

Projeto de um Aplicativo Android de Monitoramento de Veículos do Transporte Público

Geovane Fedrecheski¹
Richard Aderbal Gonçalves (Orientador)¹
Josiel Neumann Kuk (Co-orientador)¹

¹Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Rua Camargo Varela de Sá, 3 – CEP 85040-080
Guarapuava – PR – Brasil – Telefone: (42) 3629-8100

geonnave@gmail.com

Abstract. *The raise of mobile computing ensures its importance to the world nowadays. In parallel, public transportation has appearing as an important role to a wide portion of the population. In this work we design an Android based solution applied to public transportation vehicles' real-time tracking. In order to provide a well defined design, we use Software Engeneering and Human-Computer Interaction techniques (such as Requirements Specification and Interface Design). There was also developed a prototype of the main screens of the system using Android.*

Keywords. *Android, Urban Public Transportation, Real-Time Vehicle Tracking, Software Engeneering, Human-Computer Interacion.*

Resumo. *O advento da computação móvel mostra o quanto ela é importante para o nosso mundo atual. Em paralelo, o transporte público urbano tem ocupado um importantíssimo papel para uma larga escala da população. Neste trabalho apresenta-se o projeto uma solução baseada em Android aplicada ao rastreamento em tempo real de veículos do transporte público. Para fornecer um projeto bem definido, foram utilizadas técnicas de Engenharia de Software e Interação Humano-Computador (como por exemplo Especificação de Requisitos e Design de Interface). Também foi desenvolvido um protótipo das principais telas do sistema na plataforma Android.*

Palavras-chave. *Android, Transporte Público Urbano, Monitoramento de Veículos em Tempo Real, Engenharia de Software, Interação Humano-Computador.*

1. Introdução

Em muitas áreas do conhecimento, é evidente a busca do ser humano por ferramentas que possam melhorar a sua qualidade de vida. Tal afirmação se aplica fortemente à Computação: máquinas automatizadas foram desenvolvidas para, a princípio, ajudar em problemas matemáticos complexos, repetitivos e trabalhosos.

O objetivo deste trabalho está diretamente relacionado com esse princípio. Aqui é proposta a utilização da Computação Móvel para resolver um problema de dificuldade de

acesso a informações do transporte público urbano, por parte dos usuários. Assim, nesta seção apresentam-se os principais fatores e as motivações que nortearam a elaboração deste estudo, além de definir o seu escopo.

1.1. A Computação Móvel

Desde a década de 1970 até os dias atuais, a evolução e a abrangência dos computadores pessoais no mundo inteiro aconteceu de uma maneira muito intensa. Devido à essa evolução, hoje tem-se uma interessantíssima área derivada dos computadores pessoais comuns: a Computação Móvel. Ela amplia o conceito tradicional de Computação Distribuída através da utilização da comunicação sem fio, eliminando assim a limitação da mobilidade [Figueiredo e Nakamura 2003].

Geralmente, o surgimento de novas áreas de pesquisa costumam trazer à tona uma ampla gama de novas possibilidades, bem como novos impactos na sociedade. Com a Computação Móvel não é diferente, pois já é notável a presença e o impacto dos dispositivos móveis (como *tablets*, *smartphones* e PDA's) na vida contemporânea. Temos como grandes exemplos disso os *smartphones*, que podem ser definidos como “celulares que oferecem recursos avançados, frequentemente com funcionalidades iguais às de um computador ou a capacidade de fazer *download* de aplicativos” [Google 2012].

Os *smartphones* estão entre os principais responsáveis pelo advento da Computação Móvel e hoje em dia já protagonizam um grande impacto social. Segundo dados de uma pesquisa realizada em 27 países e liderada pelo Google, 14% dos brasileiros (cerca de 27 milhões de pessoas) possuem um. Destes, 73% não saem de casa sem seus aparelhos; 42% os utilizam para acessar a Internet todos os dias; 60% fazem uso deles enquanto utilizam algum meio de transporte público [Google 2012].

Os sistemas operacionais mais completos e que dominam o mercado atualmente são o Android da Google, e o iOS da Apple, sendo que o primeiro detém a maior fatia [Google 2012]. Em suas respectivas lojas de aplicativos *online*, os usuários podem escolher e baixar centenas de apps – como são chamados em seu contexto – com as mais variadas funcionalidades [Lecheta 2010]. Por exemplo, em São Paulo, um aplicativo para Android e iOS permite pedir táxi pelo celular¹. Outros aplicativos interessantes são aqueles que podem ajudar o usuário em casos de furtos, informando secreta e automaticamente a localização do dispositivo.

1.2. O Problema do Transporte Público

Assim, tendo uma visão do quão inserida no cotidiano das pessoas está a Computação Móvel, nada mais natural do que vê-la e utilizá-la cada vez mais como um instrumento de melhoria na qualidade de vida dos seus usuários. O problema que será abordado consiste na dificuldade – por parte dos usuários – do acesso à informação concernente ao transporte público nas grandes cidades.

Esse tipo de transporte representa um importantíssimo papel na vida de milhares de pessoas. São trabalhadores, estudantes e aposentados, entre outros cidadãos, que dependem desse tipo de transporte para desempenhar suas respectivas atividades. Sua massiva utilização pode acarretar em vantagens como a redução de custos, poluição – tanto sonora quanto atmosférica – e a otimização do uso das vias urbanas [Goldsmith *et al.* 2006].

¹<http://taxija.com/>

No entanto, muitas vezes esse tipo de transporte só é utilizado devido à inexistência de outras alternativas (como por exemplo um veículo próprio). A aversão pelo uso do transporte público é muitas vezes consequência da má qualidade do mesmo.

A insatisfação dos usuários tomou forma num estudo realizado pelo IPEA (Instituto de Pesquisa Econômica Aplicada), no ano de 2011. Segundo tal estudo, constatou-se que nas cidades com mais de 100 mil habitantes, 61% da população não consegue ser atendida pelo transporte público sempre que precisam [Ipea 2012]. Além disso, esse mesmo estudo aponta que, ainda nas cidades com mais de 100 mil habitantes, 37% da população avalia a quantidade de informação sobre o transporte público como “muito ruim/ruim” (contra 34% dos que a julgam “muito boa/boa”).

