino, baseados na estimativa de quantos veículos estão trafegando e quais são suas velocidades.

Informações de tempo-real são coletadas através do GPS equipado nos ônibus. Essas informações são utilizadas em um servidor (o que os autores chamam de *Data Center*) para atualizar históricos e melhorar as estimativas, uma vez que é utilizado um algoritmo para predição de tempos de viagem. Esses históricos se referem à relatórios de quatro meses, com informações sobre tempos de viagem e velocidades dos veículos. Essas

informações são analisadas e passam por um classificador, e uma vez processadas, são repassadas para qualquer dispositivo móvel conectados em uma rede sem fio, através de *broadcast* (ALVES; MARTINEZ; VIEGAS, 2012).

A Figura 2 descreve de forma simplificada o sistema proposto por Alves, Martinez e Viegas (2012). Nota-se claramente que dados históricos e de tempo-real são coletados com o auxílio do GPS instalado no ônibus. Esses dados passam pelo *Data Center* e são utilizados como entrada para um algoritmo de predição. Uma vez processados, distribuem-se os dados para dispositivos móveis através de *broadcast.*

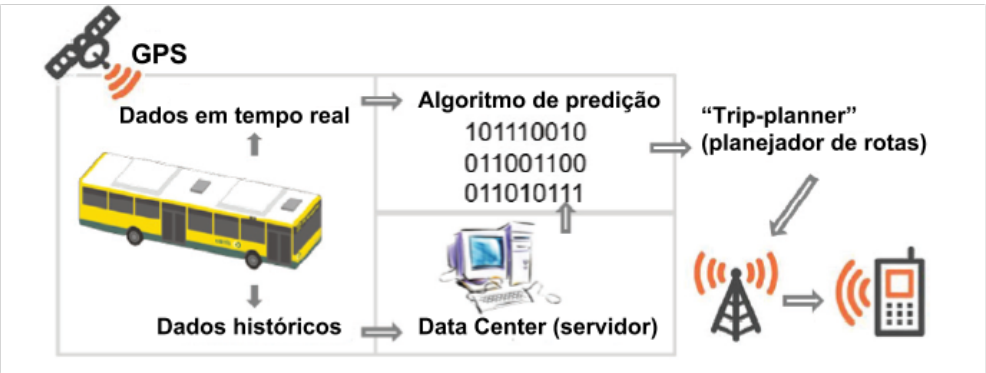

Figura 2: Arquitetura do *Trip-planner.*
Fonte: Adaptado de (ALVES; MARTINEZ; VIEGAS, 2012)

3 METODOLOGIA

Neste capítulo apresenta-se a metodologia a ser empregada a fim de alcançar os desígnios do projeto. Para tanto, apresenta-se as arquiteturas de rede a serem empregadas e, na sequência, as tecnologias, tais como o ambiente de programação, a linguagem de programação, mecanismos para controle de versão e, por fim, a plataforma de software.

3.1 FUNDAMENTOS

Nesta seção serão apresentados os fundamentos a serem empregados no projeto.

3.1.1 ARQUITETURA DTN E ARQUITETURA P2P

Sistemas P2P (*Peer-to-Peer*) são capazes de fornecer recursos de rede com sobreposição e auto-organização distribuída, com o intuito de prover uma distribuição eficiente dos dados. O mecanismo básico de funcionamento consiste em pares, nos quais uma entidade presta e ao mesmo tempo consume os recursos oferecidos por outras entidades que compõe o sistema. Outras características que estes sistemas apresentam, normalmente, são auto-organização, tolerância a falhas e escalabilidade (CACCIAPUOTI; CALEFFI; PAURA, 2011).

Em contrapartida, sistemas baseados em DTN (*Delay Tolerant Networks*) são construídos para operar sobre redes sem fio, com múltiplos saltos. A arquitetura visa oferecer conectividade mesmo se uma comunicação fim a fim não exista por um (in)determinado período de tempo e, assim sendo, os nós intermediários da rede devem armazenar os dados até uma nova oportunidade de comunicação seja possível (CACCIAPUOTI; CALEFFI; PAURA, 2011).

