Matheus Villela Moreira

Aplicativo para plataforma Android: "BusMaps", informações das linhas de ônibus da cidade de Florianópolis

Florianópolis

2013

Matheus Villela Moreira

Aplicativo para plataforma Android: "BusMaps", informações das linhas de ônibus da cidade de Florianópolis

Universidade Federal de Santa Catarina – UFSC

Orientador: Raul Sidnei Wazlawick

Florianópolis

2013

Matheus Villela Moreira

Aplicativo para plataforma Android: "BusMaps", informações das linhas de ônibus da cidade de Florianópolis

Trabalho aprovado. Florianópolis, 2013:

Raul Sidnei Wazlawick
Orientador

Antonio Carlos Mariani
Membro da Banca

José Eduardo De Lucca
Membro da Banca

Florianópolis
2013

Dedico esse trabalho à todos os usuários do transporte público da cidade de Florianópolis, que por necessidade ou opção melhoram a situação da mobilidade na cidade ao não colocar um carro a mais nas ruas.

Agradecimentos

Gostaria de agradecer à Andrea, minha mãe, que sempre me apoiou durante os estudos e que foi compreensiva quando desisti de ser Engenheiro e parti para a Computação. Também gostaria de agradecer à todos os membros do curso de Ciências da Computação, alunos e professores, em especial ao professor Raul por aceitar ser meu orientador em condições bem adversas. E não poderia deixar de agradecer ao nosso coordenador, e amigo, professor Mazzola, que me apoiou durante meus vários pedidos de prorrogação de prazo para conclusão do curso.

Resumo

Uma das soluções mais efetivas para melhoria da mobilidade urbana nos grandes centros urbanos é a melhoria do transporte público e o incentivo para que as pessoas o usem. A cidade de Florianópolis não possui trem ou metrô, o que torna o transporte por ônibus o único transporte público com expressividade na capital. A inexistência de uma malha cicloviária adequada, aliada à cultura do brasileiro de ver a bicicleta apenas como lazer e não como locomoção, colocam ainda mais importância na utilização de ônibus como solução para melhoria da mobilidade na capital.

O que temos hoje de informação sobre as linhas de ônibus para os usuários são os horários de partidas dos ônibus fixados nos terminais, além disso a prefeitura fornece em sua página na web a possibilidade de consulta desses horários. Não existe uma *API* específica pra que consultas sejam feitas, e embora as informações contenham os itinerários não existe uma representação visual em mapa dos mesmos, tornando difícil entender a rota de um ônibus pra quem não conhece muito bem a cidade.

Esse trabalho envolve a criação de um aplicativo para dispositivos móveis, especificamente para a plataforma *Android*, que além de fornecer as informações disponíveis publicamente pela prefeitura mostre as rotas das linhas visualmente em um mapa para o usuário.

Palavras-chaves: Dispositivos Móveis; Android; Ônibus; Transporte Público; Mobilidade; Florianópolis; Mapa; MapsForge; OpenStreetMap;

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

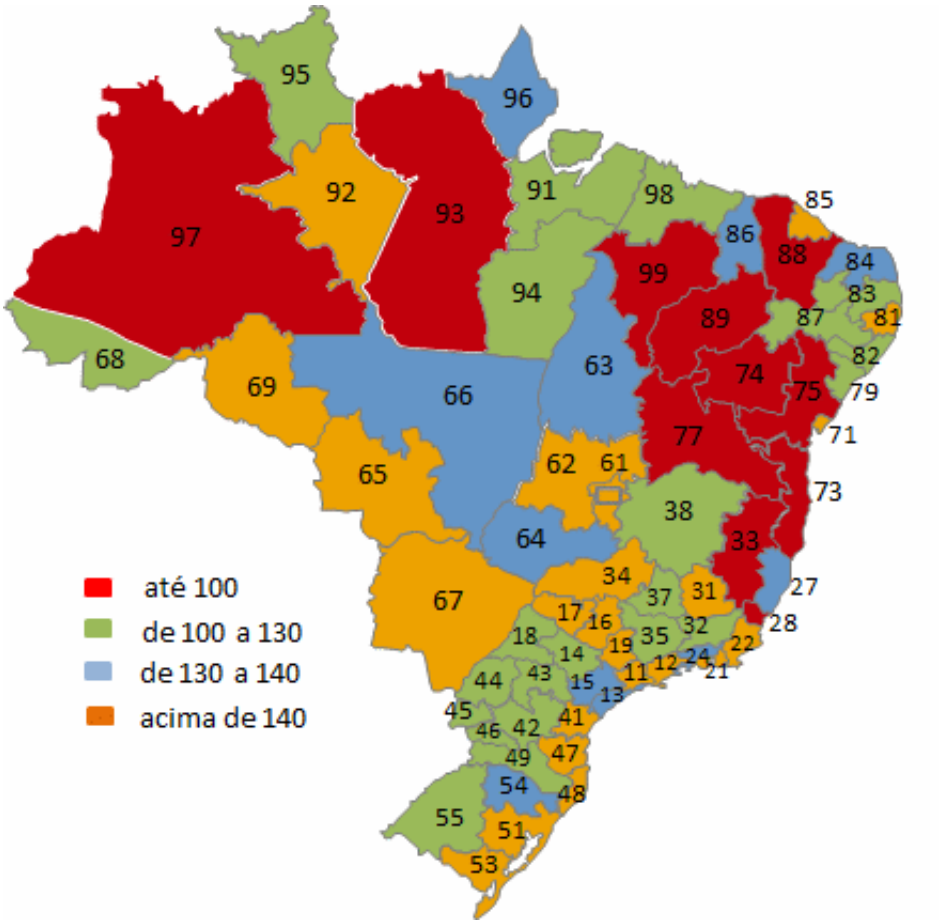
PMF	Prefeitura Municipal de Florianópolis
GTFS	General Transit Feed Specification
XML	eXtensible Markup Language
KML	Keyhole Markup Language
HTML	HyperText Markup Language
LRU	Least Recently Used
API	Application Programming Interface
DDD	Discagem Direta a Distância
MIDP	Mobile Information Device Profile
JVM	Java Virtual Machine
JCP	Java Community Process
CPU	Central Processing Unit
SAX	Simple API for XML
RAM	Random Access Memory
SO	Sistema Operacional
GPS	Global Positioning System
API	Application Programming Interface

Sumário

1 Introdução

Possuir um aparelho móvel se tornou algo comum para os brasileiros, segundo Anatel (2010) em 2010 passamos a marca de 1 linha telefônica móvel por habitante no país, atualmente segundo a mesma Anatel (2013) são 1,3545 por habitante.

Figura 1 – Densidade do Celular por Código DDD em linhas/100 habitantes - Ago/13



Fonte: Estatísticas... (2013)

Segundo pesquisa realizada pelo CETIC.br (2013) entre outubro de 2012 e fevereiro de 2013 na região sul do Brasil, apenas 17% da população não têm nenhuma linha de telefone celular.

Mas nos últimos anos estamos passando por uma segunda grande mudança na área, telefones com capacidades computacionais avançadas, denominados *Smartphones*, estão cada vez mais populares e vêm ganhando espaço no mercado sobre modelos mais simples. De acordo com o IDC (2013) no primeiro quarto fiscal de 2013, pela primeira vez a venda dos denomidados *Smartphones* ultrapassaram a de modelos simples.

O fácil acesso ao dispositivo, aliado à uma grande quantidade de sensores que os *Smartphones* possuem e a disponibilidade de conexão com a internet, os tornam uma plataforma bem específica de desenvolvimento. Uma pessoa dificilmente tiraria de sua mochila um notebook apenas para conferir uma tabela de horários, mas no caso de um *Smartphone* basta tirar do bolso e com alguns toques na tela ter a informação disponível.

Liderando o mercado de *Smartphones* estão os aparelhos rodando o sistema operacional *Android*, tendo como base o Linux e desenvolvido pela empresa Google o sistema vem tendo um papel fundamental na popularização dos *Smartphones*. Segundo a Gartner (2013) no segundo quarto fiscal de 2013 o Android aumentou ainda mais sua liderança entre os Smartphones, subindo para 79% das vendas.

Segundo (MEDEIROS, 2006), onde foram avaliados os aspectos topológicos e geométricos das mais importantes cidades do Brasil, Florianópolis foi considerada a pior cidade com relação à integração global, índice definido no estudo relacionado à dificildade de se locomover para as vias de trânsito da cidade. Tal estudo já foi usado por jornalistas locais para dizer que Florianópolis tem a pior mobilidade urbana do Brasil, conforme noticiado em Diário... (2009). Embora o estudo não aponte necessariamente isso, é evidente para quem mora na capital que a situação do trânsito é bastante complicada.

O objetivo deste projeto é o desenvolvimento de de um aplicativo para sistemas móveis, destinado ao usuário do sistema de transporte público da cidade de Florianópolis. Durante seu desenvolvimento foram encontrados vários desafios, principalmente com relação à coleta dos dados e definição de um formato adequado para representá-los. Embora soluções voltadas à navegação, onde é apresentado ao usuário como se chega de um ponto a outro, sejam de grande importância esse não é o foco principal do aplicativo desenvolvido nesse trabalho. A proposta desde o princípio foi de facilitar o acesso à informação de quem já é usuário de determinadas linhas, fornecendo informações referentes às tabelas de horário e com o auxílio de um mapa mostrar ao usuário um panorama aproximado de qual é a localização atual do ônibus que ele pretende embarcar.

1.1 Objetivos

1.1.1 Objetivo Geral

- Desenvolver um aplicativo para dispositivos móveis rodando na plataforma *Android*, a modo de facilitar o acesso dos moradores de Florianópolis às informações das linhas de ônibus.

1.1.2 Objetivos Específicos

- Determinar uma estrutura adequada para representação dos dados referentes às linhas de ônibus, considerando as informações que a prefeitura e empresas de Florianópolis fornecem.
- Coletar e adequar os dados das linhas de ônibus da página *HTML* da Prefeitura Municipal de Florianópolis.
- Avaliar e corrigir erros e inconsistências nos arquivos de trajetos disponíveis.
- Estudar a plataforma *Android* e seus componentes utilizados no desenvolvimento do aplicativo.

1.2 Justificativa

Como demonstrado por (FERRIS, 2011), a inclusão de aplicativos voltados à melhorar o acesso à informação dos usuários das linhas de ônibus tem um impacto positivo na região. As pessoas perdem menos tempo esperando o transporte público, usam com mais frequência e acham a experiência mais agradável.

