



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO**

KAIQUE JOSÉ BATISTA CRUZ

**DESENVOLVIMENTO DE APLICATIVOS
MULTIPLATAFORMA: APLICAÇÃO PRÁTICA COM UMA
SOLUÇÃO PARA MOBILIDADE URBANA**

**Castanhal, Pará
2015**

KAIQUE JOSÉ BATISTA CRUZ

**DESENVOLVIMENTO DE APLICATIVOS
MULTIPLATAFORMA: APLICAÇÃO PRÁTICA COM UMA
SOLUÇÃO PARA MOBILIDADE URBANA**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação como requisito para a obtenção do grau de Bacharel em Sistemas de Informação, pela Universidade Federal do Pará – UFPA.

Orientador:
Prof. Me. Adailton Magalhães Lima

**Castanhal, Pará
2015**

KAIQUE JOSÉ BATISTA CRUZ

**DESENVOLVIMENTO DE APLICATIVOS
MULTIPLATAFORMA: APLICAÇÃO PRÁTICA COM UMA
SOLUÇÃO PARA MOBILIDADE URBANA**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação como requisito para a obtenção do grau de Bacharel em Sistemas de Informação, pela Universidade Federal do Pará – UFPA.

Banca examinadora

Prof.º. Me. Adailton Magalhães Lima (UFPA-CUNCAST)
Orientador

Prof. Dr. Marcos César da Rocha Seruffo
Faculdade de Computação (UFPA)
Membro da banca (interno)

Prof.º. Me. Anderson Jorge Serra da Costa (UEPA)
Universidade do Estado do Pará (UEPA)
Membro da banca (externo)

Castanhal, 18 de Setembro de 2015

Dedico este trabalho aos meus pais, meu irmão, e aos amigos que sempre estiveram ao meu lado, me dando apoio e força para eu seguir em frente e percorrer o caminho trabalhoso enfrentado na graduação e na vida.

AGRADECIMENTOS

Aos meus pais, que me ajudaram a concretizar este sonho, e que apesar das dificuldades não desistiram de minha formação. Que sempre acreditaram que este sonho seria possível, chegou o momento de comemorarmos esta vitória. Obrigado pelo amor, paciência e suporte durante essa jornada (Amo vocês!). Também agradeço ao meu irmão e aos outros familiares.

Aos meus amigos da faculdade, pelas zoeiras (não foram poucas, “KKKKK...”), pelo convívio durante esses quatro anos de graduação e espero que mantenhamos contato por mais tempo. Agradeço ao Joahannes e Daniel (parceiros de trabalhos e amigos da zoeira); Michel; Tiago; Renan; Waldemar; Josimar; Keyla; Susan; Tainara; ao Wesley, Patrick e Édylle (parceiros das idas para a especialização em Belém); Angélica; Erick; Karen; e André. Àqueles que não concluíram a faculdade comigo e os outros amigos que fiz na UFPA.

Aos meus amigos do Darwin (a galera do Darwin): Paulo Roberto, Luana, Carina, Débora, Luiz e Karol (o casal), Anderson. Aos parceiros do treino de Muay Thai (“Sawadee Krap”), Dança de Salão e o pessoal do condomínio.

Ao corpo docente da Faculdade de Computação pelo conhecimento que adquiri e o profissionalismo. Agradeço especialmente ao meu professor, chefe e orientador, Adailton Lima, responsável pelo auxílio para com este trabalho, por me apresentar uma nova tecnologia para desenvolvimento mobile, e pelas oportunidades. E a todas as pessoas que torceram por mim e contribuíram direta ou indiretamente para a realização desta grande etapa que se encerra.

“Cada segundo é tempo para mudar tudo para sempre”

Charles Chaplin

RESUMO

Atualmente o desenvolvimento de aplicações móveis é uma realidade e necessidade de mercado. Assim, a busca por melhores técnicas de desenvolvimento é essencial para ganhar produtividade e concorrência. No mercado atual de desenvolvimento mobile destacam-se três plataformas principais: Android, iOS e Windows Phone. Desenvolver aplicações nativas para cada uma dessas plataformas demanda tempo, dinheiro e pessoal qualificado, pois são necessários linguagens de programação e ambientes de desenvolvimento específicos para cada um. Porém, com o desenvolvimento multiplataforma é possível gerar *releases* para diferentes plataformas codificando em apenas um ambiente e linguagem, e portanto, ganha-se tempo, reduz custos e aumenta a produtividade da equipe de desenvolvimento. Este trabalho relata uma aplicação prática, onde foi realizada uma revisão bibliográfica e comparação entre *frameworks* multiplataforma existentes, e também a implementação de uma solução para mobilidade Urbana baseada no aplicativo Rota Urbana. Este produto desenvolvido foi submetido a testes alfa e beta para validação dos requisitos e funcionalidades implementadas. Neste trabalho também são apresentados os resultados das avaliações feitas pelos usuários beta.