Um paradigma típico para as DTNs é o chamado *store-carry-forward* (armazenar, transportar e encaminhar) que consiste na armazenagem de uma mensagem em um nó, o transporte da mesma por este nó e a sua posterior distribuição para outros nós da rede, tão logo surja uma oportunidade para tal (CACCIAPUOTI; CALEFFI; PAURA, 2011). Um

exemplo simples é exposto na sequência. Supondo que um nó **S** tenha que se comunicar com um nó **D**, mas não existe um caminho direto entre os dois; por consequência disso, o nó **S** terá que armazenar a mensagem. No entanto, **S** consegue se comunicar com um terceiro nó **R**, que por sua vez se comunica com o nó **D**. O nó **S** pode então encaminhar a mensagem para o nó **R**, que irá repassá-la para o nó **D** (CACCIAPUOTI; CALEFFI; PAURA, 2011).

Como consequência dos dois mecanismos apresentados, no escopo deste exposto decidiu-se pelo uso de uma arquitetura de rede que combina os dois modelos apresentados anteriormente. Apesar de algumas disparidades entre os dois paradigmas (a mais notável: operarem em camadas distintas de rede) (CACCIAPUOTI; CALEFFI; PAURA, 2011), usar-se-á os dois mecanismos em conjunto a fim de oferecer uma arquitetura de rede decentralizada e com boa conectividade a todos os supostos usuários do aplicativo. Ademais, o fato de ambos trabalharem em ambientes decentralizados, auto-organizados e distribuídos (CACCIAPUOTI; CALEFFI; PAURA, 2011) tornam-se bastantes interessantes para o escopo do trabalho, já que se encaixa perfeitamente no ambiente em que o aplicativo desenvolvido funcionará.

Outra característica interessante, consiste na possibilidade de operação em mecanismos de rede que não são baseados em meios físicos para propagar informações, como Wi-Fi e Bluetooth, por exemplo. Essa possibilidade é ideal para smartphones, dado que uma gama abundante desses aparelhos possui no mínimo um desses recursos disponíveis.

3.2 TECNOLOGIAS

A presente seção refere-se às principais tecnologias a serem empregadas no desenvolvimento do projeto, tais como ambientes de desenvolvimento, linguagens de programação, plataforma de software e plataforma de hardware.

3.2.1 AMBIENTE DE DESENVOLVIMENTO

Pelo fato de o projeto envolver um aplicativo para o sistema operacional Android, é necessário utilizar um ambiente de desenvolvimento específico para essa plataforma.

O *site* oficial para desenvolvimento de aplicativos Android, Android Developers[1], disponibiliza algumas ferramentas para a concepção e teste de aplicativos para a plataforma.

[1] O site pode ser acessado em http://developer.android.com/

É possível utilizar um *plug-in*, juntamente com o Android SDK, e integrá-los em uma instalação existente da conhecida IDE chamada Eclipse, realizando as devidas configurações manualmente. Uma segunda opção, e que torna a preparação do ambiente de desenvolvimento mais simplificada, é baixar um *bundle* que já inclui o Android SDK, a IDE Eclipse, o *plug-in*, ferramentas de desenvolvimento e uma imagem de sistema para emular um dispositivo Android no Desktop. Dessa forma, é possível depurar aplicações Android diretamente no PC, antes de enviá-lo à um dispositivo real. A terceira opção é utilizar uma IDE alternativa chamada *Android Studio*, que atualmente encontra-se em uma versão Beta.

Por experiências dos integrantes com o referido *bundle*, em projetos anteriores, optou-se por utilizar o mesmo no desenvolvimento deste projeto.

Para testes de software será utilizada a emulação do dispositivo Android provida pelo *bundle*, até onde for possível. É dito isso pois o projeto envolve o uso de GPS, Google Maps e compartilhamento de informações entre usuários (*crowdsourcing*), o que não é possível de ser feito apenas com a utilização de um emulador. Dessa forma, será necessário testar a aplicação em um dispositivo real, como um *smartphone* ou um *tablet*, com sistema operacional Android.