Em Florianópolis existe uma grande carência de soluções na área, tanto a prefeitura como as empresas de ônibus se limitam a fornecer as tabelas de horários em suas páginas na internet. Embora já exista um aplicativo que informa as tabelas de horários a proposta do *BusMaps* vai além, usando de recursos avançados de *Smartphones Android* para mostrar ao usuário mapas e trajetos das linhas, como uma ferramenta não apenas de consulta de horários mas de informação completa de como as linhas da cidade operam.

1.3 Resultados Esperados

- Promover o transporte público em Florianópolis pela melhoria de utilização do serviço pelos usuários do aplicativo.
- Promover a utilização de mapas do OpenStreepMap sobre alternativas comerciais como as oferecidas pelo Google e Bing.
- Criar um produto que futuramente pode ser generalizado pra outras cidades em busca de um mercado consumidor maior.

2 Fundamentação

2.1 Android

Android é uma pilha de software de código aberto para uma larga variedade de dispositivos mobile e um projeto de código aberto correspondente liderado pelo Google, Android... (2013a).

Android fornece uma solução completa para dispositivos móveis: um sistema operacional, middleware e aplicativos chave, Android... (2013b).

De acordo com Android... (2013b) o sistema operacional roda no topo de um núcleo *Linux*, utilizando uma máquina virtual customizada arquitetada para otimizar recursos de hardware e memória em um ambiente de dispositivo móvel.

Problemas com relação à decisões de projeto já fizeram com que o núcleo *Linux* que roda no *Android* tivesse seu código removido do repositório principal do mesmo, tornando o núcleo do *Android* um fork, Computerworld (2010). Em março de 2012, na versão 3.3 do *Linux*, essa situação foi resolvida e as alterações feitas para o *Android* no *Linux* voltaram a serem incluídas no repositório principal novamente, CNET (2012).

A maioria do código da base do sistema *Android* é programado em *C*(41,5%) e *C++*(24,8%), com apenas 17.1% em *Java*, Ohloh (2013). Entretanto, a linguagem primária de desenvolvimento de aplicativos para o sistema é o *Java*, mas não o *Java* da empresa Sun que era utilizado em dispositivos móveis, o *MIDP*. Uma das principais diferenças está no desenvolvimento da *Dalvik*, uma *JVM* própria, além disso o *Android* não é parte do *JCP*, grupo regulamentador da linguagem *Java*, não se prendendo à usar as convenções de interfaces como para o acesso à hardware, CNET (2007).

Já são 900 milhões de dispositivos rodando o sistema, e 975 mil aplicativos publicados no Google Play, Android (2013).

Tendo em vista sua popularidade, aliada aos recursos disponíveis na plataforma, o *Android* foi o sistema escolhido para o desenvolvimento do *BusMaps*.

2.2 Java

Java é uma linguagem de programação para computadores que é concorrente, baseada em classes, orientada a objetos, e é especificamente projetada para ter o mínimo de dependências de implementação possíveis. É destinada a deixar os desenvolvedores de aplicações desenvolverem uma vez e rodar em qualquer local, ou seja, o código que

roda em uma plataforma não precisa ser recompilado para rodar em outra. Aplicativos *Java* são tipicamente compilados em bytecode que pode ser rodado em qualquer *JVM* independente da arquitetura do computador. Wikipedia (2013)

2.3 Dalvik

Dalvik é a *JVM* utilizada no Android, ela foi desenvolvida para rodar em uma *CPU* lenta, com relativamente pouca *RAM* e em um *SO* sem espaço de troca em disco(swap), isso em um hardware movido à bateria. A *Dalvik* é uma máquina de registro, ao contrário das outras *JVMs* que são máquinas de pilha, o que diminui a quantidade de instruções em cerca de 35% e evita acessos à memória desnecessários. Essa diminuição de instruções e a maneira com que os programas compilados para a *Dalvik* são estruturados faz com que um binário para *Android* seja cerca de metade do tamanho de um binário *Java* comum. Borsntein (2008)

2.4 TagSoup

TagSoup é um analizador de gramáticas, é compilante com a família de analizadores *XML SAX*, onde a análise de cada elemento é reportada geralmente apenas uma vez e não é guardada nenhuma informação em memória sobre o elemento. No entanto o *TagSoup* foi criado para análise de *HTML*, na forma que é encontrado na internet, muitas vezes repleto de erros e mal formado estruturalmente. TagSoup... (2013)

2.5 LRU cache

No *LRU* o bloco no cache que não vem sendo usado pelo maior período de tempo é trocado pelo novo. Ele é baseado na observação de que blocos os quais foram referenciados no passado recente têm mais chances de serem referenciados no futuro próximo. Zhou (2001)

Apesar de geralmente utilizado no gerenciamento de memória a nível de hardware, o algoritmo de cache *LRU* também pode ser empregado a nível de software. No *Android* temos uma implementação de *LRU* na classe *android.util.LruCache<K, V>*, a mesma é instanciada passando o tamanho máximo do cache. Em casos como armazenamento de imagens onde o tamanho do objeto pode variar bastante é comum sobreescrever sua função *protected int sizeOf(K key, V value)*, se não for sobreescrito o método retorna o valor 1, fazendo com que o tamanho do cache represente o número máximo de objetos que ele pode armazenar. LruCache... (2013)

2.6 Action Bar

O componente *Action Bar* do *Android*, que é integrado ao sistema de janelas dos aplicativos, fornece uma interface familiar entre diferentes aplicativos onde ficam disponíveis ações de usuário e indicações de navegação. O sistema trata de adequar a *Action Bar* a diferentes configurações de tela, mostrando ícones de ações apenas quando existe espaço entre outras funcionalidades. A *Action Bar* foi adicionada na versão 11 da *API* do *Android*, no *BusMaps* está disponível para aparelhos com *API* inferior à 11 com a utilização da biblioteca *ActionBarSherlock*. Recentemente a *Action Bar* passou a poder ser utilizada em dispositivos com *API* nível 7 ou superior com a biblioteca de suporte do *Android*, fornecida pelo próprio Google e parte oficial do projeto *Android*. Action... (2013)

2.7 Activity

No *Android* uma *Activity* é um componente que fornece uma tela com a qual usuários podem interagir por realizar algo, como discar o telefone, enviar um email, ou visualizar um mapa. Cada *Activity* recebe uma janela na qual ela pode desenhar a interface do usuário. A Janela tipicamente preenche toda a tela, mas pode ser menor que a tela e se posicionar sobre outras janelas. Activities... (2013)

2.8 Mapsforge

O projeto *Mapsforge* fornece software gratuito e livre para aplicativos baseados no OpenStreetMap, no momento possui uma biblioteca para renderização de mapas em dispositivos *Android*. mapsforge... (2013)

Algumas características:

- Formato de arquivos compacto, para renderização rápida de informações do OpenStreetMap.
- Estilos de mapas cutomizáveis via *XML*
- Biblioteca pequena, cerca de 300 KB
- 100% gratuita e de código aberto (licença LGPL3)

2.9 ActionBarSherlock

ActionBarSherlock é uma extensão das bibliotecas de suporte desenvolvida para facilitar o uso da *Action Bar* entre todas as versões do *Android* com o uso de uma única

API.

Apesar da *ActionBarSherlock* ter se tornado obsoleta com a inclusão das suas funcionalidades na biblioteca de suporte oficial do *Android*, isso ocorreu depois do *BusMaps* estar pronto.

3 Estado da Arte

Inúmeros projetos para *Smartphones* semelhantes ao *BusMaps* podem ser encontrados, eles variam bastante na forma com que são implementados mas buscam ajudar o usuário a resolver o mesmo problema, saber o horário do ônibus. Existem soluções que tentam abranger o maior número de localidades possíveis, em muitos casos dependendo da padronização das informações, essas serão aqui classificadas como Soluções Globais. Outras, assim como é o caso do *BusMaps* buscam cobrir apenas uma determinada região, tornando assim possível contornar a não padronização dos dados e se adaptar melhor à realidade da região e usuários.

3.1 Soluções Globais

3.1.1 Google Maps

O Google tem o Programa de parceiros do Google Transit, Google (2013a). Onde os agências de trânsito e prefeituras, ao se cadastrar, podem disponibilizar dados do transporte público para o Google de modo que essas informações estejam disponíveis no Google Maps. Nesse caso as informações precisam ser fornecidas seguindo um formato definido pelo Google, o *GTFS* - Google (2013b).

No *GTFS* todas as rotas são definidas por coordenadas de localização, tornando possível com os dados traçar em um mapa o trajeto de qualquer linha disponível. Além disso as tabelas de horários são completas, ou seja, é fornecido o horário de passagem em cada ponto de todas as linhas. Atualmente mais de 800 prefeituras no mundo fornecem essas informações para o Google, o que torna o *GTFS* o formato mais popular de dados de transporte público. Apesar dessa padronização de dados potencialmente ajudar outros desenvolvedores além do Google, os dados que são enviados nem sempre são disponibilizados publicamente. Ao entrar em contato eletronicamente com as prefeituras de Blumenau(SC), Curitiba(PR) e São José dos Campos(SP), que fazem parte do programa com o Google, pedindo uma forma de acessar os dados não obtive resposta de nenhuma delas.

Embora o Google Maps seja, possivelmente, a solução generalista que cubra a maior quantidade de municípios, as informações disponíveis na versão *Android* referente ao transporte público são bastante limitadas. O aplicativo se limita a dar opções de como ir de um ponto a outro, levando em consideração a hora atual, informando linhas que o usuário deve usar para chegar à determinada localização, e o tempo que isso irá levar.

3.1.2 Moovit

Buscando uma proposta um pouco diferente temos o *Moovit*, criado em 2011 o aplicativo oferece informações em tempo real do transporte público. O aplicativo está disponível na maioria das metrópoles ocidentais, no Brasil está disponível em 9 cidades. Assim como o *BusMaps* o *Moovit* utiliza mapas do OpenStreetMap.

As informações mostradas no aplicativo são um misto de informações estáticas, e informações dinâmicas vindas dos próprios usuários usando o modelo de Crowdsourcing. Além dos usuários poderem, usando o *GPS* e internet de seu aparelho, ajudar a determinar a posição onde um determinado ônibus está é possível comentar e dar notas às linhas, tornando o aplicativo uma mini rede social.