Palavras-chave: Programação Móvel, Desenvolvimento Multiplataforma, Desenvolvimento Nativo, Mobilidade Urbana

LISTA DE FIGURAS

Figura 1. Desenvolvimento Nativo vs. Desenvolvimento Multiplataforma [Redda, 2012].....	21
Figura 2. Arquitetura do Framework PhoneGap [Popa, 2013]	29
Figura 3. Rota Urbana (Web) – Fonte: ROTA URBANA 2015	33
Figura 4. Rota Urbana (Android) – Fonte: ROTA URBANA 2015	33
Figura 5. Caso de Uso do Rota Urbana	34
Figura 6. Diagrama de Componentes	35
Figura 7. Tela Inicial do Rota Urbana Multiplataforma.....	37
Figura 8. Menu do Rota Urbana Multiplataforma.....	38
Figura 9. Marcadores de Inicio e Fim para Pesquisa de Rotas de Ônibus	38
Figura 10. Lista de Rotas de Ônibus Encontradas.....	39
Figura 11. Tela com a Rota Escolhida desenhada no Mapa.....	40
Figura 12. Emulador do Intel XDK.....	41

LISTA DE TABELAS

Tabela 1. Participação no Mercado das Plataformas Android, iOS e Windows Phone – Fonte Techmundo 2014	20
Tabela 2. <i>Frameworks</i> para Desenvolvimento Multiplataforma.....	22
Tabela 3. Comparação de Componentes entre os <i>Frameworks</i>	24
Tabela 4. Comparação entre os <i>Frameworks</i> quanto a Plataforma <i>Mobile</i>	24
Tabela 5. Comparação entre <i>Frameworks</i> quanto a Linguagem e o Tipo de Licença	25
Tabela 6. Requisitos do Rota Urbana Multiplataforma.....	34
Tabela 7. Respostas da primeira, segunda e terceira perguntas do formulário	51
Tabela 8. Respostas da quarta, quinta e sexta perguntas do formulário	51

LISTA DE GRÁFICOS

Gráfico 1. Resultado da Primeira Pergunta do Formulário	42
Gráfico 2. Resultado da Quinta Pergunta do Formulário	43

LISTA DE ABREVIATURAS E SIGLAS

AJAX	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
GPS	<i>Global Positioning System</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
XHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>
HTML	<i>Hypertext Markup Language</i>
NFC	<i>Near Field Communication</i>
UI	<i>User Interface</i>
WORA	<i>Write once, run anywhere</i>
SMS	<i>Short Message Service</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	15
1.1.1	Gerais	15
1.1.2	Específicos	15
1.2	JUSTIFICATIVA E MOTIVAÇÃO	16
1.3	PROBLEMA E SOLUÇÃO PROPOSTA	17
1.4	METODOLOGIA	17
1.5	ORGANIZAÇÃO DO TRABALHO	18
2	REVISÃO BIBLIOGRÁFICA	19
2.1	DESENVOLVIMENTO MULTIPLATAFORMA	19
2.1.1	<i>Frameworks</i> Multiplataforma para Aplicação Prática	21
2.2	TRABALHOS CORRELATOS	25
3	APLICAÇÃO PRÁTICA	28
3.1	TECNOLOGIAS UTILIZADAS	28
3.1.1	Intel XDK	28
3.1.2	PhoneGap	29
3.1.3	jQuery Mobile	30
3.1.4	<i>Hypertext Markup Language</i>	30
3.1.5	<i>Cascading Style Sheets</i>	31
3.1.6	JavaScript	31
3.2	ROTA URBANA MULTIPLATAFORMA	32
3.2.1	Requisitos e Funcionalidades	32
3.2.2	Arquitetura da Solução	35
3.2.3	Interface da Aplicação	36
3.3	TESTES REALIZADOS	40
3.4	RESULTADOS	42
4	CONCLUSÃO E TRABALHOS FUTUROS	44
	REFERÊNCIAS	46
	APÊNDICES	49

APÊNDICE I – QUESTIONÁRIO PARA FEEDBACK.....	50
APÊNDICE II – RESULTADOS DOS FEEDBACKS	51

1 INTRODUÇÃO

Os dispositivos móveis sofreram grandes mudanças ao longo dos anos. Antes dos *smartphones*, o celular era usado basicamente para ligações e troca de mensagens de texto por Short Message Service (SMS). A chegada dos *smartphones* trouxe melhorias no hardware dos aparelhos, e possibilitou o avanço rápido nas funcionalidades e poder dos seus sistemas operacionais. Com isso, diversos aplicativos foram e são desenvolvidos para facilitar o acesso à informação e solução de problemas [Silva e Santos, 2014]. Com relação a esses avanços [Café, 2012] complementa:

Desde sua criação, os aparelhos celulares passaram por um processo de evolução constante. Diversos fabricantes do mundo todo investiram tempo e recursos financeiros para criar novos aparelhos, cada vez mais modernos e desejados pelos consumidores. Dentre os avanços obtidos destacam-se a miniaturização dos aparelhos, melhor qualidade dos sinais das operadoras (2G, 3G, 4G) e novas funcionalidades (câmera, acelerômetro, GPS¹, bluetooth). A redução do preço dos aparelhos celulares, tornando-os mais acessíveis, possibilitou a aquisição dos mesmos por parte da população menos favorecida financeiramente.

Diversos fabricantes de aparelhos celulares destacam-se no mercado: Apple, Samsung, Motorola, LG, BlackBerry, por exemplo. Cada fabricante adota um Sistema Operacional para seus celulares (Android, iOS, Windows Phone, Symbian e BlackBerry OS, entre outros).

O mercado de desenvolvimento de aplicações para esses dispositivos passou a ter a seguinte característica, de acordo com [Zimmerman, 2013]: os desenvolvedores tinham que criar aplicações nativas para somente uma plataforma por vez. Isto resulta em muitos custos, tanto financeiramente quanto em tempo para que uma aplicação seja desenvolvida e testada de acordo com cada plataforma e sistema operacional alvo.