3.2.2 LINGUAGEM DE PROGRAMAÇÃO

Em relação à linguagem de programação, não há a necessidade de fazer um levantamento dentre as linguagens de programação existentes atualmente e escolher qual será a mais adequada ao projeto. Uma vez que o projeto tem como foco o desenvolvimento de uma aplicação Android, é mandatório que a mesma seja desenvolvida utilizando a linguagem de programação Java juntamente com a API (que contém bibliotecas e ferramentas) incluída no Android SDK. A princípio, não será utilizada nenhuma biblioteca de terceiros (*third-party*).

3.2.3 CONTROLE DE VERSÃO

Será utilizado um sistema de controle de versão (CVS, *Control Version System*) para o versionamento do código-fonte do aplicativo Android. Vários são os sistemas de versionamento de arquivos, dentre os quais os mais comuns são o SVN e Git. Para este último, existem repositórios de código-fonte gratuitos, como o GitHub (https://github.com) e o BitBucket (https://bitbucket.org).

O ideal seria manter um servidor próprio com o SVN ou o Git instalados. No entanto, isso poderia envolver custos adicionais, pois seria necessário deixar uma máquina dedicada para o controle de versão e acessível para todos os integrantes do projeto, além de realizar *backups* periódicos.

O GitHub permite que qualquer pessoa crie uma quantidade ilimitada de repositórios gratuitamente e compartilhe-o com quantos colaboradores for necessário. No entanto, o código-fonte fica disponível publicamente, para qualquer um acessar. Para projetos acadêmicos (como TCCs, teses de mestrado e doutorado), isso pode não ser indicado. O GitHub também oferece repositórios privados, com acesso restrito, mas apenas a partir da assinatura de um plano.

Em contrapartida, o BitBucket oferece uma quantidade ilimitada de repositórios privados gratuitamente, permitindo até cinco colaboradores em um projeto. Como o projeto é constituído de dois integrantes e levantou-se a necessidade de versionar o código-fonte sem disponibilizá-lo publicamente, optou-se por utilizar o BitBucket como repositório de código-fonte e o Git como como sistema de controle de versão.

3.2.4 PLATAFORMA DE SOFTWARE

Em relação ao sistema operacional Android, levanta-se outra questão. O Android, como sistema operacional, possui diversas versões desde o seu lançamento. Cada uma dessas versões acaba por incluir novas funcionalidades não apenas para o usuário final, mas também, para os desenvolvedores. É certo que existe a chamada retrocompatibilidade, que permite que a grande maioria das aplicações desenvolvidas para uma versão mais recente do S.O., execute sem maiores problemas em uma versão mais antiga. No entanto, para que isso ocorra, muitas vezes a aplicação fica limitada, pois o desenvolvedor é forçado à desabilitar certos recursos que não são compatíveis, de forma alguma, com versões mais antigas da plataforma.

No momento da escrita desta proposta, a versão mais recente do Android é a versão 4.4 (KitKat). A maioria dos dispositivos disponíveis no mercado – *smartphones* e *tablets* – possuem uma versão do Android 4.0+ instalada. A mais recente versão do S.O., a versão 5.0 (Lollipop), está prestes a ser lançada, e apenas alguns poucos dispositivos já existentes no mercado receberão a atualização para o novo sistema.

A Tabela 1 fornece informações referentes ao *market share* das versões do Android, desde a 2.2 (Froyo) até a 4.4 (KitKat). Os dados foram coletados através da aplicação Google Play Store em um período de sete dias, finalizando no dia 03 de novembro de 2014.

É possível verificar o percentual de quantos dispositivos executam uma determinada versão do sistema operacional. Essas informações são úteis para que o desenvolvedor determine qual versão da plataforma terá como "alvo".

Tabela 1: Market Share das versões do Android. Fonte: (Android Developers, 2014b)

Versão	**Codinome**	**API**	**Distribuição**
2.2	Froyo	8	0,6%
2.3.3 - 2.3.7	Gingerbread	10	9,7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	8,5%
4.1.x	Jelly Bean	16	22,8%
4.2.x	Jelly Bean	17	20,8%
4.3	Jelly Bean	18	7,3%
4.4	KitKat	19	30,2%

Observa-se na tabela que cerca de 10,4% usuários usa um dispositivo com uma versão antiga da plataforma, de codinome Froyo (2.2) ou Gingerbread (2.3.3 - 2.3.7). Uma parcela ainda menor utiliza as versões Ice Cream Sandwich (4.0.3 - 4.0.4), 8,5%. Grande parte dos usuários – cerca de 50% – utiliza o Android Jelly Bean (4.1.x - 4.3) e 30,2% utiliza a última versão lançada, KitKat (4.4).