3.2 Soluções Locais

3.2.1 OneBusAway

O *OneBusAway* é um aplicativo de informações do transporte público criado para a cidade de Seatle e região, no estado de Washington nos EUA. O aplicativo foi desenvolvido na Universidade de Washington, é de código aberto, e já foi tema de diversas publicações.

Embora a agência de transporte da região de Seatle tivesse infra-estrutura, disponibilizando informações em tempo real com *GPS* nos veículos desde o final dos anos 90, a forma com que o usuário recebia era muito rudimentar. Essa foi a questão que motivou a criação do projeto, que não somente inclui o aplicativo para *Android* mas também soluções para outras plataformas.

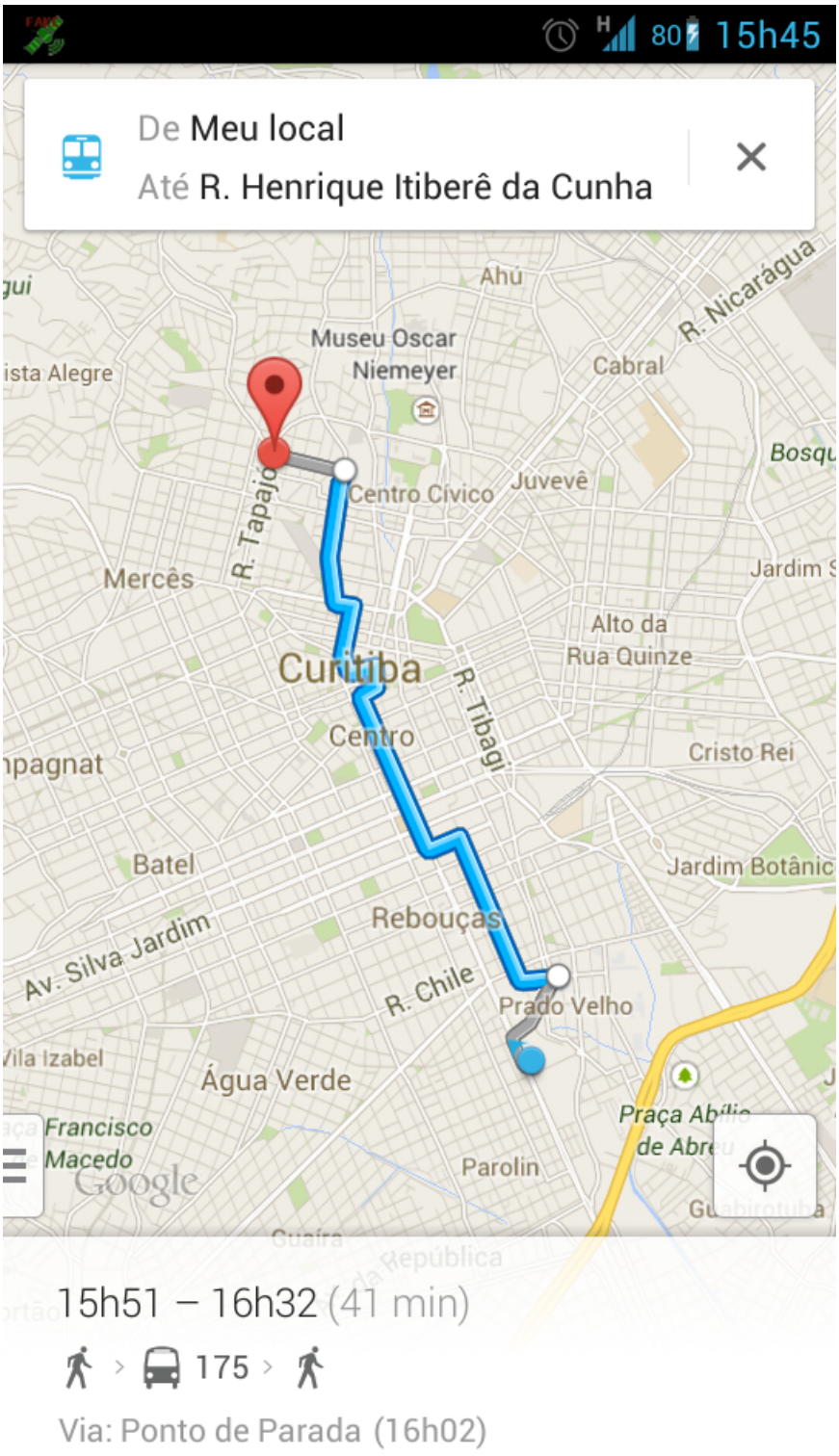
De acordo com (FERRIS, 2011, p. 117) a mudança da satisfação com o transporte público, com o uso do *OneBusAway*, foi positiva para 92% da população local.

3.2.2 MovelBus

> MovelBus é um aplicativo gratuito para celulares compatíveis com Java ou smartphones com Android que contém todos os horários de ônibus cadastrados em um banco de dados interno, possibilitando uma consulta rápida a qualquer momento sem necessidade de acesso a Internet. MovelBus (2013)

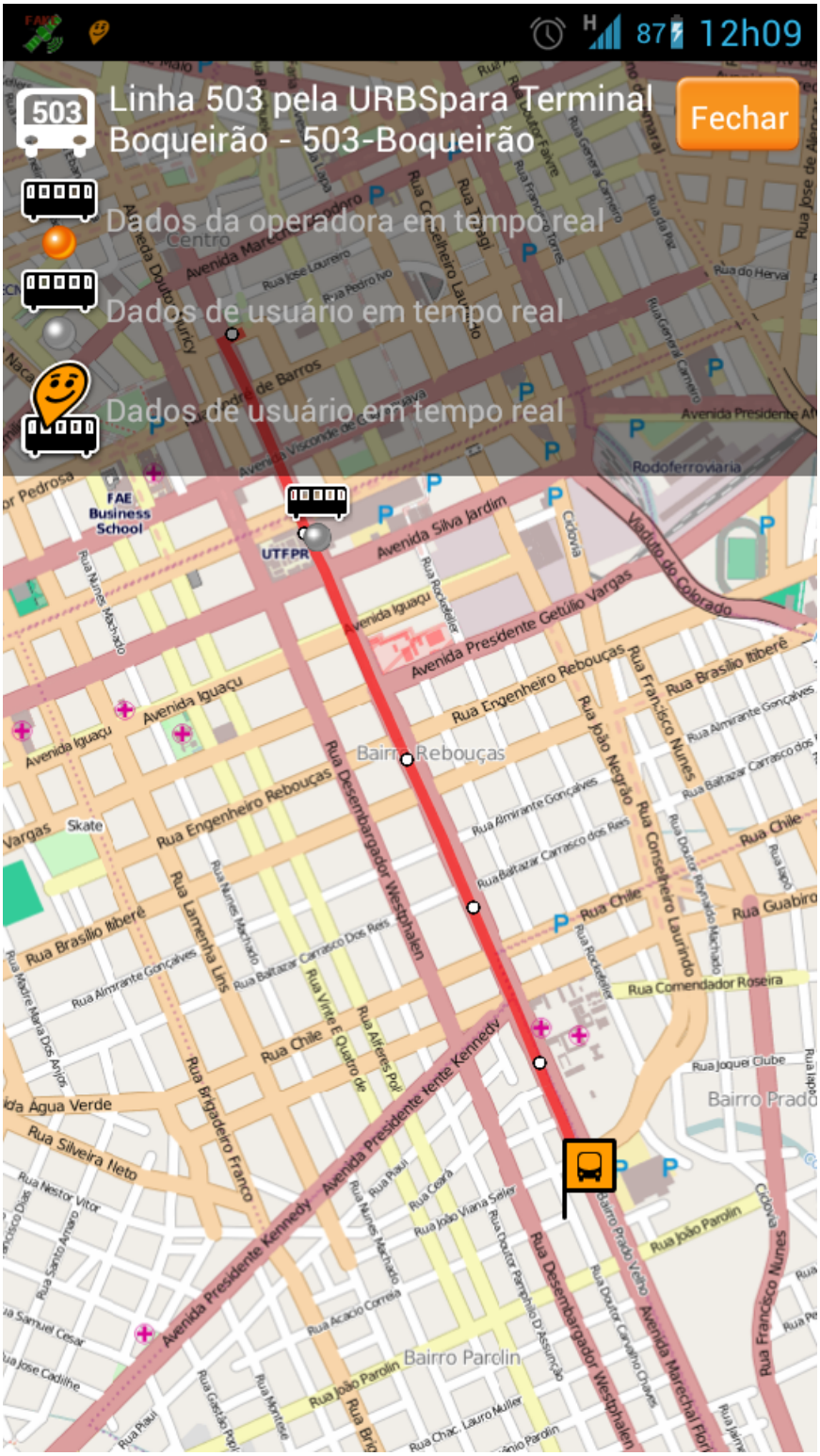
Anteriormente a única opção de aplicativo de transporte público para *Android* na cidade de Florianópolis, além da capital cobre as linhas de ônibus em Biguaçu, Blumenau e Gaspar. O *MovelBus* também é disponibilizado para aparelhos móveis mais simples e em uma versão *WAP*, nesse caso podendo ser acessado por um navegador. O aplicativo se limita a mostrar os horários de partida de cada linha, tendo uma interface bastante simples porém prática para o propósito.