Com essas dificuldades enfrentadas pelos desenvolvedores, surgiram os *cross-platform mobile frameworks*, que são *frameworks* para o desenvolvimento multiplataforma. Nesta abordagem uma das formas é onde o desenvolvedor codifica com as tecnologias web, JavaScript, *Hypertext Markup Language* (HTML) e *Cascading Style Sheets* (CSS3), para gerar *releases* de aplicações para diferentes plataformas móveis. Este trabalho objetiva trazer esse novo modo de implementação de aplicações por meio de uma Aplicação Prática.

¹ *Global Positioning System*, em português, Sistema de Posicionamento Global. É um sistema de posicionamento por satélite que fornece a um aparelho receptor móvel a localização geográfica.

1.1 OBJETIVOS

Nas subseções seguintes serão apresentados os objetivos gerais e específicos deste trabalho de conclusão de curso.

1.1.1 Gerais

O objetivo deste trabalho é identificar quais as abordagens e *frameworks* multiplataforma existentes no mercado e classificá-los frente às necessidades de aplicações mobile existentes, onde será realizado o desenvolvimento de uma versão do aplicativo existente na plataforma Android Rota Urbana (o qual fornece as rotas de ônibus da cidade de Belém, permitindo que o usuário saiba qual a melhor rota a ser seguida) para *mobile* multiplataforma – para as plataformas Android, iOS e Windows Phone –. Após isso, obter *feedback* dos usuários para verificar a aceitação do público referente a nova versão do aplicativo.

1.1.2 Específicos

- Pesquisar opções de *frameworks* disponíveis no mercado para desenvolvimento mobile multiplataforma;
- Definir critérios de comparação e classificar os *frameworks* encontrados;
- Projetar a arquitetura da solução a ser desenvolvida;
- Fazer a migração de uma versão da aplicação Rota Urbana utilizando um dos *frameworks* escolhidos;
- Realizar o desenvolvimento de uma aplicação para verificar as funcionalidades junto aos usuários do aplicativo migrado, por meio de um formulário que estará acessível no aplicativo e na rede social do Rota Urbana.

1.2 JUSTIFICATIVA E MOTIVAÇÃO

No cenário de desenvolvimento de aplicativos nativos, os programadores normalmente “devem esperar restrições e custos associados com o desenvolvimento e implantação de certas plataformas, tais como a necessidade de uma licença de desenvolvedor da Apple e aprovação da Apple para distribuir aplicativos para o iTunes Apps Store” [Smutný, 2012]. A fragmentação dos dispositivos também é um problema, com diferentes sistemas operacionais móveis e APIs de programação (sendo que elas podem ser restritas para certas aplicações e linguagens de programação) [Smutný, 2012].

Este trabalho é justificado pelo desafio de desenvolver uma aplicação nativa, apresentando os *frameworks* multiplataforma, os quais permitem gerar *releases* a mais de uma plataforma com apenas uma linguagem de programação. Contudo, o mercado de desenvolvimento com *frameworks* multiplataforma está em crescimento.

Devido ao fato de haver a necessidade de criar uma aplicação móvel que funcione em plataformas diferentes. Torna-se estratégico desenvolver aplicativos móveis com uso de um *framework* para *mobile* multiplataforma, permitindo aplicar o conhecimento em uma linguagem de programação, deixando que o *framework* gerasse os *releases* para as plataformas que se deseja, permitindo a equipe de desenvolvimento aumentar a produtividade e a empresa ter maior alcance no mercado.

Em relação ao cenário *cross-platform* [Redda, 2012] recomenda:

“Existem várias ferramentas multi-plataforma no mercado e cada um deles tem suas próprias características únicas que os torna indispensável e ao mesmo tempo não têm algumas funcionalidades chave que um desenvolvedor pode realmente deseja explorar. Decidir sobre um framework multiplataforma que resolva todos os problemas parece uma expectativa irreal. Os desenvolvedores devem planejar meticulosamente o que eles esperam de uma ferramenta, deve saber com antecedência o seu público-alvo e plataforma objetivo, deve ter uma ideia clara do que eles pretendem explorar nos recursos do dispositivo, e deve decidir como querem implementá-los.”

Em virtude a esse desafio para com o desenvolvimento multiplataforma, propõe-se fazer a migração do Rota Urbana com utilização de um *framework* para *mobile* multiplataforma, migrando-se para as versões Android, iOS, Windows Phone, que são as plataformas que hoje possuem o maior número de usuários no mercado.

1.3 PROBLEMA E SOLUÇÃO PROPOSTA

“O desenvolvimento de um aplicativo nativo para aparelhos celulares exige conhecimentos específicos a respeito das tecnologias utilizadas pela plataforma na qual se deseja executá-lo” [Silva e Santos, 2014].

Segundo [Sambasivan et al., 2011]: “A principal desvantagem de aplicativos nativos é que eles precisam ser desenvolvidos separadamente para cada plataforma, aumentando assim o tempo de desenvolvimento, custo e o esforço.”

Portanto, será apresentada uma solução de migração parcial das funcionalidades presentes no aplicativo Rota Urbana, embora seja necessário acrescentar trechos de códigos específicos de cada plataforma, pois existem funcionalidades disponibilizadas pelos frameworks que não são compatíveis em certas plataformas. Como diz [Myer, 2012]:

“Se você estiver trabalhando com vários dispositivos, você precisará de ambientes separados para cada projeto. Por exemplo, você não será capaz de manter o seu projeto Android PhoneGap com o Xcode. outra maneira de se enquadrar esta é você potencialmente simplificar o quanto seu código web deve mudar para ser executado em diferentes dispositivos, mas você ainda deve manter projetos separados.”