1.3. Proposta de Solução Utilizando Computação Móvel

Devido à reprovação dos usuários, constata-se que a falta de informação é em parte responsável pela má qualidade do transporte público. Assim seria interessante encontrar uma forma de diversificar e facilitar o acesso a esse tipo de informação. Por exemplo, horários de ônibus e itinerários precisam ser amplamente acessíveis. A partir disso, apresenta-se uma solução baseada na Computação Móvel, a qual é brevemente explicada abaixo:

- Um módulo para efetuar o rastreamento dos veículos do transporte público através do GPS (*Global Positioning System*) de aparelhos celulares, o qual utilizaria suas tecnologias de transmissão de dados para enviar os dados a um servidor;
- Um servidor, cujo objetivo é receber esses dados, processá-los e responder às requisições do aplicativo *client-side*;
- Um aplicativo para *smartphones*, destinado ao usuário final, cuja principal funcionalidade será a de exibir dados recebidos do servidor em tempo real a respeito do posicionamento dos veículos.

Apesar da descrição apresentada do escopo completo de uma possível solução, é importante citar que não é objetivo deste trabalho estudar e desenvolver toda a infraestrutura necessária para implementá-la na prática. Este trabalho visa um planejamento para um desenvolvimento futuro da solução supracitada.

Além disso, aborda-se apenas o módulo do usuário final, denominado “Cadê meu ônibus?”, virtualmente ignorando-se os detalhes técnicos de como as informações do rastreamento serão obtidas e processadas no servidor. Essa abordagem realiza-se na forma de um projeto da aplicação *client-side*, nas áreas de Engenharia de Software (ES) e Interação Humano-Computador (IHC).

O projeto de ES é focado no levantamento de requisitos, com objetivo de produzir um documento de Especificação de Requisitos de Software (ERSw). Já o projeto de IHC aborda o *Design* de Interface, partindo da elaboração dos Mockups das telas e chegando até a prototipagem de algumas delas.

2. Trabalhos e Sistemas Correlatos

As tecnologias de Computação móvel e geoposicionamento (detecção de posição geográfica) vem sendo utilizadas em conjuntos há alguns anos, inclusive essa interessante combinação já tem sido aplicada ao problema do transporte público urbano. Nesta seção serão citados alguns trabalhos já existentes nessa área.

2.1. *Bus Tracker*

O primeiro estudo que abordado é a aplicação *Bus Tracker* [Siqueira 2012]. Este trabalho compreende o projeto e o desenvolvimento de um protótipo de um sistema que permite aos usuários de transporte público acompanhar a posição e o deslocamento dos veículos em tempo real. Esta aplicação é dividida em três módulos principais: cliente, servidor e base de dados.

A aplicação cliente foi desenvolvida com base na plataforma iOS, da Apple². Ela apresenta as seguinte funcionalidades: localizar determinada linha de ônibus através de um sistema de busca e permitir que ele visualize, em um mapa, a posição dos veículos desta linha em tempo real; configuração de eventos de aproximação de ônibus de um determinado local.

O servidor deve processar os dados de localização dos ônibus em tempo real e atender às requisições da aplicação cliente. O processamento consiste em acessar as informações na base de dados e processá-las de acordo com o que foi solicitado na requisição.

A base de dados tem por finalidade armazenar todas as informações de rastreamento dos veículos em tempo real. No entanto, ela utiliza informações fictícias, visto que Siqueira [2012] não define uma infraestrutura de rastreamento para os veículos.

2.2. *OneBusAway*

Ferris [2011] define um estudo aprofundado sobre os benefícios do transporte público, bem como acerca dos problemas que ele pode acabar ocasionando. Além disso, apresenta uma proposta como solução baseada em sistemas de trânsito em tempo real focados no transporte público.

O sistema desenvolvido por Ferris chama-se *OneBusAway* e opera nas plataformas: *web*, *interactive voice response phone* (resposta interativa por voz), SMS (*Short Message Service*) e plataformas móveis (Android, iOS e WindowsPhone). Na época em que o estudo foi realizado, o sistema era utilizado semanalmente por 50.000 usuários de ônibus e metrô da região Puget Sound, no estado de Washington.

O projeto é completamente *open-source*, e possui uma comunidade ativa de desenvolvedores contribuindo para a melhoria do *OneBusAway*, por exemplo expandindo-o para outras plataformas.

2.3. *Busão Curitibano*

No ano passado, a URBS (Urbanização de Curitiba S/A) inaugurou o seu Centro de Controle Operacional, cujo objetivo é acompanhar o que acontece nos ônibus e nas ruas, formando um núcleo de comando *online* com comunicação direta com motoristas de ônibus e do trânsito em geral [URBS 2012].

Já no presente ano, entra no mercado o aplicativo Busão Curitibano³. Ele utiliza-se dos dados sobre os ônibus coletados pela URBS para fornecer aos usuários um sistema de monitoramento dos ônibus em tempo real, bem como informações sobre linhas, pontos, horários e linhas turísticas.

²iOS é um Sistema Operacional para dispositivos móveis da Apple (uma forte empresa de tecnologia dos EUA) – <http://www.apple.com/ios/>

³<http://busaocuritibano.com.br>

Foram elaboradas versões tanto para a plataforma Android quanto para o iOS. Atualmente, ele está em pleno funcionamento e já foi baixado em mais de 18 mil *smartphones*.

3. Android

Na computação móvel, a tecnologia Android está entre as mais avançadas em termos de mercado, funcionalidade e interface com o usuário [Google 2012]. Esta seção irá descrever o surgimento da plataforma Android, bem como apresentar uma breve descrição técnica sobre a mesma.

3.1. História e Filosofia

O Android consiste em uma plataforma de desenvolvimento de aplicações para dispositivos móveis baseada em um sistema operacional Linux, com diversas aplicações já instaladas, além de oferecer também um ambiente de desenvolvimento poderoso e flexível [Lecheta 2010].

Ele foi lançado em 2007 pela *Open Handset Alliance*⁴ (OHA), a qual era um grupo recém formado por fortes empresas do mercado de telefonia e liderado pelo Google. Na época a OHA era constituída por cerca de 30 integrantes, mas hoje já conta com cerca de 90 empresas. O objetivo deste grupo era criar uma plataforma única e aberta para celulares e que pudesse se adequar aos interesses dos usuários comuns, dos desenvolvedores e dos ambientes corporativos.

Devido ao fato de que esta plataforma é baseada em princípios *open source*, ela vem apresentando excelentes resultados e ótimas perspectivas de aperfeiçoamento. A sua filosofia aberta permite que as empresas fabricantes de celulares customizem o Android sob a perspectiva que lhes convir, bem como facilita para que desenvolvedores do mundo todo contribuam corrigindo *bugs*, adicionando novas funcionalidades ou desenvolvendo novas aplicações [Lecheta 2010].