Figura 2 – Google Maps



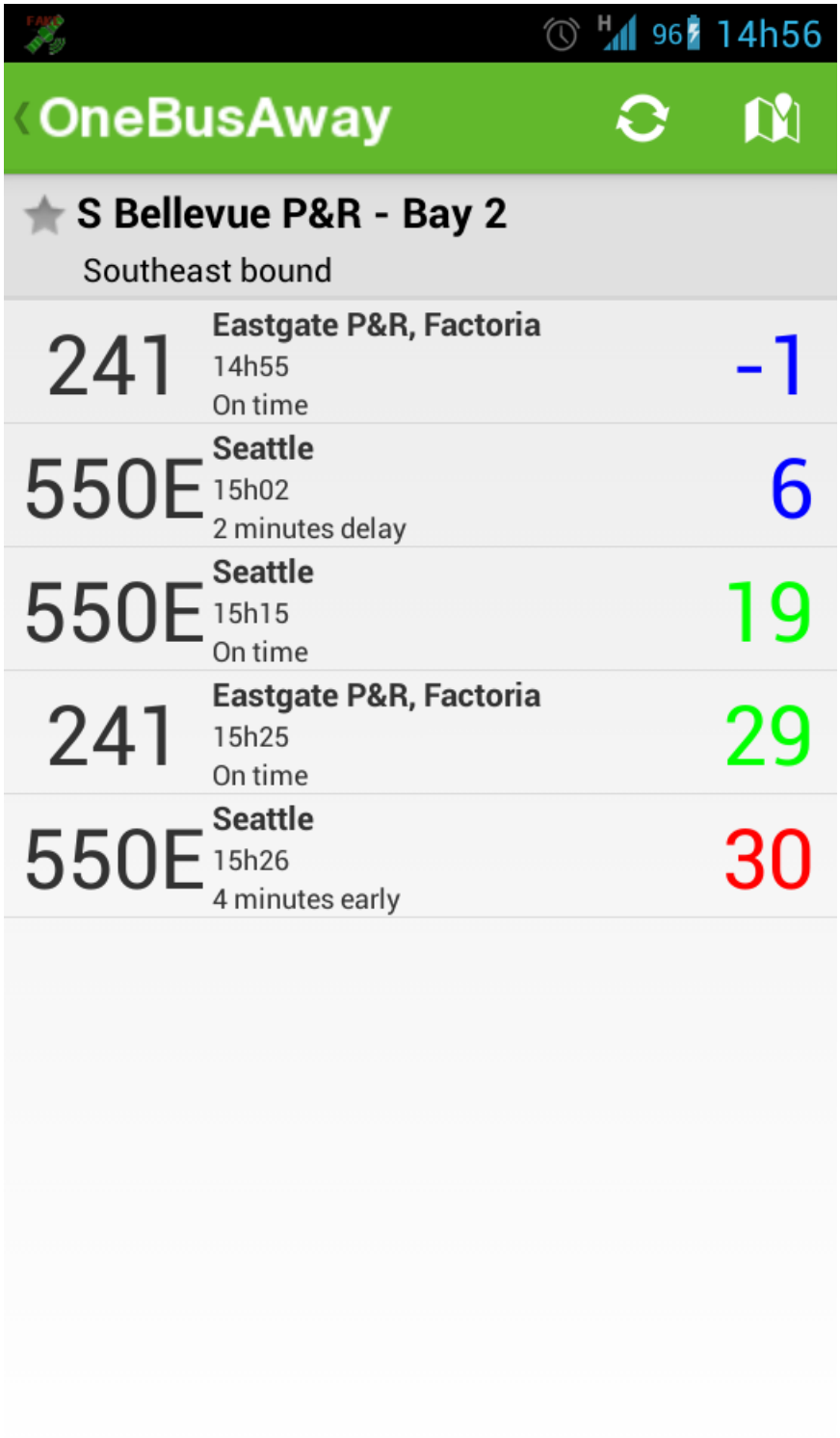
Aplicativo Google Maps rodando em um aparelho Android, a parte em azul se refere à linha de ônibus número 175, os trechos em cinza os que o usuário deve fazer caminhando. Para o usuário é mostrado o tempo total do trajeto, de 41 minutos, que supostamente leva em consideração a velocidade que o ônibus leva pra fazer o percurso naquela hora do dia. O tempo que o usuário espera no ponto para que o ônibus chegue também é levado em consideração. Fonte: o autor

Figura 3 – Moovit



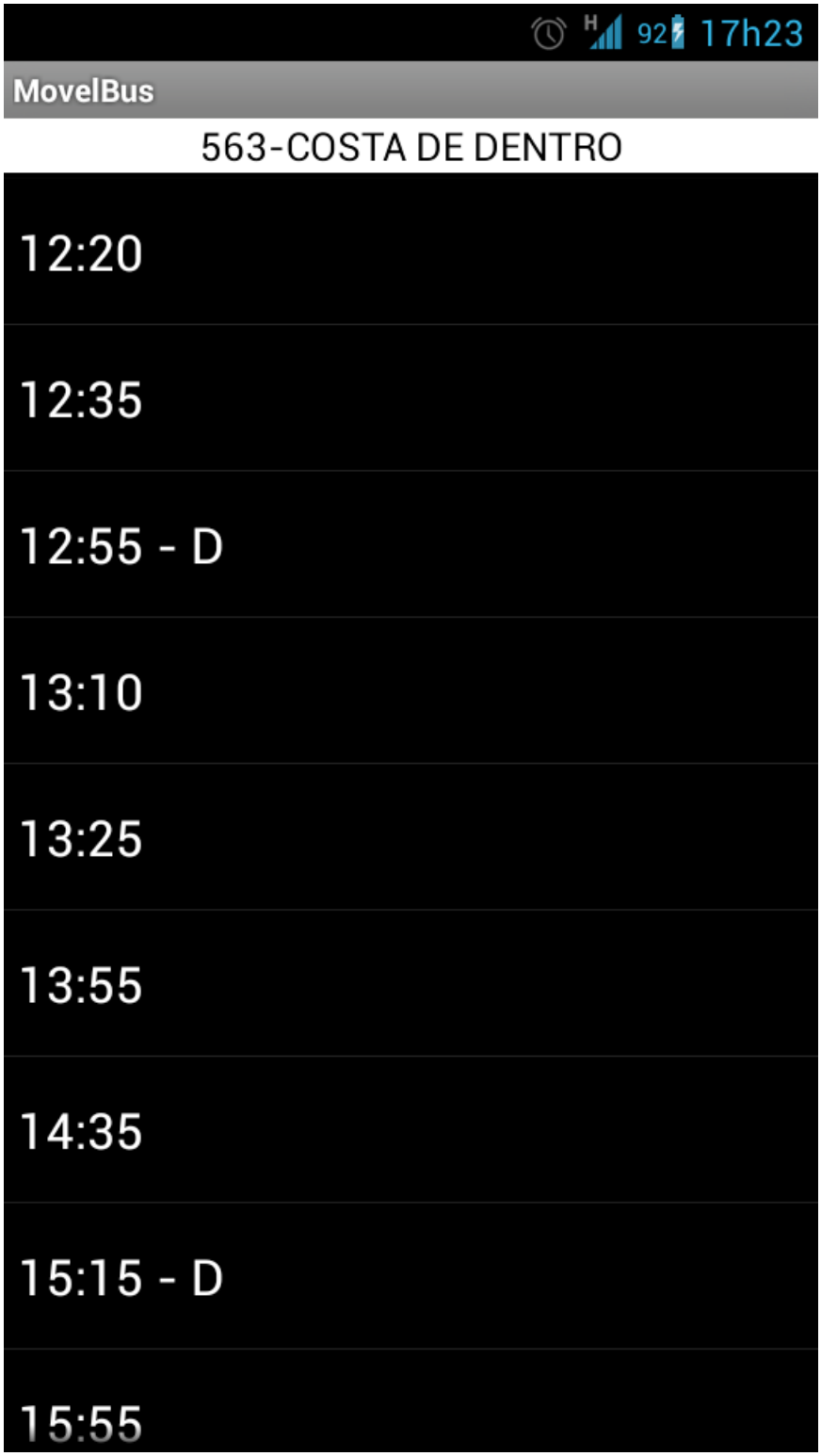
Moovit mostrando uma linha de ônibus da cidade de Curitiba. Apesar do aplicativo no Brasil ter patrocínio da operadora TIM observamos um erro de tradução na segunda legenda, o ônibus com um círculo cinza na verdade indica uma estimativa de posição e não de um usuário em tempo real, que seria a terceira legenda. Fonte: o autor

Figura 4 – OneBusAway



No OneBusAway, após o usuário selecionar um ponto de ônibus no mapa essa tela é apresentada, indicando os ônibus que recém passaram no ponto e os que passarão em um período próximo. É mostrado o quanto cada ônibus está adiantado ou atrasado, usando as informações em tempo real da agência de trânsito para isso. Fonte: o autor

Figura 5 – MovelBus



Principal tela do MovelBus, onde são mostrados os horários de partida de uma linha de ônibus. O "D"que acompanha alguns horários se refere à uma peculiaridade explicada por uma legenda nos dados originais da prefeitura, mas no aplicativo ele é mostrado sem uma explicação do que representa. Fonte: o autor

4 Desenvolvimento do Projeto

Nessa seção será detalhado o processo de desenvolvimento responsável pela criação do aplicativo proposto. Isso envolve os dois principais passos do projeto, o desenvolvimento de um aplicativo capaz de fazer a coleta dos dados da página da prefeitura, e a criação do aplicativo voltado ao usuário final para o sistema *Android*.

4.1 Coleta de Dados

Uma das maiores preocupações com relação ao projeto era da quantidade de interferência necessária para avaliar inconsistências e alterações nos dados das linhas, logo o processo de aquisição desses dados foi feitoda forma mais generalista e automatizado possível.

A prefeitura da cidade de Florianópolis não fornece uma *API*, nem segue nenhum padrão para recuperação das informações necessárias para o aplicativo. O que existe é uma página *HTML* onde o usuário pode navegar e ver essas informações, que se limitam à ônibus de linhas municipais, não cobrindo inter-municipais. Essas informações são as seguintes:

- Horários de Partida
- Preço da Passagem
- Tempo Médio para Realização do Percurso
- Extensão do Percurso
- Itinerário
- Características Específicas(Cadeirante, Carro Surf, etc)
- Sentido

Sendo que nem todas as linhas possuem todas as informações listadas, a página possui erros de codificação e não existe um padrão para as características específicas. Outro problema são informações apresentadas de forma inconsistente, como linhas com o sentido "Bairro -> Centro"e outras com "BC"para indicar a mesma coisa.

Mas um dos maiores problemas é a forma com que o itinerário é apresentado, se trata apenas de uma lista das ruas pelas quais a linha passa. No desenvolvimento do aplicativo final para o usuário precisaríamos do trajeto em formato de rota de GPS

para apresentação do mesmo, para isso existiriam 3 possibilidades: desenhar usando algum software específico de mapas o trajeto, utilizar algum programa que desenhasse esse trajeto baseado no nome das ruas, ou a solução mais simples encontrada que foi de utilizar os mapas disponibilizados na página do MObfloripa que é um portal sobre mobilidade que fornece informações sobre as linhas, com os trajetos desenhados em mapas do Google e que podem ser exportados para formato de trajeto de GPS.

4.1.1 Coleta de Dados da Página da PMF

Atualmente a prefeitura disponibiliza os dados referentes às linhas de ônibus municipais em uma página *HTML*, não existe a opção de coleta dos dados utilizando uma *API* ou acessando uma base de dados diretamente. Logo a única opção prontamente disponível foi de extrair os dados diretamente da página *HTML* da *PMF*.