Dessa forma, para cada plataforma acrescentasse código (por exemplo, modificações no layout da aplicação ou no arquivo de configuração) ou retirasse código (funções não implementadas pela incompatibilidade).

Por meio de um estudo bibliográfico identificar o *framework* que melhor se adequa a solução a ser desenvolvida. Realizando avaliações para com os releases dos ambientes móveis escolhidos a fim de identificar o quão útil o desenvolvimento *mobile* multiplataforma pode ser.

1.4 METODOLOGIA

Realizar um estudo bibliográfico referente ao desenvolvimento *mobile* multiplataforma, para identificar o *framework* que melhor se adequa na aplicação prática.

Por meio de uma engenharia reversa, com base nas versões web e Android existentes, desenvolver o aplicativo utilizando o *framework* escolhido em conjunto com as tecnologias *web*.

Com o aplicativo desenvolvido, testá-lo em campo a solução desenvolvida em diferentes dispositivos, tanto em laboratório quanto liberando o uso das aplicações ao público final. O público final (usuários beta) poderão responder a um questionário onde será avaliado o aplicativo e o que se entende por desenvolvimento multiplataforma.

1.5 ORGANIZAÇÃO DO TRABALHO

Os capítulos posteriores estão organizados da seguinte forma:

No capítulo 2, apresenta-se a revisão bibliográfica, que fala sobre o desenvolvimento multiplataforma, seus conceitos e dados sobre o mercado de aplicativos móveis. Assim como um comparativo entre *frameworks* multiplataforma, no intuito de escolher o melhor para a aplicação prática, e apresentação de trabalhos correlatos.

No capítulo 3, tem-se a descrição da aplicação prática realizada, com apresentação das ferramentas utilizadas no desenvolvimento e da solução proposta, os testes feitos e resultados obtidos.

No capítulo 4 é feita a conclusão deste trabalho e posposta de trabalhos futuros.

Por fim, são apresentadas as referências e os apêndices I e II.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentadas as características do desenvolvimento multiplataforma, expondo a situação do mercado de desenvolvimento de aplicativos, conceitos e *frameworks* disponíveis. Também são destacados os principais trabalhos correlatos a este.

2.1 DESENVOLVIMENTO MULTIPLATAFORMA

Atualmente as aplicações mobile são vistas muitas vezes não como um diferencial de mercado, mas como necessidade do mesmo. Notam-se diversos sistemas operacionais mobile no mercado, Android², iOS³, Windows Phone⁴, BlackBerry OS⁵ e outros, todavia, desenvolver uma aplicação que funcione em todos esses sistemas operacionais é custoso devido a diversos motivos, como linguagens e ambientes de desenvolvimento específicos para cada um [Sambasivan et al., 2011].

Para desenvolver para a plataforma Android é necessário ter conhecimento na linguagem Java; já para a plataforma iOS deve-se conhecer a linguagem Objective-C e mais recentemente a linguagem Swift; para desenvolver nativamente para a plataforma Windows Phone deve-se conhecer a linguagem C# e do ambiente de desenvolvimento .NET. Mesmo que o programador apresente conhecimentos sólidos em algoritmos, as sintaxes destas linguagens são diferentes, e para cada linguagem existe uma curva de aprendizagem para ganhar proficiência. Isto gera gastos também, já que é difícil um programador que tenha conhecimento para desenvolvimento em mais de uma plataforma, e mesmo que saiba mais de uma linguagem de programação com proficiência, sem dúvida será um programador com custo elevado ao projeto.

² O Android é sistema operacional baseado no núcleo do Linux, sendo projetado para *smartphones e tablets*, com interface para relógio de pulso, TVs e carros, atualmente desenvolvido pela empresa Google. Segue o endereço eletrônico para mais detalhes: <https://www.android.com/>.

³ O iOS é o sistema operacional para iPhone, iPad e iPod Touch, desenvolvido pela Apple Inc. Segue o endereço eletrônico para mais informações: <http://www.apple.com/br/ios/>.

⁴ O Windows Phone é o sistema operacional para *smartphones*, desenvolvido pela Microsoft. Segue o *site* da plataforma: <https://www.windowsphone.com/pt-br>.

⁵ O BlackBerry OS é o sistema operacional para *smartphones e tables* desenvolvidos pela empresa canadense BlackBerry. Segue o *site* da empresa: <http://global.blackberry.com/en.html>.

Segundo ilustrado na Tabela 1, o mercado de desenvolvimento para Android é bastante favorável levando em consideração o grande número de usuários, seguido da plataforma iOS e da plataforma para Windows Phone. Porém, a receita por aplicativo é muito maior na plataforma iOS da Apple, o que significa que seus usuários estão mais dispostos a pagar pela aplicação ou serviços associados a esta [Louis, 2013] e [Hamann, 2014].