3.2. A Plataforma Android

A plataforma Android é baseada no *kernel* 2.6 do Linux, sendo que suas aplicações são escritas na linguagem Java e executadas sobre a máquina virtual Dalvik⁵, a qual é otimizada para dispositivos móveis.

O *kernel* Linux utilizado pelo Android é responsável pelo gerenciamento de memória, processos, redes, drivers e segurança do sistema de arquivos. Isso garante confiabilidade e estabilidade ao sistema [Love 2004].

Como supracitado, as aplicações são desenvolvidas em Java e executadas na máquina virtual Dalvik. Durante o desenvolvimento, ao compilar o código java, os *bytecodes* gerados são compactados em arquivos com a extensão *.dex* (*Dalvik Executable*) e posteriormente compactados em um único arquivo *.apk* (*Android Package File*), o qual representa a aplicação final e pronta para ser instalada [Lecheta 2010].

⁴<http://www.openhandsetalliance.com/>

⁵Responsável por executar os *bytecodes* gerados pelo compilador.

3.3. Mapas e GPS

Uma das funcionalidades que mais chamam a atenção na plataforma Android é a relativamente fácil integração de uma aplicação com o Google Maps [Lecheta 2010]. Para tanto, o Google oferece um pacote de serviços para o Android, no qual está incluída a Google Maps API (*Application Programming Interface*).

Atualmente esta API está na versão 2, e fornece uma série de recursos nativos e atualizados para o desenvolvedor (como por exemplo *zoom* e navegação pelo mapa). Ela também permite ao desenvolvedor implementar algumas customizações no mapa, como por exemplo adicionar objetos e marcações ao mapa, bem como oferece a possibilidade de interações com o usuário a partir dessas marcações [Google 2013].

4. Os Processos de ES e IHC

O objetivo desta seção é introduzir alguns conceitos sobre as áreas do conhecimento conhecidas como Engenharia de Software e Interação Humano-Computador, a fim de oferecer um embasamento teórico para os resultados expostos na Seção 5.

4.1. Engenharia de Software

A Engenharia de Software é uma área do conhecimento que coexiste com a Ciência da Computação. Nela encontramos teorias, métodos e ferramentas para projeto e desenvolvimento de *software* – desde os mais simples até os mais complexos e abstratos. Na maioria dos casos, os sistemas são relativamente grandes e complexos [Sommerville 1995].

Além disso, o fato dos projetos de software serem abstratos e virtualmente não terem uma forma física é o principal fator que difere a Engenharia de Software das outras engenharias. Pode-se dizer que a ES não possui restrições físicas ou de processos de manufatura. Disso podem advir vantagens (praticamente qualquer sistema computável *pode* ser projetado), bem como fatores complicantes ou indesejáveis (complexidade muito alta, inviabilidade econômica). Para evitar esses fatores, outros tipos de restrições devem ser consideradas [Sommerville 1995].

Assim, o foco da ES é o desenvolvimento de sistemas de *software* de alta qualidade dentro de custos adequados. Para se alcançar esse desenvolvimento deve-se elaborar um projeto formal de *software*. Filho [2006] e Sommerville [2007] apresentam visões amplamente detalhadas e que se complementam nas suas abordagens sobre os processos e projetos de Engenharia de Software.

Neste trabalho, a partir de um estudo sobre as referências citadas acima, foi desenvolvido um Documento de Especificação de Requisitos de Software. O intuito deste documento é compilar todos os artefatos⁶ produzidos durante a execução dos processos de ES. Além disso, também entram nesse documento definições básicas como o escopo e o propósito do *software* desenvolvido, além de Diagramas de Sequência e de Casos de Uso.

4.2. Interação Humano-Computador

Interação Humano-Computador é uma disciplina interessada no projeto, implementação e avaliação de sistemas computacionais interativos para uso humano, juntamente com os

⁶Um artefato (produto do trabalho) é um modelo, documento ou código produzido por uma atividade.

fenômenos relacionados a esse uso [Barbosa e da Silva 2010]. Não obstante, os objetos de estudo de IHC podem ser agrupados em cinco tópicos inter-relacionados:

- A natureza da Interação Humano-Computador;
- O uso de sistemas interativos situado em contexto;
- Características humanas;
- Arquitetura de sistemas computacionais e da interface com usuários; e
- Processos de desenvolvimento preocupados com uso.

A partir desses objetos de estudo surgiram processos de IHC que são de absoluta importância em sistemas interativos. Devido à isso, é importante que haja coerência entre as atividades de ES e IHC, aumentando assim as chances de se alcançar um software de qualidade [Barbosa e da Silva 2010]. Uma das principais abordagens visando esse objetivo é a definição de processos de IHC que devem ser incorporados aos processos propostos pela ES.

Gulliksen *et al.* [2005] identificaram alguns princípios-chave que devem ser considerados a fim de se atender aos requisitos adequados de usabilidade, dentre os quais podemos citar:

- Foco no usuário: os objetivos, anseios e necessidades do usuário devem ser levados em conta;
- Desenvolvimento evolucionário: o desenvolvimento deve ser iterativo e incremental;
- Representações de *design* simples: o *design* deve ser representado de forma que seja facilmente entendido pelo usuário;
- Prototipação: protótipos devem ser usados desde o início para visualizar e avaliar ideias e soluções de design;
- Design holístico: o sistema deve ser sempre projetado a partir de uma visão paralela sobre todos os seus componentes;

O foco principal de IHC utilizado neste trabalho reside na interface gráfica para o usuário. O *design* das interfaces de um sistema geralmente vem precedido de esboços ou representações de baixa fidelidade⁷. Essas representações são também denominadas Mockups, e podem ser construídas tanto manualmente quanto com o auxílio de ferramentas computacionais [Barbosa e da Silva 2010].

4.3. Documento de Especificação de Requisitos de Software

Requisitos de Software definem o mapeamento das propriedades que um *software* deve ter para atender um determinado problema. Na Engenharia de Software, os requisitos normalmente são elicitados e desenvolvidos através de um processo de Análise de Requisitos, que os reúne de uma maneira organizada e detalhada, especificando desde o escopo do *software* até os modelos de dados, funções e fluxos de controle [Paula Filho 2001].

A Análise de Requisitos segue um fluxo de requisitos, o qual começa com a definição de um problema e culmina em uma Especificação de Requisitos de Software. Esta última definição consiste em um documento de Especificação de Requisitos de Software

⁷Uma representação é dita de baixa fidelidade quando se trata de um rascunho ou esboço da interface, sem muita preocupação com detalhes ou aspectos gráficos.