Para extrair esses dados foi escolhido a criação de um programa em *Java* para isso, mantendo a mesma linguagem de programação que seria usada no aplicativo Android.

A extração de dados de uma página *HTML* pode ser realizada com a ajuda de bibliotecas da parsing disponíveis que ajudam com o processo, a biblioteca *TagSoup* foi a escolhida pela quantidade de referências encontradas sobre a mesma e sua demonstração de adequação para o propósito.

Esse processo de extração consiste em identificar elementos no *HTML* que nos indicam onde estamos na página, exemplificando na página inicial de consulta das linhas da cidades temos um *<select>* com o campo *name* de valor *empresa*, sabemos que todos os *<option>* que estão dentro do *<select>* se referem à cada uma das empresas.

4.1.2 Coleta de Percursos do MObfloripa

A página do *MObfloripa* fornece além de notícias sobre mobilidade na cidade um serviço de consulta sobre as linhas de ônibus da cidade, com um diferencial sobre a página da prefeitura que é o de visualizar o trajeto de cada linha em um mapa do Google.

Ao abrir um mapa com o trajeto de uma linha disponível é possível fazer download desse trajeto, em formato *KML*. Devido à complexidade de como a página do Google é organizada, e ao fato de que atualizações desses trajetos seriam raríssimas se escolheu por fazer download manualmente de cada trajeto. Como o nome do arquivo gerado começa com o número da linha esse processo não foi muito demorado, possivelmente demoraria mais se tentasse automatizar e fazer essa extração programaticamente.

O formato em que os mapas se encontram é o *KML*, que é um *XML* facilmente analisável. No arquivo um trajeto é representado em uma série de localizações, sendo que locais como terminais têm apenas um par latitude/longitude pra representar onde ficam, e uma rua possui usa série de pares latitude/longitude que descrevem o percurso da mesma.

Além disso uma localização como terminal tem uma imagem associada ao mesmo, e uma localização como uma rua uma cor específica que define como será desenhada no mapa.

Os trajetos do *MObfloripa* apresentam alguns problemas que precisaram de tratamento especial. Um dos maiores é que em linhas que não são circulares os percursos de ida e volta estão no mesmo arquivo, sendo que a ida está em uma cor e a volta em outra mas não existe uma padronização de que cor é a ida ou a volta. Outro problema é que em linhas não circulares pontos como terminais estão presentes apenas em um sentido. Por fim como o objetivo deles é apenas a visualização do usuário do mapa nem sempre o sentido dos pontos de longitude/latitude de uma rua estão corretos.

Para o projeto houve a necessidade de separarmos os percursos de ida e volta, tratando os mesmos como percursos distintos. Relacionando os informações de cada linha extraídas da página da *PMF* com o percurso a quantidade de informações para determinar que parte é o percurso de ida e qual o de volta não eram suficientes para a utilização de um processo automatizado. Felizmente como sentidos diferentes são representados por cores distintas bastava determinar uma cor pra ida e outra pra volta, e se estivesse invertida definir aquele percurso como invertido.

Os pontos de localização como terminais que só estavam presentes em um dos sentidos foram adicionados programaticamente quando se tratavam de pontos finais, mas os terminais por onde o ônibus passava no meio do percurso precisaram ser adicionados nos arquivos de percurso manualmente, como não são muitas linhas com essa característica(somente parte das linhas que vão para o norte da ilha) o trabalho foi possivelmente menor que tentar automatizar o processo.

4.1.3 Organização dos Dados

No aplicativo *Android* desenvolvido existem três principais classes que representam dados: *BusLineInfo*, *Route* e *Bound*. Por padronização de projeto, todos os termos que se encontram na página de informações da *PMF* e aqui são usados são mantidos em português, em outras situações os termos utilizados são em inglês. Variáveis que representam valores coletados, mas que o aplicativo não utiliza no momento, estão omitidas de modo a limitar a explicação ao que realmente se encontra implementado no aplicativo.

4.1.3.1 BusLineInfo

Tipos contidos na classe:

```
1     private List<Long> voltaListSemana;
2     private List<Long> voltaListSabado;
3     private List<Long> voltaListDomingo;
4     private List<Long> idaListSemana;
```

```
5     private List<Long> idaListSabado;
6     private List<Long> idaListDomingo;
7
8     private long timeIda;
9     private long timeVolta;
10    private String fullName;
11    private String name;
12    private String number;
13
14    private Route routeIda;
15    private Route routeVolta;
16    private String sentidoIda;
17    private String sentidoVolta;
18
19    private int usedCount;
20    private boolean ida;
21    private boolean favorite;
```

Essa estrutura representa todos os dados referentes à uma linha de ônibus. As 3 últimas variáveis são dados persistidos no banco de dados, *usedCount* indica quantas vezes o usuário já consultou determinada linha, *ida* indica se a última visualização da linha foi no seu sentido de ida ou volta, *favorite* determina se é uma linha que o usuário selecionou como uma de suas favoritas.

As listas representam os horários das linhas, como o aplicativo não mostra no momento peculiaridades de cada horário, como se o ônibus está adequado para cadeirantes, basta armazenar o horário de partida, no caso isso é feito usando a classe *Long* já que uma lista não aceitaria o primitivo *long*.

Todas as variáveis do tipo *String* são definições da própria página da *PMF* sobre detalhes das linhas, *routeIda* e *routeVolta* representam o trajeto que é feito nos dois sentidos, *timeIda* e *timeVolta* o tempo médio que a linha leva para fazer cada rota.O número da linha, definido por *number*, é to tipo *String* pois pode conter letras.

Uma linha que opera de forma circular tem apenas dados referentes à volta preenchidos, linhas consideradas circulares são aquelas que só têm horários de partida de um ponto.

4.1.3.2 Route

```
1     private List<Location> points;
2     private float distance;
```

Uma *Route* possui uma lista de coordenadas geográficas, representadas por objetos da classe interna do *Android android.location.Location*, as coordenadas já estão ordenadas na lista no sentido com que a rota é percorrida pelos ônibus de sua linha.

A variável *distance* indica, em quilômetros, a distância total do percurso de sua *Route*. Esse valor é pré calculado pelo aplicativo que coleta os dados e já vem incluso nos dados do aplicativo *Android*. O custo, a nível de armazenamento, é muito pequeno e se trata de um valor que simplifica o cálculo de estimativas da posição estimada de um determinado ônibus.

4.1.3.3 Bound

```
1     private double up;
2     private double right;
3     private double down;
4     private double left;
```

Bound se trata de um retângulo com lados paralelos aos meridianos e paralelos, suas variáveis servem para definir os limites latitudinais e longitudinais do retângulo. A motivação da criação do *Bound* está na maior facilidade computacional de se encontrar linhas com rotas que passam próximas a determinada localização. Pra chegar à que linhas passam próximas a uma localização, sem algo como o *Bound*, seria necessário calcular a distância entre a localização e cada *Location* contido em cada *Route*. Por outro lado, verificar se uma localização encontra-se dentro de um *Bound* tem um custo computacional bem inferior.

Outra limitação que torna o uso do *Bound* necessário é que os *BusLineInfo*'s não são mantidos todos em memória, mas sim somente os em cache. Logo não existe acesso imediato às listas de *Location*'s contidas nas *Route*'s. Mudar essa estrutura, para uma em que todas as linhas estão sempre em memória, traria pelo menos dois segundos de delay à inicialização do aplicativo, além de acarretar em possíveis problemas de ultrapassagem do limite de memória destinada ao aplicativo.

4.1.3.4 BusLinesCache

Classe que estende *android.util.LruCache<K, V>*, sendo K uma *String* definindo o número da linha, e V uma *BusLineInfo*.

Método get sobreescrito:

```
1     @Override
2     public BusLineInfo get(String key) throws IOException {
3         BusLineInfo info = super.get(key);
4         if (info == null) {
5             File lineFile = new File(this.dir, key + ".bin");
6             info = BusLineInfo.GetBusLineInfo(lineFile);
```

```
7                this.put(key, info);
8            }
9            return info;
10       }
```

As *BusLineInfo*'s são armazenadas em disco no aplicativo, em um formato binário que a função *BusLineInfo.GetBusLineInfo(File file)* trata de transformar em uma *BusLineInfo*. O campo *dir* é passado no construtor da classe, e indica em que diretório os arquivos das linhas estão.

4.2 Desenvolvimento do Aplicativo para Android

4.2.1 Escolha de Bibliotecas

Se tratando de um projeto inicialmente sem fins lucrativos, e sem nenhum financiamento, um de seus pré requisitos é de custar o mínimo possível.

Devido à essa questão a própria viabilidade do mesmo fora definida pela disponilidade de bibliotecas de terceiros de custo zero.

A biblioteca *Mapsforge* foi escolhida para renderização dos mapas. Embora tenha uma grande utilização ela se demonstrou bastante imatura, existindo um conflito entre a documentação e as versões da mesma. As informações contidas na documentação do wiki da mesma eram referentes à versão em desenvolvimento, enquando a última versão estável tinha diferenças consideráveis de interface que mudavam bastante como as implementações são feitas. Devido à dificuldade com isso se escolheu pela versão em desenvolvimento, mesmo sendo considerada instável, a mesma se demonstrou aceitável para o projeto.

Uma das características da *Mapsforge*, de trabalhar com arquivos internos, de pequeno tamanho para os dados dos mapas foi fundamental para o projeto. Após transformado para o formato interno do *Mapsforge* o mapa de Florianópolis fica com pouco mais de 1 MB. Não é necessária conexão com a internet em nenhum momento para o uso da biblioteca.

A biblioteca *ActionBarSherlock* foi incluída por ser a única opção de oferecer a *Action Bar* no aplicativo sem perder uma boa fatia de usuários por não suportar aparelhos antigos. A mesma não apresentou nenhum problema e se demonstrou perfeita para o propósito.

4.2.1.1 Conflitos entre Bibliotecas

Um dos problemas encontrados no desenvolvimento foi de um conflito entre as bibliotecas *MapsForge* e *ActionBarSherlock*, que criou a necessidade de edição e recompi-

lação do código da biblioteca *MapsForge*. Isso ocorreu pois para utilizar ambas as bibliotecas é preciso a nossa *Activity* estendesse uma *Activity* específica da biblioteca, como nenhum dos projetos usa uma versão *Java* que suporte herança múltipla a única solução foi de incorporar o código presente dentro da *Activity* de uma das bibliotecas dentro da outra.