Tabela 1. Participação no Mercado das Plataformas Android, iOS e Windows Phone – Fonte Techmundo 2014

	Android	iOS	Windows Phone
<i>Market share (global)</i>	84,70%	11,70%	2,50%
Usuários ativos	1bilhão	800 milhões	60 milhões
Numero de <i>apps</i>	1,3 milhão	1,2 milhão	300 mil
Media de <i>downloads</i> por <i>app</i> *	60 mil	40 mil	4 mil
Receita media por <i>download</i> *	US\$ 0,018	US\$ 0,10	US\$ 0,15
Receita media por <i>app</i> *	US\$ 1.125	US\$ 21.276	US\$ 2.222

*Dados de 2013 (Forbes)

Com o objetivo de amenizar esse problema surgiram os chamados *webapps*. Segundo [KESSI et al., 2008, p. 2], a web é *cross platform*, permitindo escrever páginas web que irão funcionar em diferentes sistemas operacionais e plataformas como Windows, Mac OS X, Linux, iPhone, Android, Windows Phone, BlackBerry e outros. As plataformas *mobile* possuem seus *browsers*, os quais interpretam *scripts* do JavaScript, e já suportam HTML5 (versão 5 do HTML) e CSS3 (versão 3 do CSS). *Webapps* são desenvolvidos utilizando as tecnologias web e não linguagem nativa da plataforma, eles são acessados pelo navegador e não são instalados no dispositivo.

Como alternativa, surgiram os *hybrid apps* ou aplicativos híbridos, que são aplicações que utilizam HTML e CSS para interface do usuário, e JavaScript para a lógica de negócios. Todavia, os aplicativos híbridos têm acesso limitado ao *hardware* e certas funcionalidades dos aparelhos (Acelerômetro, Geolocalização, Câmera, Contatos, Banco de Dados, etc.), mas acessam mais funcionalidades que os *webapps*. Além disso é possível publicá-los nas lojas de aplicativos, como: Google Play e Apple Store.

[Café, 2012] cita as características de uma aplicação *cross-platform*:

A ideia de se criar um único aplicativo que seja compatível com diversas plataformas simultaneamente permite que o desenvolvedor foque em uma única linguagem, uma única ferramenta de desenvolvimento e API, possibilitando o reuso do código, desenvolvimento mais rápido, diminuição da complexidade e recursos a serem utilizados.

No desenvolvimento de uma aplicação multiplataforma ou *cross platform*, aplica-se o conceito WORA (*Write Once, Run Anywhere*, em português, Escreva Uma Vez, Execute Em

Qualquer Lugar). Conceito este que permite a reutilização de código e interoperabilidade, e o desenvolvimento de aplicativos *cross platforms*, seja para *web*, *desktops* e *mobile*. Como exemplo tem-se a linguagem Java, para a plataforma *desktop* (J2SE), *web* (J2EE) e *mobile* (J2ME). A linguagem é a mesma para todas as três plataformas, sendo diferenciadas pelo uso de *Applications Programming Interface* (API – em português, Interface de Programação de Aplicativos) específicas de cada uma.

A Figura 1 expõe que quanto mais próximo do desenvolvimento web, menos se gasta e menos se perde tempo em comparação ao desenvolvimento de aplicações nativas, contudo perde-se em desempenho e interface do usuário. Segundo [Palmieri, Singh e Cicchetti, 2012] diminui-se o tempo de desenvolvimento e manutenção (WORA), conhecimento da API, maior facilidade no desenvolvimento e o aumento de participação no mercado.

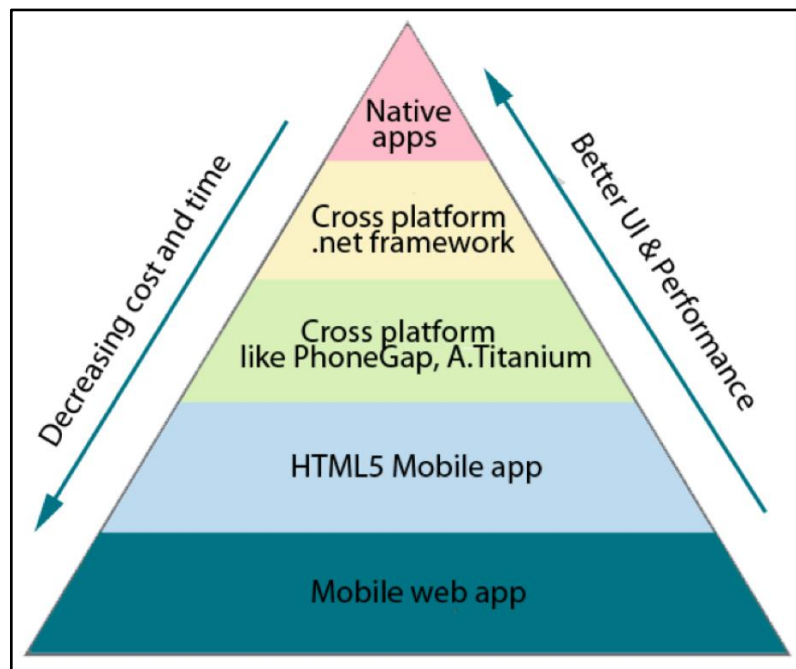


Figura 1. Desenvolvimento Nativo vs. Desenvolvimento Multiplataforma [Redda, 2012]

Na subseção seguinte serão apresentados os *frameworks* que foram escolhidos para o desenvolvimento da aplicação prática. Por meio de um processo comparativo foi escolhido somente um *framework* para implementação da solução de mobilidade urbana.

2.1.1 Frameworks Multiplataforma para Aplicação Prática

Existem diversas plataformas, *frameworks* e bibliotecas para o desenvolvimento multiplataforma. A Tabela 2 mostra essa diversidade. Algumas delas são utilizadas para

criação de jogos. Abaixo é explicado o que é uma plataforma, um *framework* e uma biblioteca segundo a definição de [Hartmann, Stead e Degani, 2011].