(ERSw), o qual é responsável por armazenar e centralizar de uma maneira organizada todos os artefatos produzidos pelo processo de análise.

Os artefatos produzidos pelo processo de análise e elaboração de requisitos deste trabalho foram armazenados em um documento de ERSw. A estrutura utilizada foi adotada a partir dos modelos descritos em Filho [2006] e Sommerville [2007], com algumas adaptações retiradas do modelo do RUP (*Rational Unified Process*). Nas seções posteriores, apresenta-se alguns dos principais artefatos presentes neste documento, como parte dos resultados obtidos com este trabalho.

5. Projeto

O projeto desenvolvido neste trabalho aborda os processos de ES e IHC de uma forma prática. Ele contém alguns artefatos importantes extraídos do documento de Especificação de Requisitos de *software*, bem como alguns *screenshots* e discussões sobre os principais Mockups e protótipos de interface do sistema.

5.1. Principais Artefatos do Documento de ERSw

Nesta seção serão listados alguns dos principais artefatos contidos no documento de ERSw. Como o referido documento, na íntegra, possui um total de páginas que não se adequa ao tamanho deste artigo, disponibilizamos uma versão completa do ERSw no Material Complementar A [Fedrecheski 2013a].

5.1.1. Escopo

O objetivo da seção Escopo, no ERSw, é descrever o produto de *software* a partir do seu nome, explicando as características do mesmo. Além disso, é interessante também que se descreva os benefícios que se espera com o produto e o valor destes para o cliente [Paula Filho 2001]. Para um melhor entendimento, tem-se a seguir o escopo do projeto desenvolvido.

O *software* Cadê meu ônibus? será um aplicativo para a plataforma móvel Android. Seu objetivo principal será exibir informações ao usuário sobre o geoposicionamento de veículos do transporte público em tempo real. Essas informações poderão ser visualizadas em um mapa, após a seleção de uma linha. Além da visualização, também será possível ao *software* emitir alertas ao usuário conforme a ocorrência de eventos pré-configurados. Para tanto, será necessária a conexão com a Internet.

Ele irá também apresentar algumas funcionalidades que estarão disponíveis tanto em modo *online*, quanto em modo *offline* – com restrições nesta última. As principais funções disponíveis também no modo *offline* são: Consulta, listagem e exibição de linhas, pontos e horários.

Este *software* faz parte de um projeto mais amplo, do qual também fazem parte: uma infraestrutura de rastreamento dos veículos em tempo real e um servidor cujo objetivo é processar os dados de tempo real e responder às requisições da aplicação Cadê meu ônibus?.

Ao facilitar o acesso às informações tanto em tempo real quanto estáticas aos usuários do transporte público urbano, o Cadê meu ônibus? visa agregar valor à qualidade

deste serviço – cujos usuários estão frequentemente insatisfeitos [Ipea 2012] –, assim incentivando a sua utilização.

5.1.2. Requisitos Funcionais

Os requisitos funcionais descrevem, de uma maneira relativamente técnica, as funções que o produto de *software* possui [Paula Filho 2001]. Eles podem ser descritos tanto na forma de uma listagem detalhada com as funções do produto, quanto na forma de Casos de Uso. A primeira, é mais indicada ao desenvolvedor, visto que descreve as funções de uma maneira relativamente técnica. Já a segunda forma é mais indicada a ser exibida para o cliente, pois explica as funções do sistema como se fosse "contando uma história" sobre como o sistema será utilizado.

Devido ao diferente contexto em que cada estilo de descrição dos requisitos funcionais tem maior aplicabilidade, decidiu-se neste trabalho utilizar as duas formas. Nesta seção ilustram-se as principais funções do produto de uma forma mais técnica, deixando a abordagem de casos de uso para a seção subsequente.

Exibição em Mapa (RF01)

1. Quando estiver conectado à Internet, o sistema deve ser capaz de exibir a posição dos ônibus de uma (ou mais) linha em tempo real;
2. Ao exibir a posição do ônibus em tempo real, o sistema deve indicar a sua direção de deslocamento;
3. Ao exibir a posição do ônibus em tempo real, o sistema deve permitir ao usuário selecionar o referido ônibus no mapa a fim de informá-lo sobre a precisão, em segundos, dessa informação;
4. O sistema deve ser capaz de exibir um ponto de ônibus no mapa, bem como oferecer informações detalhadas a partir do mesmo;
5. O sistema deve ser capaz de exibir o conjunto de pontos de uma linha no mapa;
6. O sistema deve ser capaz de exibir um conjunto de linhas em um mapa, bem como destacar seus pontos de intersecção. À essa visualização denominamos “rota”;
7. Na visualização das rotas, o sistema deve ser capaz de indicar um ponto de subida (no ônibus) e um ponto de descida;

Listagem de Linhas (RF02)

1. O sistema deve exibir uma listagem com todas as linhas cadastradas;
2. O sistema deve permitir a entrada de dados do usuário a fim de realizar um filtro por nomes das linhas;
3. O sistema deve permitir que o usuário acesse informações detalhadas sobre uma determinada linha através de uma caixa de diálogo ou visualizando-a diretamente no mapa (RF01);

Ver Informações Detalhadas de uma Linha (RF04)

1. O sistema deve oferecer informações detalhadas sobre uma linha numa caixa de diálogo lançada através da seleção de uma linha em uma listagem (RF02) (RF06);

2. Nessas informações detalhadas, o sistema deverá oferecer as seguintes opções ao usuário: visualizar o conjunto de pontos pelos quais esta linha passa (RF15); visualizar os horários desta linha (em um determinado ponto e dia da semana); adicionar linha aos favoritos; iniciar o cadastro de um alerta sobre esta linha (RF09); configurar o alerta de descida sobre esta linha (RF10).

Listagem de Alertas (RF07)

1. O sistema deve exibir uma listagem com todos os alertas de aproximação de ônibus cadastrados;
2. O sistema deve permitir ao usuário cadastrar novos alertas;
3. O sistema deve permitir ao usuário editar alertas existentes;
4. O sistema deve permitir ao usuário excluir alertas existentes;
5. O sistema deve oferecer um tipo de alerta especial, denominado Alerta de Descida. O funcionamento deste alerta consiste em, durante uma viagem, alertar o usuário que o próximo ponto é o seu ponto de destino, ou seja, o ponto de descida.