Portanto, devido à popularidade da versão Jelly Bean (4.1.x - 4.3) do sistema operacional Android, o desenvolvimento do projeto terá como foco dispositivos que executam, no mínimo, esta versão da plataforma.

3.2.5 PLATAFORMA DE HARDWARE

Para o projeto em questão não será necessária a utilização de uma plataforma de hardware, pois a aplicação executará diretamente sobre um dispositivo móvel, tal como um *smartphone* ou um *tablet*.

4 RECURSOS DE HARDWARE E SOFTWARE

O presente capítulo menciona os recursos de *hardware* e *software* necessários para o desenvolvimento do projeto.

4.1 RECURSOS DE HARDWARE

Uma vez que o projeto diz respeito à uma aplicação Android que executará sobre um dispositivo móvel (*smartphones* e *tablets*), não é necessário a compra e/ou o uso de componentes de hardware (microcontroladores, circuitos integrados, fontes de alimentação, bateria, entre outros) para o desenvolvimento do projeto.

4.2 RECURSOS DE SOFTWARE

Os recursos de software englobam ambientes de desenvolvimento, linguagem de programação e plataformas de software.

Conforme citado no capítulo anterior, para a programação e teste da aplicação Android será utilizado o ambiente de desenvolvimento **Eclipse ADT**, que nada mais é que um *bundle* (pacote) que inclui diversos componentes necessários para a concepção do aplicação. Segundo o site oficial para desenvolvimento de aplicações Android, Android Developers (2014a), os componentes inclusos são os seguintes:

- Ambiente de desenvolvimento Eclipse + *plugin* ADT (*Android Developer Tools*);
- Android SDK (*Software Development Kit*);
- Ferramentas adicionais para a plataforma Android;
- Uma versão da plataforma Android (geralmente a mais recente);
- Uma versão da imagem do sistema Android para o emulador.

É possível realizar o *download* do *bundle* gratuitamente no Android Developers, no seguinte link: <http://developer.android.com/sdk/index.html>. Há versões disponíveis para Windows e Linux, tanto 32-bit quanto 64-bit. Para Mac OS X, apenas a versão 64-bit encontra-se disponível.

Segundo Sims (2014), a linguagem de programação "oficial" para o desenvolvimento de aplicações Android é Java. Grande parte das aplicações são escritas em Java e suas API (*Application Programming Interface*) são projetadas para serem chamadas primeiramente por Java. É possível desenvolver utilizando linguagens como C e C++, mas isso é algo que a própria Google, desenvolvedora do Android, não incentiva (SIMS, 2014).

Pode-se fazer o *download* de várias versões do JDK (*Java Development Kit*) gratuitamente a partir do site oficial da Oracle: <http://oracle.com>. O JDK é necessário para a compilação de aplicativos na linguagem Java e por consequência, o Eclipse ADT irá utilizá-lo juntamente com o Android SDK para compilação da aplicação Android.

Durante todo o desenvolvimento do projeto, será utilizado o Git como sistema de controle de versão, conforme citado no capítulo anterior. Será criado um repositório privado no BitBucket (<https://bitbucket.org>) para código-fonte da aplicação Android. Outro repositório foi criado no GitHub (<https://github.com>) para o "versionamento" desta proposta, que está sendo escrita em LaTeX. A criação dos repositórios, tanto no BitBucket quanto no GitHub, é gratuita.

Por fim, a plataforma de *software* sobre a qual executará a aplicação Android, é o próprio sistema operacional Android, disponível em diversos *smartphones* e *tablets* existentes no mercado. Conforme estudo apresentado no capítulo anterior, o desenvolvimento da aplicação Android terá como foco sistemas operacionais que executam, no mínimo, a versão 4.1.x (Jelly Bean).