- **Plataforma:** Um conjunto de *frameworks*, ferramentas e serviços que não só permitem ao usuário construir um aplicativo móvel completo, mas também para configurar, empacotar e distribuir para lojas de aplicativos ou a nuvem.
- **Framework:** Um conjunto de bibliotecas, componentes de software e padrões de arquitetura que fornece ao desenvolvedor um conjunto de ferramentas abrangente para construir um aplicativo móvel completo.
- **Biblioteca:** Pequeno, kit de ferramentas independente que oferece uma funcionalidade muito específica para o usuário. Normalmente utilizado em conjunto com outras bibliotecas e ferramentas para criar um aplicativo móvel completo. Como os *widgets* UI (*User Interface*) e bibliotecas gráficas 3D.

Tabela 2. Frameworks para Desenvolvimento Multiplataforma

<i>Framework</i>	Endereço Eletrônico	<i>Framework</i>	Endereço Eletrônico
Apache Flex	http://flex.apache.org/	Kivy	http://kivy.org/#home
Agate	http://www.applusform.com/en/	Marmalade	https://www.madewithmarmalade.com/
Appcelerator Titanium Mobile	http://www.appcelerator.com/	Moai	http://getmoai.com/
AppConKit	http://www.weptun.com/	Mobify.js	http://www.mobify.com/mobify.js/
AppGyver Steroids	http://www.appgyver.com/steroids	Mobione	http://www.genuitec.com/
Application Craft	http://www.applicationcraft.com/	The-M-Project	http://www.the-m-project.org/
ChocolateChip-UI	http://chocolatechip-ui.com/	NeoMAD	http://neomades.com/en/
CNET iPhone UI*	https://code.google.com/p/ciui-dev/	NimbleKit	http://nimblekit.com/index.php
Corona	https://coronalabs.com/	NSB/AppStudio	https://www.nsbasic.com/
DHTMLX Touch	http://dhtmlx.com/docs/products/dhtmlxTouch/	PhoneGap	http://phonegap.com/
Dojo Mobile	http://dojotoolkit.org	qooxdoo	http://qooxdoo.org/
Framework7	http://www.idangero.us/framework7/#.Vb7k4fmGNT8	Quick Connect	http://quickconnect.pbworks.com/w/page/9183363/FrontPage
Gideros	http://giderosmobile.com/	RareWire	http://www.rarewire.com/
GWT mobile webkit + gwt mobile ui	https://code.google.com/p/gwt-mobile-webkit/	Rhodes	http://rhone.com/
Handheld Designer	http://handhelddesigner.com/	Sencha Touch	https://www.sencha.com/products/touch/#overview
Ionic	http://ionicframework.com/	ShiVa 3D	http://www.shivaengine.com/
iPhone Universal	https://code.google.com/p/iphone-universal/	Trigger.io	https://trigger.io/
iUI	http://www.iui-js.org/	ViziApps	http://www.viziapps.com/
iWebKit	http://snippetspace.com/	V-Play	http://v-play.net/
Jo HTML5 Mobile App	http://joapp.com/	WebApp.net	http://trywebapp.net/

Framework			
JQ Touch	http://jqtouch.com/	Wink	http://www.winktoolkit.org/
jQuery Mobile	http://jquerymobile.com/	Xamarin	http://xamarin.com/
Kendo UI	http://www.telerik.com/kendo-ui	Zepto.js	http://zeptojs.com/

Dentre os muitos *frameworks* disponíveis no mercado, foram escolhidos somente nove deles para comparação. Para escolha desses nove *frameworks*, levou-se em consideração artigos publicados, o mercado e o quão diferentes são. Antes de listar os *frameworks* escolhidos é importante explicar os componentes de um dispositivo mobile com os quais esses *frameworks* podem trabalhar. Esses componentes servem de base para escolha do *framework* que será utilizada para a implementação da aplicação prática.

- **Acelerômetro:** utilizado para detectar a orientação do dispositivo móvel ou a forma com o usuário segura o telefone. Pode ser usado para detectar movimentação em três dimensões.
- **Arquivo:** API para escrita, leitura e navegação pela hierarquia do sistema de arquivo.
- **Banco de Dados:** utilizado para criação de banco de dados no telefone móvel.
- **Bluetooth:** permite comunicação sem fio para a transmissão de dados entre dispositivos, desde que estejam próximos.
- **Calendário:** fornece acesso à funcionalidade calendário nativo do telefone.
- **Câmera:** permite utilizar a câmera para captura de fotos e vídeos.
- **Código de Barras:** API usada para leitura e codificação de códigos de barras através de uma câmera do dispositivo móvel.
- **Contatos:** tem acesso aos contatos do dispositivo.
- **Geolocalização:** é uma API utilizada para fornecer ou acessar a localização geográfica (latitude e longitude) de um dispositivo móvel.
- **Media:** permite acesso as funcionalidades multimídia da plataforma móvel como, reprodução e gravação de áudio e vídeo.
- **NFC:** *Near Field Communication*, permite a troca de informações entre dispositivos sem a necessidade de cabos ou fios, sendo necessária apenas uma aproximação física.
- **Notificação:** notificação visual, auditiva e vibracional.
- **Rede:** fornece acesso para conexão com a Internet para o aplicativo.

A Tabela 3 mostra a comparação dos componentes para os quais os *frameworks* apresentam funcionalidades. A letra “X” indica que o *framework* implementa uma funcionalidade para o componente, o símbolo “-” significa que o *framework* não implementa uma funcionalidade para o componente. Para a aplicação Rota Urbana, a qual será migrada

para a versão multiplataforma, somente serão levados em consideração os seguintes componentes de um dispositivo mobile (devido serem os componentes necessários a aplicação): Geolocalização, Notificação e Rede.