Rastreamento de Ônibus em tempo real (RF19)

1. O sistema deve ser capaz de enviar requisições sobre o posicionamento de um ou mais ônibus à um servidor, bem como, a partir das respostas obtidas, exibir esse posicionamento no mapa (RF01);
2. Além da informação sobre o geoposicionamento, a direção de deslocamento do ônibus também deve ser obtida através da requisição ao servidor.

5.1.3. Casos de Uso

Os casos de uso descrevem o comportamento esperado do produto em uma hipotética interação com o usuário. Eles normalmente são expressos através de diagramas e fluxos de caso de uso (os fluxos usualmente são representados em forma de tabela). Os diagramas de casos de uso descrevem os relacionamentos entre dos casos de uso entre si e com os atores, enquanto que os fluxos descrevem com detalhes cada caso de uso. Nesta seção será apresentado apenas um caso de uso completo, portanto tem-se na Figura 1 o diagrama global dos casos de uso.

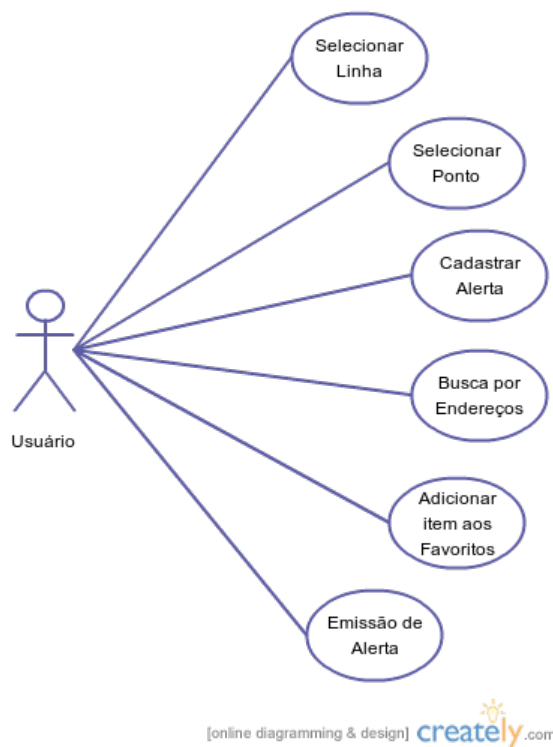


Figura 1. Diagrama global de Casos de Uso.

A seguir apresenta-se caso de uso Selecionar Linha completo, a fim de exemplificar uma amostra do trabalho desenvolvido neste tipo de artefato. A presença de apenas um caso de uso nesta seção também se justifica devido ao fato de que tudo o que está nos casos de uso também está nos Requisitos Funcionais (RF02 e RF04), mas sob um ponto de vista distinto.

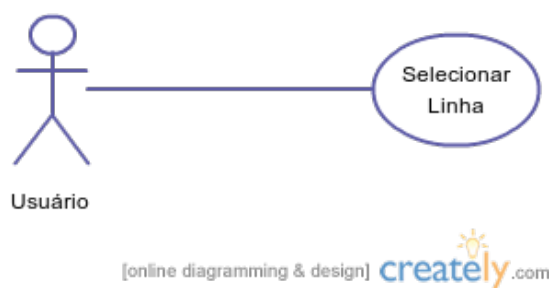


Figura 2. Diagrama do caso de uso Selecionar Linha.

Tabela 1. Fluxo do caso de uso Selecionar Linha

Resumo	O usuário deseja selecionar uma linha de transporte para poder então visualizar informações sobre localização dos ônibus, pontos e horários.
Ator	Usuário
Pré-Condições	O sistema tem conhecimento das linhas, pontos e respectivos horários.
Pós-Condições	O usuário encontra e seleciona uma linha, podendo então visualizar informações detalhadas sobre a mesma.
Fluxo Principal	<ol style="list-style-type: none"> 1. O usuário informa ao sistema que deseja selecionar uma linha; 2. O sistema exibe as linhas cadastradas; 3. O usuário seleciona uma linha, sendo então capaz de visualizar os ônibus que a estão cobrindo em tempo real, pontos pelos quais essa linha passa, seus horários, bem como configurar um alerta para essa linha; 4. Após navegar nas informações disponíveis desejadas, o usuário finaliza a operação.
Fluxo Alternativo	<p>* A qualquer momento, o usuário cancela a operação;</p> <ol style="list-style-type: none"> 1a. O usuário deseja selecionar uma linha a partir dos Favoritos: <ol style="list-style-type: none"> 1 O sistema exibe as linhas presentes na lista de Favoritos: <ol style="list-style-type: none"> 1a. O usuário seleciona uma linha e prossegue para o passo 4; 1b. O usuário não encontra a linha desejada e finaliza a operação; 3a. O usuário digita o nome ou parte do nome de uma linha, a fim de filtrar a listagem original de linhas, assim encontrando mais facilmente a linha desejada;
Fluxo de Exceção	<ol style="list-style-type: none"> 3b. O usuário não encontra a linha desejada: <ol style="list-style-type: none"> 1 Usuário finaliza operação.

Além disso, também apresenta-se o Diagrama de Sequência derivado deste Caso de Uso. Os Diagramas de Sequência são modelos baseados em UML que descrevem a sequência das interações entre o usuário e os módulos do sistema, bem como entre os próprios módulos. Seu objetivo é descrever com um maior nível de detalhamento as ações do sistema com base em interações hipotéticas com o usuário ou entre os módulos do sistema [Sommerville 2007].