4.2.2 Permissões do Usuário

Quando um usuário vai instalar um aplicativo *Android* ele é informado sobre as permissões que o aplicativo precisa, isso será detalhado em outro momento mas no desenvolvimento do mesmo a ideia era de usar o mínimo possível.

Uma grande vantagem do aplicativo no momento é que ele não precisa de acesso à internet em nenhum momento. Ao começar com o desenvolvimento o mesmo foi desenvolvido fazendo download do mapa e das informações das linhas, isso seguia a ideia da criação de um aplicativo modular, onde o usuário pudesse escolher a sua cidade e aí sim as informações fossem obtidas das internet. Seguindo esse formato o aplicativo ficaria muito complicado inicialmente, e não faria sentido que o usuário escolhesse entre apenas uma cidade. Além disso seria necessário a utilização de um servidor para o download dos mapas e informações. Devido à essas questões foi escolhido por colocar os mapas e informações em arquivos incluidos no pacote de distribuição do aplicativo, o mesmo é distribuído pelo Google Play não causando necessidade de gastos com servidores para distruibuição desses arquivos.

Outra possível permissão que até então foi evitada é a de coletar a informação da posição atual do usuário, necessária para saber o local atual do usuário segundo o GPS. Essa é uma permissão um pouco sensitiva para o usuário, e o ganho de informar o usuário da posição atual dele no mapa possivelmente ainda não justifica a inclusão dessa permissão, considerando que não está ainda implementado um sistema de navegação no aplicativo.

A única permissão que o aplicativo utiliza no momento se deve à uma questão técnica, referente à biblioteca de renderização dos mapas, MapsForge. A mesma precisa que o mapa esteja em um sistema de arquivos, pois precisa de abertura do arquivo em modo de leitura randômica (RandomAccessFile) , o que não é possível quando o arquivo do mapa está dentro do pacote em que o aplicativo é distruibuído.

4.2.3 Desenvolvimento - O Aplicativo

4.2.3.1 Funcionalidades

O aplicativo foi desenvolvido pensando nas seguintes funcionalidades iniciais:

- mostrar os horários de uma linha;
- mostrar no mapa o percurso de uma linha;
- permitir a busca e seleção de uma linha por seu nome ou número;
- possibilidade de favoritar linhas, tendo em vista um acesso mais rápido à linhas de utilização frequente;
- visualização da posição estimada de ônibus que estão fazendo o percurso de uma linha;
- visualização e seleção de linhas que passam próximas a determinado local;

4.2.3.2 Preocupação com Memória

Um ponto de preocupação no aplicativo é com memória, a biblioteca *MapsForge* utiliza muita para renderização do mapa. Isso devido à necessidade de ter pelo menos um bitmap do tamanho da resolução do aparelho aberto, e da limitação que o *Android* limita com relação à heap máxima destinada à cada aplicativo.

Devido à essa limitação se o aplicativo mantivesse todas as informações das linhas em memória seria ultrapassado o limite máximo da maioria dos dispositivos. Esse problema foi contornado com a utilização de um cache de um nível em *LRU*, onde cada linha tem um tamanho 1 e o tamanho máximo do cache é de 10.

4.2.3.3 Primeira Execução

Na primeira execução são necessárias algumas operações pra que o aplicativo fique pronto para o uso. Essas operações levam atualmente aproximadamente 2 segundos, variando dependendo do dispositivo. O layout é apresentado vazio com uma animação de caregamento sendo mostrada ao usuário.

Como já comentado a biblioteca MapsForge precisa que o mapa esteja no armazenamento interno do dispositivo. O primeiro passo feito é copiar o mapa contido dentro do pacote do aplicativo, essa cópia é enviada para a pasta de arquivos internos do usuário relacionada ao aplicativo.

Outro passo necessário é criarmos um banco de dados, contendo apenas as informações mais básicas de cada linha. Cada linha vem em um arquivo separado, comprimidas em um arquivo ZIP colocado dentro do pacote do aplicativo. Os dados básicos de uma linha são o nome da mesma e os "quadrados"que ajudam na busca de que linhas passam próximas a determinada localização. Nesse momento também são inicializados outros dados que ficarão no banco de dados: se a linha é favorita, quantas vezes o usuário a abriu e se na última vez que a viu ela era apresentada no sentido de ida ou volta.

4.2.3.4 Tela Inicial

Após as operações iniciais a tela inicial é apresentada para o usuário contendo o mapa da cidade de Florianópolis, isso é feito utilizando a biblioteca *MapsForge*. Para o aplicativo isso é um processo simples, apenas indicando pra *MapsForge* o caminho do mapa.

Como opção para o usuário temos a opção de busca, apresentada ao usuário como um dos botões da Action Bar em formato de lupa, icone bastante utilizado no Android com esse propósito.

4.2.3.5 Busca

A opção de busca tem um papel fundamental no aplicativo, pois é por ela que o usuário encontra a linha que deseja, não existe uma interface de listas onde se encontra determinada linha.

Foi utilizado um componente específico da *Action Bar* do *Android* para a busca, uma *SearchView*. Ao fazer uma busca pela *SearchView* o aplicativo pega as informações do banco de dados que se encaixam com a mesma, fazendo uma organização de listagem que segue a seguinte ordem de precedência: linhas favoritas, linhas mais acessadas, linhas com maior número de horários.

A lista com as linhas encontradas é mostrada abaixo da *SearchView*, por cima da exibição do mapa. A cada caractere entrado a lista é atualizada, não sendo necessária a realização efetiva da busca no banco de dados mas apenas em uma lista que já se encontra na memória.

4.2.3.6 Exibição do Trajeto

Quando uma linha é selecionada carregamos todas as informações dela na memória, em nosso cache *LRU*, após o carregamento quase instantâneo o trajeto é mostrado sobre o mapa para o usuário.

A biblioteca MapsForge possui suporte pra exibição de trajetos e imagens sobre o mapa, isso é feito de maneira muito simples, passando uma lista de pontos de localização. Pra adicionar imagens basta criar um "Marker"e adicionar ao mapa, no caso do aplicativo desenhamos um ícone verde indicando o ponto inicial de um trajeto e um ícone vermelho indicando o ponto final.

4.2.3.7 Posição Estimada dos Ônibus

Para poder mostrar ao usuário a posição estimada dos ônibus é necesário primeiramente saber que ônibus estão no meio do percurso. Esses ônibus são os que já partiram, ou seja, o horário de partida deles é inferior ao horário atual e que tenham partido em

no máximo X minutos atrás, sendo que X é o tempo que ele leva pra fazer seu percurso. Como o cálculo depende do horário atual é necessário que o dispositivo esteja com a hora correta.

Após isso é desenhado no mapa a posição de cada ônibus em percurso. Para chegar na posição que um ônibus se encontra é feito primeiramento o cálculo da relação de tempo que ele está na rua, dividindo esse tempo pelo tempo total de percurso. Depois essa relação é multiplicada pela distância total do percurso, chegando à distância que ele se encontra no momento. Para determinar que coordenada no mapa ele está é sendo somanda a distância entre os pontos de coordenada do percurso, ao alcançar a distância o ônibus é desenhado na coordenada anterior em que a soma foi alcançada.

4.2.3.8 Tabela de Horários

Quando uma linha está selecionada a *Action Bar* abresenta um botão para ir para a tabela de horários, quando o usuário seleciona esse botão o mapa é escondido e a tabela de horários mostrada.

A tabela de horários consiste em um ViewPager com 3 abas: semana, sábado e domingo, referentes aos diferentes horários que as linhas têm em diferentes dias da semana. O conteúdo de cada aba apresenta o sentido da linha em seu título, seguido de um GridView contendo cada horário daquela linha referente ao dia específico.

5 Resultados Obtidos

5.1 Base de Usuários

Disponível gratuitamente para usuários *Android* pelo Google Play desde 06/05/2013 BusMaps... (2013), o BusMaps já foi instalado, segundo o painel de desenvolvedores do Google Play, em mais de 4300 aparelhos, sendo que cerca de 53% dessas instalações continuam ativas. Não foram encontradas estatísticas globais públicas de confiança sobre taxa de retenção em aplicativos gratuitos para *Android*, de modo a verificar se a porcentagem atual do aplicativo indica algo positivo.

Alguns comentários de usuários:

> O melhor em seu segmento Alem de me ajudar com os horarios, me ajudou a conhecer melhor a cidade e me deu mais opcoes de transporte para locais que eu ja conhecia. Excelente APP tanto para nativos como turistas. Deveria receber incentivo da prefeitura. BusMaps... (2013)

> No geral, é excelente! O design é meio simples demais, mas a utilidade é sensacional! Ótimo trabalho de quem fez. BusMaps... (2013)

> Utililidade publica Este aplicativo deveria ter patrocinio ou ao menos divulgação da prefeitura. Talvez um banner junto aos terminais. BusMaps... (2013)

Pelas avaliações dos usuários, em uma nota que varia de 1 a 5, o aplicativo atualmente tem uma média de 4,6 BusMaps... (2013). Boa parte das avaliações negativas vêm com comentários relacionados à falta de cobertura das linhas intermunicipais, e mais recentemente algumas sobre falta de atualização das tabelas de horários.

5.2 Divulgação na Mídia

Foram publicadas matérias sobre o BusMaps em diversos meios de comunicação que contribuiram na popularização do aplicativo.