5 VIABILIDADE E CRONOGRAMA PRELIMINAR

Neste capítulo serão apresentadas a viabilidade técnica e financeira do projeto, bem como o cronograma preliminar para o desenvolvimento do mesmo.

5.1 VIABILIDADE

Uma vez que o projeto envolve programação em Java, é altamente viável tecnicamente, pois ambos os integrantes são familiarizados com esta linguagem de programação. Além disso, a aplicação derivada do projeto executará em *smartphones* e *tablets* com sistema operacional Android, muito popular e disseminado entre a população nos dias de hoje. Os próprios integrantes do projeto possuem tais dispositivos, o que facilita ainda mais o teste e a constatação da aplicação prática do projeto.

Além disso, o projeto é viável financeiramente. Segundo a Prof^a. Dr^a. Ana Cristina Kochem Vendramin, orientadora deste projeto, não será necessária a compra de equipamentos e, uma vez que o projeto envolverá apenas *software*, com a utilização de ferramentas gratuitas ou livres – conforme já exposto neste documento – os custos financeiros serão mínimos ou nulos.

5.2 CRONOGRAMA PRELIMINAR

A Tabela 2 fornece informações acerca das etapas necessárias para o desenvolvimento do projeto, bem como os seus tempos de duração estimados.

Tabela 2: Cronograma preliminar. Fonte: Autoria própria.

Número	**Etapa**	**Dias**
1	Estudo de APIs e do Android SDK	10
2	Projeto de software (modelagem com UML)	10
3	Implementação do projeto de software	30
4	Testes do protótipo utilizando um dispositivo móvel – emulado – com sistema operacional Android	7
5	Testes do protótipo utilizando um dispositivo móvel – real – com sistema operacional Android	7
6	Finalização do artefato	14
7	Finalização da documentação	14
	Total	92

6 CONTEXTO

O Contexto do projeto se aplica à cidade de Curitiba, em especial aos usuários de transporte público da cidade, haja vista que o aplicativo visa informar, com a maior precisão possível, a localização dos automóveis que operam em uma determinada linha de ônibus.

Apesar do caráter altruísta do projeto, o mesmo desenvolver-se-á sem qualquer ligação externa aos elementos do grupo ou fora do ambiente da UTFPR. Logo, é um aplicativo de propósito independente, mas com foco voltado à população em geral.

Como consequência, a alocação de recurso destinados ao progresso do projeto será de responsabilidade de seus desenvolvedores, exclusivamente, assim como a responsabilidade por sua construção.

7 CONCLUSÃO

Sabe-se que transporte coletivo é um tema em voga na maioria das cidades brasileiras, e também é uma área na qual aloca-se grande quantidade de recursos. Em especial, na cidade de Curitiba, tida por muitos anos como possuidora do melhor transporte público do país, há uma preocupação grande nesse aspecto.

Ao longo do ano é possível visualizar por toda cidade, campanhas publicitárias voltadas para esse propósito, tendo em vista conscientizar o usuário do transporte coletivo acerca de alguns comportamentos, e também, tentando melhorar a qualidade do serviço prestado. Verifica-se também que as tentativas de melhora do transporte coletivo não partem apenas da prefeitura da cidade, como documentou-se nesse exposto, um grupo de desenvolvedores buscou materializar um aplicativo que mostrasse, em tempo real, a localização de cada ônibus em uma(s) determinada(s) linha(s) de interesse. No entanto, a aplicação apresentava um funcionamento um tanto instável, já que dependia dos serviços de um servidor da URBS para seu funcionamento correto.

Numa tratativa que visa resolver esse problema, elaborou-se um projeto com intuito de elaborar um aplicativo que funcione sob uma arquitetura *ad-hoc* e utilize recursos disponíveis em um *smartphone*, e portanto, presente no dia a dia da maioria dos usuários de ônibus na cidade. Entretanto, na acepção da proposta de projeto, encontrou-se certa dificuldade em definir o que o aplicativo ia fazer, quais seriam os recursos existentes no programa e principalmente, a forma de propagar a informação que faria o sistema, como um todo, funcionar.