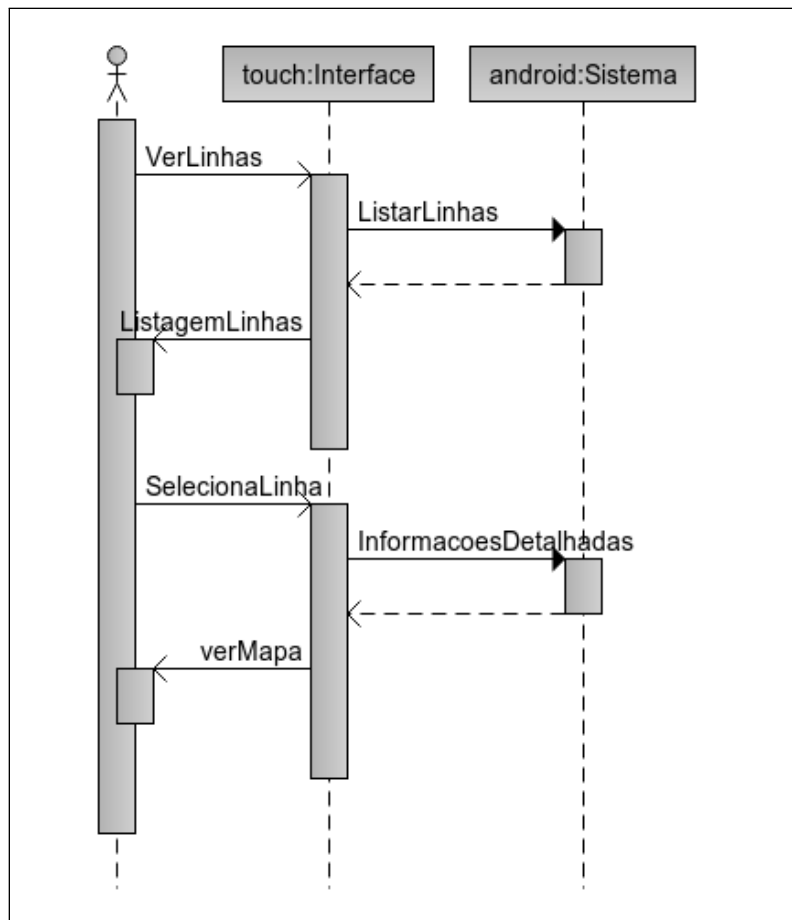


Figura 3. Diagrama de Sequência do caso de uso Selecionar Linha.

5.2. Mockups e Protótipo

Após a abordagem dos principais artefatos de ES, esta seção irá tratar a questão do Design de Interface, considerando princípios de Interação Humano-Computador. Os artefatos de IHC apresentados são os Mockups e os Protótipos.

5.2.1. Mockups da Telas

Os Mockups projetados são representações de interface de baixa fidelidade, e foram desenvolvidos com o auxílio do *software* FluidUI [FluidSoftware 2013]. Nesta seção serão listados alguns dos principais Mockups desenvolvidos.

O *design* de aplicações para dispositivos móveis apresenta algumas mudanças em relação às demais plataformas. Isso se deve principalmente ao tamanho reduzido da tela, o qual força os desenvolvedores a pensarem em maneiras simples e objetivas para apresentar as funções do produto ao usuário.

Por exemplo, em [Barbosa e da Silva 2010] uma consideração importante durante o design de IHC são os acessos ubíquos. Eles definem acessos às principais funções do produto, e portanto devem estar disponíveis à qualquer momento da interação. No

entanto, em dispositivos móveis nem sempre é possível apresentar uma lista com componentes de acesso ubíquos.

Sendo assim, a adaptação para esse caso é trazer, na tela inicial do sistema, atalhos para as principais funções do produto e, nas demais telas, oferecer um atalho direto para a tela inicial. Para o Android, em específico, há uma convenção que sugere a utilização do estilo *dashboard* (ou seja, um *layout* de grade com ícones e uma possível descrição adjacente para cada função [Google 2010]) para a tela inicial do aplicativo. A Figura 5.2.1 ilustra o Mockup da tela inicial.

Já a Figura 5.2.1 apresenta um Mockup baseado naquilo que foi descrito no requisito funcional Listagem de Linhas (RF02). Ao clicar na região principal, com o nome da linha, o usuário pode obter as informações detalhadas sobre a linha, enquanto que ao clicar na seta para a direita, a interface muda para uma exibição da linha no mapa.

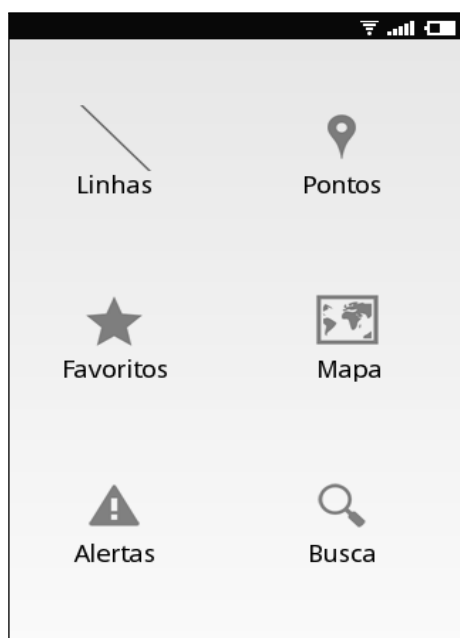


Figura 4. Mockup da tela inicial do sistema.

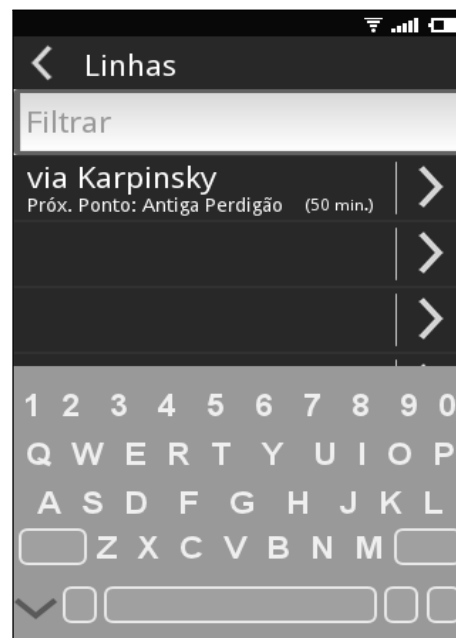


Figura 5. Mockup da tela com a listagem das linhas.

A elaboração dos Mockups foi capaz de abranger o sistema como um todo, isto é, todos os Requisitos Funcionais (definidos nos sub-item 5.1.2). Caso seja de interesse do leitor, todos os Mockups estão disponíveis para consulta no Material Suplementar B [Fedrecheski 2013b].

5.2.2. Protótipo

O escopo da prototipagem abordada neste trabalho reside na interação do usuário com o sistema. Portanto, o protótipo desenvolvido está focado somente no *design* da interface, deixando de lado – pelo menos por enquanto – as questões de funcionalidade.

Os protótipos foram baseadas nos Mockups – os quais, por sua vez, foram gerados a partir dos Requisitos Funcionais. Logo, as interfaces aqui prototipadas derivaram,

em última instância, dos requisitos funcionais, sendo eles: Listagem de Linhas (RF02), Listagem de Pontos (RF03), Exibição em Mapa (RF01), Listagem de Favoritos (RF06), Listagem de Alertas (RF04) e Busca por Endereço (RF11).

Como comentado anteriormente, o acesso inicial do *software* deve oferecer acessos globais das principais funcionalidades ao usuário. O motivo dessa abordagem, além de facilitar a utilização do *software*, reside em mostrar ao usuário “aquilo que ele pode fazer com o *software*”, sem complicações.