Tabela 3. Comparação de Componentes entre os *Frameworks*

Componente	Apache Flex	Appcelerator	AppGyver	Corona	Marmalade	MobiOne	PhoneGap	Rhodes	Xamarin
Acelerômetro	X	X	X	X	X	X	X	X	X
Arquivo	X	X	X	X	X	X	X	X	X
Banco de Dados	X	X	X	X	X	X	X	X	X
Bluetooth	X	X	X	X	X	X	X	X	X
Calendário	X	X	X	–	–	X	X	X	X
Câmera	X	–	X	X	X	X	X	X	X
Código de Barra	X	–	X	–	–	X	X	X	–
Contatos	X	X	X	X	X	X	X	X	–
GeoLocalização	X	X	X	X	X	X	X	X	X
Media	X	X	X	X	X	X	X	X	X
NFC	X	X	X	X	X	X	X	X	X
Notificação	X	X	X	X	X	X	X	X	X
Rede	X	X	X	X	X	X	X	X	X

A Tabela 3 expõe a capacidade de cada *framework*, a fim de verificação se é possível desenvolver uma aplicação que necessita de componentes específicos.

A Tabela 4 diz respeito a qual plataforma *mobile* o *framework* gera *releases*. A letra “X” indica que o *framework* gera *release* para a plataforma, o símbolo “–” significa que o *framework* não gera *release* para a plataforma.

Tabela 4. Comparação entre os *Frameworks* quanto a Plataforma *Mobile*

<i>FRAMEWORK</i>	ANDROID	iOS	WINDOWS PHONE
Apache Flex	X	X	–
Appcelerator	X	X	–
AppGyver	X	X	–
Corona	X	X	–
Marmalade	X	X	X
MobiOne	X	X	–
PhoneGap	X	X	X
Rhodes	X	X	X
Xamarin	X	X	X

A Tabela 5 lista em qual linguagem de programação são desenvolvidas as aplicações, são: ActionScript, JavaScript, Lua, C++, Ruby e C#, e quase todas os *frameworks* fazem uso de

HTML5 e CSS3. Além disso, as licenças variam de *Open Source*, *Free*, Comercial e algumas tem licenças *Open Source* e Comercial.

Tabela 5. Comparação entre *Frameworks* quanto a Linguagem e o Tipo de Licença

FRAMEWORK	LINGUAGENS	TIPO DE LICENÇA
Apache Flex	ActionScript	Open Source
Appcelerator	JavaScript	Open Source
AppGyver	JavaScript, HTML5 e CSS3	Free
Corona	Lua	Comercial
Marmalade	C++, HTML5 e JavaScript	Comercial
MobiOne	JavaScript, HTML5 e CSS3	OS/Comercial
PhoneGap	JavaScript, HTML5 e CSS3	Open Source
Rhodes	Ruby, JavaScript, HTML5 e CSS	OS/Comercial
Xamarin	C#	Comercial

A Tabela 5 é importante para ilustrar ao desenvolvedor que queira buscar o desenvolvimento multiplataforma com quais linguagens de programação ele deverá ter conhecimento para programar em determinado *framework*. Assim como verificar a licença que é utilizado por ele.

As Tabelas 3, 4 e 5 servem de base para a escolha de qual *framework* ou plataforma de desenvolvimento foi escolhido para implementação da aplicação Rota Urbana. Todos os *frameworks* apresentam funcionalidades para os componentes: Geolocalização, Notificação e Rede, também permitem a utilização da API do Google Maps (necessária para a aplicação).

Como o foco deste trabalho é que sejam gerados *releases* para as Plataformas Android, iOS e Windows Phone, foram somente filtrados para os *frameworks* (Tabela 4): Marmalade, PhoneGap, Rhodes e Xamarin. Desses quatro *frameworks*, três deles apresentam licenças Comerciais (Tabela 5) – Não sendo grande valia ao autor deste trabalho, por questões economia de custos para o desenvolvimento do trabalho –, restando somente o *framework* PhoneGap, é *Open Source*, utiliza as linguagens JavaScript, HTML5 e CSS3 (as quais o autor tem domínio), gera *release* para as três plataformas e tem suporte aos componentes necessários e é possível utilizar a API do Google Maps em conjunto.

2.2 TRABALHOS CORRELATOS

Existem diversos trabalhos relacionados ao desenvolvimento multiplataforma. Estes trabalhos tratam muitas vezes de comparações entre *frameworks*; comparação por meio de

uma aplicação feita com os *frameworks*; outros trabalhos utilizam somente um *framework* e mostram as vantagens que ele traz. Nesta seção serão elencados três trabalhos.

O primeiro trabalho é uma tese de mestrado de [Redda, 2012] intitulada em: “*Cross platform Mobile Applications Development – Mobile Apps Mobility*”. Esta dissertação enfatiza a problemática do desenvolvimento mobile para aplicações nativas e por meio de uma pesquisa, o autor expõe diversas ferramentas (*frameworks* e plataformas) para o desenvolvimento multiplataforma. Também é explorado o conceito de WORA, reduz-se tempo com escrita e manutenção para somente um código, ser capaz de escrever um código e dispor para múltiplos dispositivos e plataformas, faz valer a pena o esforço e descobertas sobre desenvolvimento aplicações *mobile cross-platform* [Redda, 2012].