> Os passageiros de ônibus de Florianópolis ganharam uma nova opção para se informar sobre os trajetos e horários das empresas Transol, Canasvieiras, Estrela e Insular. O aplicativo para celulares com a plataforma Android, chamado de Bus Maps. O aplicativo para celulares com a plataforma Android, chamado de Bus Maps, pode ser baixado gratuitamente e faz uma previsão do momento em que o ônibus irá chegar até o ponto de ônibus desejado, além de mostrar qual o trajeto de cada uma das linhas disponíveis no local. Diário... (2013)

Figura 6 – Painel de Desenvolvedores do Google Play - Instalações Totais

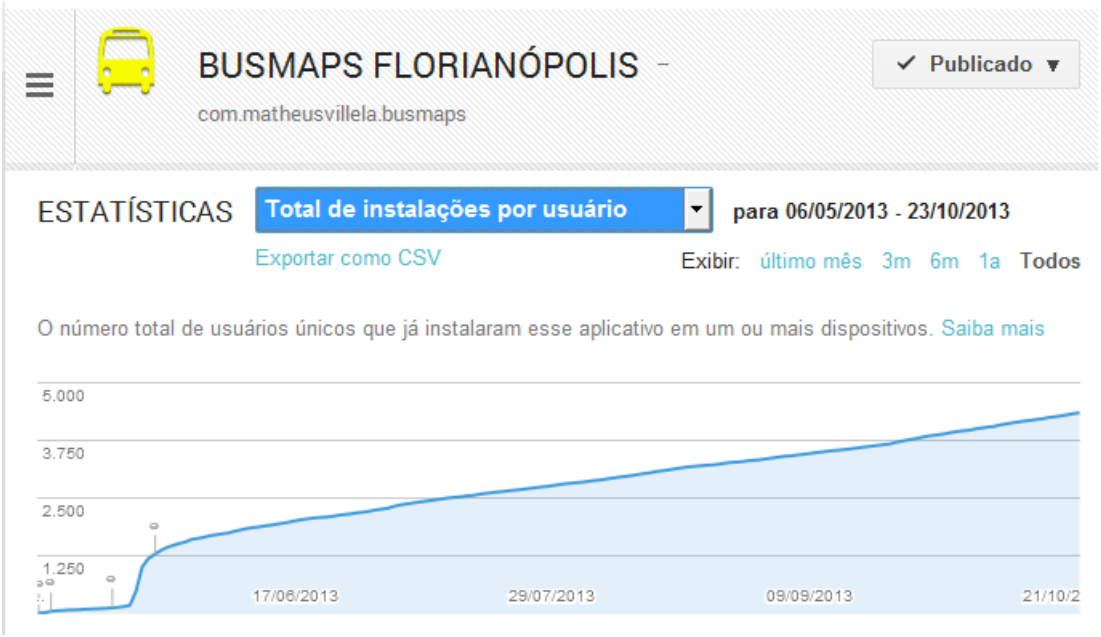


Imagem do painel de desenvolvedores do Google Play mostra o gráfico de instalações totais do *BusMaps* em um período de pouco mais de 5 meses. Houve um pico inicial de instalações logo após a inclusão de todas as linhas municipais da cidade e divulgação da mídia local, após isso o aumento de instalações continuou de maneira constante. Fonte: o autor

> Uma novidade que promete facilitar a vida de muita gente, talvez a sua vida também. Esperar um onibus sem saber se ele está no horário, ou em quanto tempo ele chega no ponto onde você está não é das coisas mais agradáveis não, principalmente se voce está com pressa não é isso mesmo? Bom, também não adianta procurar no próprio ponto informações desse tipo, porque nos nossos pontos quando existem a informação não existe, pelo menos aqui na capital. Foi por isso que um estudante de florianopolis decidiu arregaçar as mangas. Estudante... (2013)

> O "BUSMAPS" é um aplicativo gratuito que pode ser baixado no celular para quem usa a plataforma Android. Ele mostra os horários de saída dos ônibus das empresas de transporte do município de Florianópolis e apresenta um mapa com os trajetos operados para cada linha. Jornal... (2013)

> Com o mesmo objetivo, o estudante da sétima fase do curso de Ciências da Computação Matheus Villela desenvolveu o Bus Maps. O aluno, que entre 2010 e 2012 participou da criação da versão móvel do Windows Live Messenger para duas operadoras de celular, produziu o aplicativo como Trabalho de Conclusão de Curso. Além dos horários e mapas dos trajetos de toda as linhas municipais, o programa disponibiliza a localização aproximada e previsão de chegada do ônibus desejado. Até agora, o "Bus Maps" acumula cerca de dois mil downloads e a meta de seu criador é que atinja o público de dez mil usuários. Notícias... (2013)

Figura 7 – Painel de Desenvolvedores do Google Play - Instalações Atuais

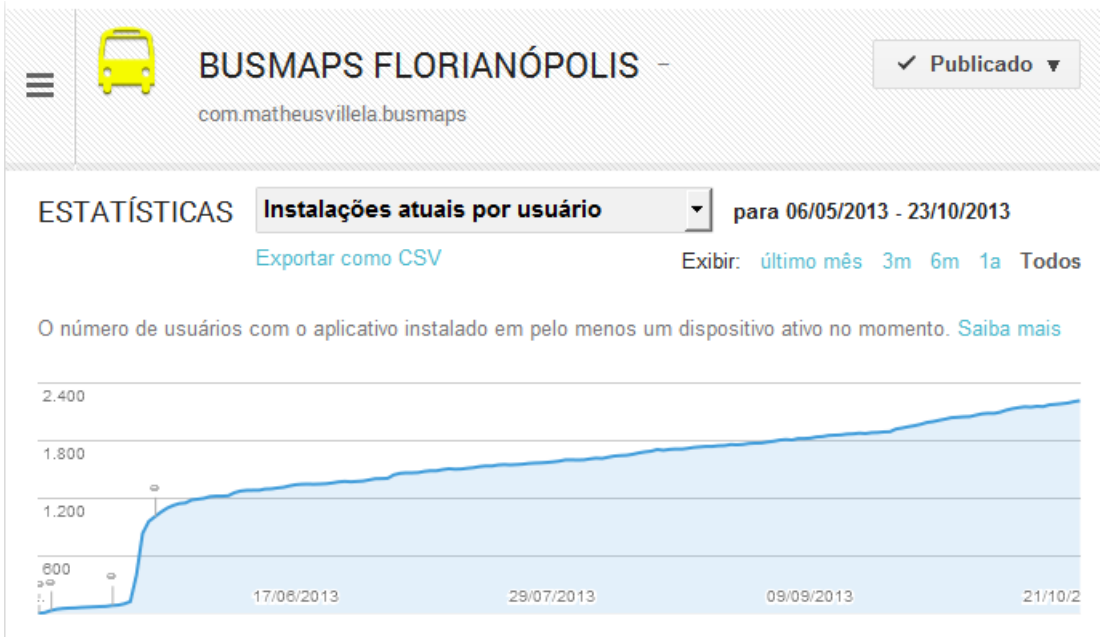


Imagem do painel de desenvolvedores do Google Play mostra o gráfico da quantidade de dispositivos que continuam com o BusMaps instalado. Fonte: o autor

6 Conclusões

Este trabalho trouxe aos habitantes da cidade de Florianópolis uma nova opção de informação sobre as linhas de ônibus da cidade. Se antes sem um navegador a única informação que os usuários tinham eram as tabelas de horários, agora é possível com o BusMaps consultar os trajetos das linhas e ter uma ideia de aproximadamente onde os ônibus se encontram.

Na época que o aplicativo foi lançado, a prefeitura ainda não tinha definido os detalhes técnicos do processo licitatório, mas uma grande mudança na parte de informação foi incluída à ele:

> A tecnologia também vai estar presente no novo modelo com a Central de Controle, que vai coordenar uma frota de ônibus equipados com câmeras de videomonitoramento e sistema de localização, o que vai permitir ao passageiro, por smartphone ou telefone celular, buscar informações precisas sobre itinerário. Todos os terminais serão equipados com painéis de LED. PMF (2013)

Enquanto a solução da prefeitura não estiver pronta o BusMaps continuará sendo útil, e como as linhas do núcleo metropolitano e intermunicipais não fazem parte do processo licitatório o BusMaps pode continuar como principal solução da região, desde que passe a cobrir as cidades vizinhas de Florianópolis e se adeque à nova realidade de disponibilização dos dados que a prefeitura implantar.

6.1 Sugestões para Trabalhos Futuros

6.1.1 Desenvolvimento para Outras Plataformas

Embora o *Android* venha crescendo bastante no mercado brasileiro, no Brasil os *Smartphones* ainda são minoria. O desenvolvimento de uma versão, mesmo que mais limitada, para dispositivos mais simples poderia ter um bom alcance na cidade devido à boa recepção que o BusMaps vem tendo.

O desenvolvimento de versões para outros *Smartphones* é outra possibilidade, nesse caso seria possível manter as mesmas funcionalidades. No entanto, considerando o perfil dos usuários de outros *Smartphones* e a fatia de mercado, o trabalho necessário para isso poderia ter resultados melhores sendo empregado na melhoria da versão atual, para Android.

6.1.2 Cobertura de Cidades do Núcleo Metropolitano

Atualmente o aplicativo só tem informações de linhas de ônibus que estão na página da *PMF*, que são apenas as linhas municipais da cidade de Florianópolis, não cobrindo linhas intermunicipais nem de outros municípios da região.

Para cobertura das linhas intermunicipais é necessário criar um analizador para boa parte das páginas das empresas de ônibus que atuam na região, visto que não existe um local que centraliza as informações nesse caso. Se for criado analizador para a página de uma empresa que atue, por exemplo, em Palhoça e São José e que também possua linhas intermunicipais com Florianópolis, o trabalho de se adicionar as linhas municipais da empresa para essas duas cidades se torna bem mais fácil.

Tabela 1 – Estimativas de população para cidades do núcleo metropolitano de Florianópolis com data de referência em 1º de julho de 2013

COD. MUNIC	NOME DO MUNICÍPIO	POPULAÇÃO ESTIMADA
05407	Florianópolis	453.285
16602	São José	224.779
11900	Palhoça	150.623
02305	Biguaçu	62.383
15703	Santo Amaro da Imperatriz	21.221
06009	Governador Celso Ramos	13.655
01208	Antônio Carlos	7.906
00606	Águas Mornas	5.926
17253	São Pedro de Alcântara	5.139

Fonte dos dados: IBGE (2013)

Vendo a população das cidades vizinhas, e considerando a influência delas em Florianópolis, existe uma boa oportunidade de crescimento do alcance de pessoas que se beneficiariam pelo aplicativo ao aumentar a quantidade de cidades cobertas. Somando as populações das cidades de São José, Palhoça e Biguaçu temos um número bem próximo da população de Florianópolis. É de se imaginar que a inclusão das linhas municipais dessas cidades dobraria o público do aplicativo, levando em conta que linhas intermunicipais seriam adicionadas o público seria maior ainda.

6.1.3 Cobertura de Outras Regiões

Existem muitas regiões onde o BusMaps teria um bom público, que não possuem um aplicativo semelhante ou o que existe é muito limitado. O maior problema está na disponibilidade dos trajetos em formato de coordenadas, e também da necessidade de cuidar de atualizações quando as tabelas são alteradas. Outras questão é que, como Florianópolis faz parte da realidade do autor, é muito mais fácil se manter motivado em manter o projeto atualizado para a cidade do que seria pra uma localidade sem nenhuma relação.