Assim como a página inicial do aplicativo (Figura 5.2.2) foi projetada com uma fidelidade considerável em relação ao seu Mockup, a Listagem das Linhas (Figura 5.2.2) também seguiu tal padrão. Nela, além do nome de cada linha, é possível visualizar o próximo ponto em que um ônibus deverá passar (baseado na tabela estática de horários), bem como o tempo calculado para que ocorra tal evento. Além disso, há também um botão cujo *affordance* é “mais informações sobre um determinado item”, que transfere a interação para uma visualização em mapa. A Listagem de Pontos é muito semelhante à Listagem de Linhas, pois os valores “Linha” e “Ponto” são apenas invertidos de uma para a outra.

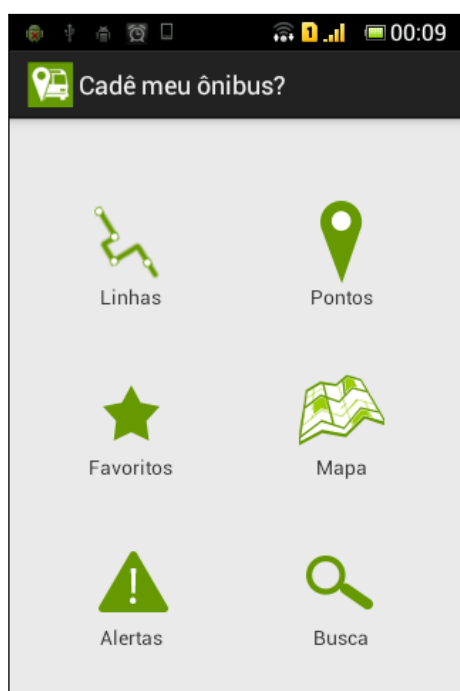


Figura 6. Tela inicial do sistema.

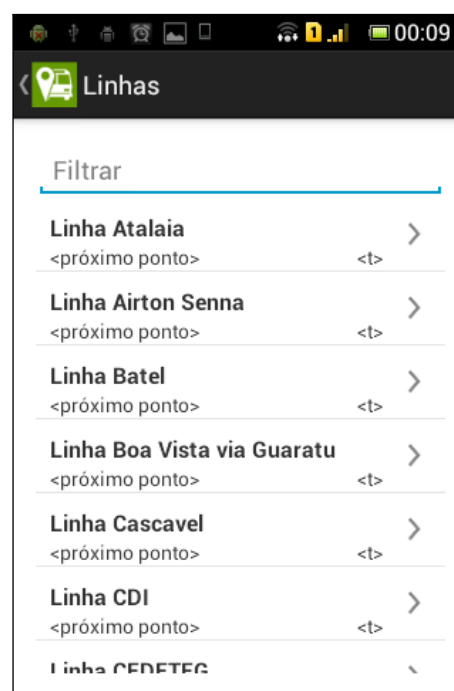


Figura 7. Listagem das Linhas.

As próximas telas a serem abordadas não foram apresentadas previamente neste artigo através de Mockups, porém, como supracitado, encontram-se disponíveis no Material Suplementar B [Fedrecheski 2013b].

A Listagem de Favoritos (Figura 5.2.2) exibe listagens de Linhas, Pontos e Rotas marcados como favoritos, e pode ser acessada através do menu principal no ícone cuja legenda é “Favoritos”.

A Exibição em Mapa (Figura 5.2.2) deve carregar um mapa obtido a partir da utilização da Google Maps API. O objetivo final, quando as funcionalidades da aplicação

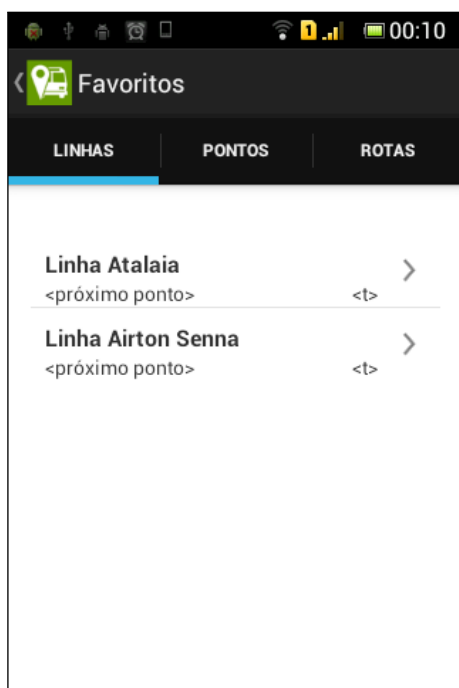


Figura 8. Listagem de Favoritos.



Figura 9. Exibição em Mapa.

tiverem sido implementadas, é carregar o mapa centralizado na posição do usuário e exibir todos os pontos existentes na cidade.

Como mencionado no requisito Listagem de Alertas (RF07), o sistema deve exibir uma listagem com todos os alertas cadastrados. Há um tipo especial de alerta (Alerta de Descida) que estará sempre presente no sistema, podendo estar ativado ou desativado. Por esse motivo, decidiu-se que ele estará sempre presente no topo desta listagem. Os demais alertas aparecerão listados logo abaixo do Alerta de Descida. A partir desta listagem o usuário deverá ser capaz de selecionar e então editar ou excluir os alertas existentes. A Figura 5.2.2 apresenta a interface prototipada para esta listagem.

A função de Busca por Endereços (RF11), previamente esboçada nos Mockups sem a utilização de abas, faz uso das mesmas na versão prototipada (Figura 5.2.2) para o câmbio entre as opções Linhas, Pontos e Rotas. Essa escolha justifica-se devido ao fato de que ela segue o padrão utilizado na Listagem de Favoritos e que o uso de abas para alternar entre interfaces relacionadas na *ActionBar*⁸ do Android é encorajado pela comunidade de desenvolvimento [Google 2013].

O retorno da função de busca de Linhas por endereço será: um conjunto de Linhas tal que cada linha pertencente a esse conjunto tenha pelo menos um ponto coberto pelo raio estabelecido pelo usuário. Na sequência, o usuário pode selecionar a linha desejada, para então visualizar no mapa o conjunto de pontos que a compõe.

Os protótipos desenvolvidos tiveram uma abrangência significativamente menor do que os Mockups em termos de completude do sistema. Isso se deve à maior comple-

⁸*ActionBar* é uma característica adicionada recentemente ao Android (a partir da versão 3.0) que provê algumas ações e acessos globais ao usuário, como por exemplo um atalho para a página inicial do aplicativo ou um conjunto de abas.