A escolha para a arquitetura de rede foi pensada de modo a aliar conceitos de DTN e P2P. A priori, a ideia apresentou-se confusa, já que as duas abstrações parecem paradoxais, todavia, existem registros na literatura de algoritmos que funcionam com essas características. Passados esses questionamentos iniciais o projeto progrediu de forma tranquila. A equipe trabalhou em conjunto quase na totalidade do tempo, o que facilitou a tomada de decisões e resolução de conflitos.

No que tange projetos futuros, um enfoque interessante que poderia ser discutido apoia-se num mecanismo que garanta algum tipo de redundância na rede de comunicação. De certa forma, isso suportaria um prisma egrégio do projeto que é a atualização das informações em tempo real, ou seja, se uma abordagem falhar, existiria outra para garantir a propagação dos dados em um tempo exequível. Outra singularidade consiste em tornar acessíveis aos usuários, todas as linhas de ônibus cadastradas na URBS, sejam elas metropolitanas ou urbanas, de forma que haja cobertura total de informações em toda a cidade.

Por fim, busca-se com esse projeto fornecer um mecanismo de fácil manipulação e que traga benefícios à população em geral.

REFERÊNCIAS

ALVES, D.; MARTINEZ, L. M.; VIEGAS, J. M. Retrieving real-time information to users in public transport networks: An application to the lisbon bus system. *Procedia - Social and Behavioral Sciences*, v. 54, n. 0, p. 470 – 482, 2012. ISSN 1877-0428. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877042812042279>.

Android Developers. *Android SDK*. 2014. Disponível em: <http://developer.android.com/sdk/index.html>. Acesso em: 21 de novembro de 2014.

Android Developers. *Dashboards*. 2014. Disponível em: <https://developer.android.com/about/dashboards/index.html>. Acesso em: 09 de novembro de 2014.

CACCIAPUOTI, A. S.; CALEFFI, M.; PAURA, L. Mobile p2p: peer-to-peer systems over delay tolerant networks. *Delay Tolerant Networks: Protocols and Applications*, p. 1–35, 2011.

HOWE, J. The Rise of Crowdsourcing. *Wired Magazine*, v. 14, n. 6, 06 2006. Disponível em: <http://www.wired.com/wired/archive/14.06/crowds.html>.

KANHERE, S. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. In: *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*. [S.l.: s.n.], 2011. v. 2, p. 3–6.

PAN, G. et al. Trace analysis and mining for smart cities: issues, methods, and applications. *Communications Magazine, IEEE*, v. 51, n. 6, p. 120–126, 2013. ISSN 0163-6804.

PARADELLS, J. et al. Infrastructureless smart cities. use cases and performance. In: *Smart Communications in Network Technologies (SaCoNeT), 2014 International Conference on*. [S.l.: s.n.], 2014. p. 1–6.

SCHUURMAN, D. et al. Smart Ideas for Smart Cities: Investigating Crowdsourcing for Generating and Selecting Ideas for ICT Innovation in a City Context. *JTAER*, v. 7, n. 3, 2012. Disponível em: <http://dblp.uni-trier.de/db/journals/jtaer/jtaer7.html\#SchuurmanBMM12>.

SIMS, G. *I want to develop Android Apps - What language should I learn?* 2014. Disponível em: <http://www.androidauthority.com/want-develop-android-apps-languages-learn-391008/>. Acesso em: 21 de novembro de 2014.

SUJATHA, K. et al. Design and Development of Android Mobile based Bus Tracking System. *First International Conference on Networks & Soft Computing (ICNSC)*, p. 231–235, 2014.

TECHTUDO. *Waze*. 2014. Disponível em: <http://www.techtudo.com.br/tudo-sobre/waze.html>. Acesso em: 29 de outubro de 2014.

WALRAVENS, N. Mobile Business and the Smart City: Developing a Business Model Framework to Include Public Design Parameters for Mobile City Services. *Journal of theoretical and applied electronic commerce research*, scielocl, v. 7, p. 121 – 135, 12 2012. ISSN 0718-1876. Disponível em: <http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-18762012000300011&nrm=iso>.