6.1.4 Suporte a GTFS

O Google não fornece para download o arquivo *GTFS* das cidades que disponibilizam os dados para eles, e em muitos casos as cidades não fazem questão alguma de deixar esses dados públicos.

Apesar disso, cerca de 700 agências de trânsito no mundo fornecem publicamente seus dados no formato, usando para isso o projeto *GTFS Data Exchange*, uma página que fornece uma maneira eficiente de agências publicarem os dados e dos desenvolvedores de aplicativos que os usam serem notificados quando ocorrem mudanças. GTFS... (2013). Embora exista um projeto como esse, nenhuma das 9 cidades brasileiras que publicam dados em *GTFS* para o Google o fazem no *GTFS Data Exchange*. Logo se fosse dado suporte a *GTFS*, para cobrir alguma das cidades brasileiras, seria necessário entrar em contato diretamente com uma dessas prefeituras e conseguir algum acordo de acesso aos dados.

A grande vantagem estaria em poder transformar o BusMaps em um aplicativo generalista, com um potencial de usuários bem maior, mas nesse caso a concorrência de aplicativos como o *Moovit* dificultaria sua adoção.

Referências

ACTION Bar | Android Developers. [S.l.], 2013. Disponível em: <http://developer-.android.com/guide/topics/ui/actionbar.html>. Acesso em: 30.10.2013. Citado na página 16.

ACTIVITIES | Android Developers. [S.l.], 2013. Disponível em: <http://developer-.android.com/guide/components/activities.html>. Acesso em: 01.11.2013. Citado na página 16.

ANATEL. *Brasil ultrapassa um celular por habitante*. [S.l.], 2010. Disponível em: <http://www.anatel.gov.br/Portal/exibirPortalNoticias-.do?acao=carregaNoticia\codigo=21613>. Acesso em: 22.10.2013. Citado na página 11.

ANATEL. *Brasil alcança 268,44 milhões de acessos móveis em agosto*. [S.l.], 2013. Disponível em: <http://www.anatel.gov.br/Portal/exibirPortalNoticias-.do?acao=carregaNoticia\codigo=30969>. Acesso em: 22.10.2013. Citado na página 11.

ANDROID. *Android*. [S.l.], 2013. Disponível em: <http://www.android.com/>. Acesso em: 28.10.2013. Citado na página 14.

ANDROID Developers. [S.l.], 2013. Disponível em: <http://source.android.com/>. Acesso em: 28.10.2013. Citado na página 14.

ANDROID Overview | Open Handset Alliance. [S.l.], 2013. Disponível em: <http:-//www.openhandsetalliance.com/android_overview.html>. Acesso em: 28.10.2013. Citado na página 14.

BORSNTEIN, D. *Dalvik VM Internals*. [S.l.], 2008. Disponível em: <https://sites.google-.com/site/io/dalvik-vm-internals/2008-05-29-Presentation-Of-Dalvik-VM-Internals-.pdf?attredirects=0>. Acesso em: 28.10.2013. Citado na página 15.

BUSMAPS Florianópolis - Aplicativos para Android no Google Play. [S.l.], 2013. Disponível em: <https://play.google.com/store/apps/details?id=com.matheusvillela-.busmaps>. Acesso em: 28.10.2013. Citado na página 34.

CETIC.BR. *J2A - PROPORÇÃO DE INDIVÍDUOS, POR QUANTIDADE DE LINHAS DE TELEFONE CELULAR*. [S.l.], 2013. Disponível em: <http://www-.cetic.br/usuarios/tic/2012/J2a.html>. Acesso em: 23.10.2013. Citado na página 11.

CNET. *Google's Android parts ways with Java industry group*. [S.l.], 2007. Disponível em: <http://news.cnet.com/8301-13580_3-9815495-39.html>. Acesso em: 28.10.2013. Citado na página 14.

CNET. *Linux and Android, together at last*. [S.l.], 2012. Disponível em: <http://news-.cnet.com/8301-30685_3-57399951-264/linux-and-android-together-at-last/>. Acesso em: 28.10.2013. Citado na página 14.

COMPUTERWORLD. *Android/Linux kernel fight continues.* [S.l.], 2010. Disponível em: <http://blogs.computerworld.com/16900/android_linux_kernel_fight_continues>. Acesso em: 28.10.2013. Citado na página 14.

DIáRIO Catarinense - Aplicativo mostra trajetos e horários em que ônibus passam nos pontos da Capital. [S.l.], 2013. Disponível em: <http://diariocatarinense.clicrbs.com.br-/sc/noticia/2009/05/florianopolis-tem-pior-mobilidade-urbana-do-brasil-2523317.html>. Acesso em: 28.10.2013. Citado na página 34.

DIáRIO Catarinense - Florianópolis tem pior mobilidade urbana do Brasil. [S.l.], 2009. Disponível em: <http://diariocatarinense.clicrbs.com.br/sc/noticia/2009-/05/florianopolis-tem-pior-mobilidade-urbana-do-brasil-2523317.html>. Acesso em: 23.10.2013. Citado na página 12.

ESTATíSTICAS de Celulares por Código de Área (DDD). [S.l.], 2013. Disponível em: <http://www.teleco.com.br/ncelddd.asp>. Acesso em: 23.10.2013. Citado na página 11.

ESTUDANTE de Florianópolis desenvolve aplicativo que auxilia no uso do transporte coletivo - G1 Santa Catarina - Jornal do Almoço - Catálogo de Vídeos. [S.l.], 2013. Disponível em: <http://g1.globo.com/sc/santa-catarina/jornal-do-almoco-/videos/t/edicoes/v/estudante-de-florianopolis-desenvolve-aplicativo-que-auxilia-no-uso-do-transporte-coletivo/2591221/>. Acesso em: 28.10.2013. Citado na página 35.

FERRIS, B. *OneBusAway: Improving the Usability of Public Transit.* Tese (Doutorado) — Department of Computer Science and Engineering, University of Washington, 2011. Disponível em: <http://onebusaway.gatech.edu/xwiki/bin/download/Main/Research-/BrianFerris-Dissertation.pdf>. Acesso em: 01.11.2013. Citado 2 vezes nas páginas 13 e 19.

GARTNER. *Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time.* [S.l.], 2013. Disponível em: <http://www.gartner.com/newsroom/id/2573415>. Acesso em: 23.10.2013. Citado na página 12.

GOOGLE. *Programa de parceiros do Google Transit.* [S.l.], 2013. Disponível em: <http://maps.google.com/help/maps/mapcontent/transit/participate.html>. Acesso em: 23.10.2013. Citado na página 18.

GOOGLE. *What is GTFS?* [S.l.], 2013. Disponível em: <https://developers.google.com-/transit/gtfs/>. Acesso em: 23.10.2013. Citado na página 18.

GTFS Data Exchange. [S.l.], 2013. Disponível em: <http://www.gtfs-data-exchange-.com/>. Acesso em: 31.10.2013. Citado na página 39.

IBGE. *ESTIMATIVAS DA POPULAÇÃO RESIDENTE NOS MUNICÍPIOS BRASILEIROS COM DATA DE REFERÊN-CIA EM 1º DE JULHO DE 2013.* [S.l.], 2013. Disponível em: <ftp://ftp.ibge.gov.br/Estimativas$_d$e$_P$opulacao/Estimativas$_2$013/estimativa$_2$013$_d$ou.pdf>. Acesso em: 31.10.2013. Citado na página 38.

IDC. *More Smartphones Were Shipped in Q1 2013 Than Feature Phones, An Industry First According to IDC*. [S.l.], 2013. Disponível em: <http://www.idc.com-/getdoc.jsp?containerId=prUS24085413>. Acesso em: 23.10.2013. Citado na página 11.

JORNAL da BAND - Aplicativo para smartphone auxilia usuários de coletivos informando horário e trajeto dos ônibus. [S.l.], 2013. Disponível em: <http://bandsc.com.br/canais/noticias-/aplicativo_para_smartphone_auxilia_usuarios_de_coletivos_informando_horario_e_trajeto_dos_onibus_-.html>. Acesso em: 28.10.2013. Citado na página 35.

LRUCACHE | Android Developers. [S.l.], 2013. Disponível em: <http://developer-.android.com/reference/android/util/LruCache.html>. Acesso em: 29.10.2013. Citado na página 15.

MAPSFORGE - free mapping and navigation tools. [S.l.], 2013. Disponível em: <http://code.google.com/p/mapsforge/>. Acesso em: 01.11.2013. Citado na página 16.

MEDEIROS, V. *URBIS BRASILIAE OU SOBRE CIDADES DO BRASIL: inserindo assentamentos urbanos do país em investigações configuracionais comparativas*. Tese (Doutorado) — Universidade de Brasília, 2006. Citado na página 12.

MOVELBUS. [S.l.], 2013. Disponível em: <http://www.movelbus.com.br/>. Acesso em: 31.10.2013. Citado na página 19.

NOTíCIAS da UFSC - Estudantes da UFSC criam aplicativos de celulares para facilitar o dia a dia. [S.l.], 2013. Disponível em: <http://noticias.ufsc.br/2013/07/estudantes-da-ufsc-criam-aplicativos-de-celulares-para-facilitar-o-dia-a-dia/>. Acesso em: 31.10.2013. Citado na página 35.

OHLOH. *The Android Open Source Project on Ohloh : Languages Page*. [S.l.], 2013. Disponível em: <http://www.ohloh.net/p/android/analyses/latest/languages_summary>. Acesso em: 28.10.2013. Citado na página 14.

PMF. *Conheça as diretrizes do edital do transporte coletivo*. [S.l.], 2013. Disponível em: <http://www.pmf.sc.gov.br/noticias/index.php?pagina=notpagina\noti=9940>. Acesso em: 01.11.2013. Citado na página 37.

TAGSOUP - Just Keep On Truckin. [S.l.], 2013. Disponível em: <http://ccil.org-/~cowan/XML/tagsoup/>. Acesso em: 29.10.2013. Citado na página 15.

WIKIPEDIA. *Java (programming language)*. 2013. Disponível em: <http://en.wikipedia-.org/wiki/Java_(programming_language)>. Acesso em: 28.10.2013. Citado na página 15.

ZHOU, Y. *Definitions of various cache algorithms*. [S.l.], 2001. Disponível em: <https:-//www.usenix.org/legacy/events/usenix01/full_papers/zhou/zhou_html/node3.html>. Acesso em: 29.10.2013. Citado na página 15.