Figura 10. Listagem de alertas.



Figura 11. Busca de Linhas por endereços.

xidade e atenção para cada detalhe que se faz necessária nesta fase. Mesmo que o termo “protótipo” sugira uma implementação mais propensa à falhas e com menos qualidade que um produto final (portanto com menos riscos), ainda considera-se o protótipo desenvolvido em um contexto evolutivo, ou seja, o trata como já sendo parte de um produto final.

6. Conclusão

Neste trabalho apresentou-se o projeto de um aplicativo para celulares Android cujo intuito é auxiliar os usuários do transporte público urbano a obterem informações sobre esse tipo de serviço. O grande diferencial – mesmo que já não tão inédito, como vimos na Seção 2 – é a funcionalidade de rastreamento dos veículos em tempo real. Além do mais, informações estáticas sobre linhas, pontos e horários também são fornecidas através do aplicativo proposto.

Através de um estudo preliminar de algumas técnicas de Engenharia de Software e Interação Humano-Computador, e um posterior desenvolvimento das mesmas, foi possível delimitar um escopo bem definido para a proposta de solução inicialmente apresentada. Além disso, a aplicação dos processos de ES e IHC neste trabalho contribuiu para um melhor entendimento dos conteúdos abordados por cada uma das técnicas.

A utilização dessas técnicas serviu de base para a elaboração de um modelo de interação de baixa fidelidade (Mockups), o que culminou no desenvolvimento de um Protótipo voltado ao *design* de interface da aplicação proposta.

As principais dificuldades enfrentadas no desenvolvimento deste trabalho consis-

tem na prototipagem do aplicativo Android, bem como na definição do modelo do documento de ERSw a ser utilizado e a elaboração das tabelas de fluxo dos Caso de Uso.

Com excessão dos Diagramas de Classe, todos os objetivos estabelecidos no início deste trabalho foram alcançados. Além disso, um objetivo adicional (a criação de diagramas de sequência) também foi alcançado. Considera-se, portanto, que o trabalho desenvolvido atendeu significativamente às expectativas iniciais, inclusive superando-as em alguns casos.

7. Trabalhos Futuros

Como continuação deste trabalho, espera-se que o legado deixado por este projeto possa ser utilizado para futuras melhorias da solução apresentada. Além disso, também fica em aberto o estudo e a implementação do módulo servidor e da infraestrutura de rastreamento dos veículos em tempo real.

8. Agradecimentos

Agradeço primeiramente aos meus pais, que sempre me incentivaram a buscar aquilo que me fizesse sentir feliz – isso culminou em um forte interesse pelos estudos, e obviamente o seu apoio nesse sentido foi sempre incondicional. Agradeço também à minha estimada amiga Sádina, à quem devo parte do meu *insight* para o início deste trabalho, e por ter me apoiado todo o tempo que passou ao meu lado.

Um agradecimento especial ao meu orientador, Richard, que sempre foi muito além das questões técnicas e ocupou também uma posição de mentor durante os nossos trabalhos juntos. Agradeço também ao professor Josiel, que é um grande companheiro e foi meu co-orientador neste trabalho.

Agradeço às professoras Evanise e Ináli por efetuar análises preliminares de alguns artefatos produzidos durante a execução deste trabalho. Agradeço também ao meu amigo Rodrigo que elaborou o *design* dos ícones utilizados no protótipo, e à equipe do software FluidUI por ceder uma licença para o desenvolvimento dos Mockups deste trabalho. Finalmente, agradeço a todos os amigos e professores que me acompanharam durante esta jornada e a todos aqueles que contribuíram direta ou indiretamente para com a execução deste trabalho.

Referências

- [Barbosa e da Silva 2010] Barbosa, S. D. J. e da Silva, B. S. (2010). *Interação Humano-Computador*. Campus.
- [Fedrecheski 2013a] Fedrecheski, G. (2013a). Material suplementar a. <https://dl.dropboxusercontent.com/u/28722106/TCC/Material%20Suplementar%20A%20-%20ERSw.pdf>.
- [Fedrecheski 2013b] Fedrecheski, G. (2013b). Material suplementar b. <https://dl.dropboxusercontent.com/u/28722106/TCC/Material%20Suplementar%20B%20-%20Mockups.pdf>.
- [Ferris 2011] Ferris, B. (2011). *OneBusAway: Improving the Usability of Public Transit*. PhD thesis, University of Washington.
- [Figueiredo e Nakamura 2003] Figueiredo, C. M. S. e Nakamura, E. (2003). Computação móvel: Novas oportunidades e novos desafios.
- [FluidSoftware 2013] FluidSoftware (2013). Fluid ui. <https://www.fluidui.com/>.
- [Goldsmith *et al.* 2006] Goldsmith, S., Killorin, M., e Larson, E. (2006). The economic benefits of public transportation in anchorage.
- [Google 2010] Google (2010). Android ui design patterns. <https://dl.google.com/googlego/2010/android-android-ui-design-patterns.pdf>.
- [Google 2012] Google (2012). Nosso planeta mobile: Brasil. como entender o usuário de celular. <http://www.thinkwithgoogle.com/mobileplanet/pt-br/>.
- [Google 2013] Google (2013). Android developers. <http://developer.android.com>.
- [Gulliksen *et al.* 2005] Gulliksen, J., Goransson, B., Boivie, I., Blomkvist, S., Persson, J., e Cajander, A. (2005). Key principles for user-centred systems design.
- [Ipea 2012] Ipea (2012). Mobilidade urbana 2a edição análise preliminar dos dados coletados em 2011.
- [Lecheta 2010] Lecheta, R. R. (2010). *Google Android*. novatec, 1 edição.
- [Love 2004] Love, R. (2004). *Desenvolvimento do Kernel do Linux*. Ciência Moderna.
- [Paula Filho 2001] Paula Filho, W. P. (2001). *Engenharia de Software*. LTC, 1 edição.
- [Siqueira 2012] Siqueira, L. G. (2012). Aplicação bus tracker – oferecendo uma melhor experiência aos usuários do transporte urbano, a partir da utilização de informações de rastreamento veicular.
- [Sommerville 1995] Sommerville, I. (1995). *Software Engineering*. Addison-Wesley, 5 edição.
- [Sommerville 2007] Sommerville, I. (2007). *Software Engineering*. Addison-Wesley, 8 edição.
- [URBS 2012] URBS (2012). Urbs - urbanização de curitiba s/a. <http://www.urbs.curitiba.pr.gov.br/>.