É apresentada a arquitetura, linguagens de programação, licenças de uso, para quais sistemas operacionais mobile elas geram *releases* e comparações entre essas ferramentas. Sendo realizado um estudo de caso com as plataformas Appcelerator Titanium e Xamarin, com o objetivo de criar uma mesma aplicação, para efeito de comparação entre as plataformas elencadas.

O segundo trabalho, trata-se de um artigo de [Café, 2012] intitulado em: “Desenvolvimento de Cross-Platform Mobile Apps Utilizando o Titanium Mobile”. Este trabalho é semelhante ao anterior, com relação à apresentação da problemática do desenvolvimento para aplicações nativas. “Cada plataforma possui sua *guideline* que diz como os aplicativos devem se parecer e se comportar. Um aplicativo *cross-platform* deve seguir os padrões de cada plataforma, muitas vezes será necessário escrever trechos de códigos específico para cada uma” [Café, 2012].

São relatadas diversas ferramentas para o desenvolvimento multiplataforma, mas é destacado somente uma para migração de uma aplicação em Android para multiplataforma (em Android e iOS), como um estudo de caso.

O último trabalho relatado é um artigo de [Silva et al., 2013] intitulado em: “Desenvolvimento de um Caderno de Campo para Plataformas Móveis utilizando PhoneGap”. Este trabalho ilustra as vantagens do desenvolvimento multiplataforma. Apresenta o PhoneGap como o *framework* para o estudo de caso (migração do “Caderno de Campo Web” para o “Caderno de Campo Mobile”).

Os três trabalhos se assemelham tanto com a conclusão da importância do desenvolvimento mobile multiplataforma, quanto com a implementação de uma aplicação

para comprovação do estudo realizado, cumprindo com seus objetivos. [Silva et al., 2013] complementa:

“Ao longo desse trabalho, pode-se observar que a abordagem de desenvolvimento de aplicativos híbridos é uma boa opção quando se faz é necessário o uso de soluções independentes de plataforma. Neste contexto, o uso de tecnologias web é fundamental para a criação de soluções flexíveis, padronizadas e de fácil aprendizado ao usuário final. Além disso, sua adoção reduz o custo de desenvolvimento, uma vez que os recursos computacionais provêm de softwares livres, e contemplam questões de portabilidade.”

Assim, este trabalho se assemelha aos outros quanto ao destaque da importância de aplicações independentes de plataformas, a comparação entre *frameworks* e plataformas de desenvolvimento e, por apresentar uma aplicação prática por meio da migração para o multiplataforma. Este trabalho distingue-se dos trabalhos correlatos pela exposição dos testes realizados com usuários betas, levantando seus feedbacks, a fim de proporcionar uma aplicação de boa qualidade. Também são expressas as dificuldades enfrentadas durante a confecção deste trabalho, enfatizando a implementação do aplicativo Rota Urbana Multiplataforma.

3 APLICAÇÃO PRÁTICA

Nas subseções seguintes serão apresentadas as tecnologias utilizadas para o desenvolvimento da aplicação prática, sobre o aplicativo para solução proposta, os testes que foram realizados e os resultados obtidos.

3.1 TECNOLOGIAS UTILIZADAS

Nesta seção serão listadas as ferramentas e tecnologias utilizadas para o desenvolvimento da aplicação.

3.1.1 Intel XDK

O Intel XDK é um ambiente de desenvolvimento multiplataforma para aplicações Android, iOS, Windows Phone 8, criada pela Intel. Utiliza as tecnologias HTML5, CSS, JavaScript e *backend* Node-Webkit (proporciona a criação de aplicações nativas com as tecnologias web). O Intel XDK É compatível com os sistemas operacionais, Windows, OS X, Linux Ubuntu [INTEL XDK, 2015].

O Intel XDK (versão 2366) traz consigo jQuery Mobile, App Framework, Bootstrap, Topcoat, Ratchet, Ionic e o Framework 7, utilizados para estruturação de páginas e navegação entre páginas. Tem suporte para a API do PhoneGap sendo possível a inclusão de *plug-ins* de terceiros.

Para testes, a ferramenta tem emuladores que simulam diversas plataformas mobile e também um aplicativo, App Preview, para testes no dispositivo (disponível para Android, iOS, Windows Phone). Assim como é possível fazer a depuração do aplicativo na ferramenta.

“O processo de compilação empacota o *app* em um formato que pode ser submetido a uma loja de *app* móvel e, finalmente, baixado e instalado no dispositivo móvel do cliente” [Karadimce, e Bogatinoska, 2014].

3.1.2 PhoneGap

O PhoneGap é uma distribuição do Apache Cordova. Foi criado em 2009 pela Nitobi. Em 2011, a Adobe comprou a Nitobi, e os direitos sobre a marca PhoneGap. É um *framework open source* para criação de aplicações mobile nativas que utiliza HTML5, CSS3 e JavaScript. O PhoneGap suporta diversas plataformas: Android, iOS, Windows Phone, BlackBerry, webOS, Symbian e Bada [PHONEGAP, 2015].

Com o PhoneGap é possível desenvolver uma aplicação com as tecnologias web de forma abstrata e gerar *builds* para as diversas plataformas, contudo, não existe implementação de acesso a determinados componentes de determinada plataforma. O PhoneGap fornece acesso aos seguintes componentes do dispositivo móvel listados na Tabela 3.

[Myer, 2012] complementa: “você estará usando tecnologias web que você já entende... Isso também significa que você deve ser capaz de portar seu código web para vários dispositivos diferentes, com pouca ou nenhuma alteração”.

Abaixo, na Figura 2, segue a arquitetura do PhoneGap. Segundo [Popa, 2013], a camada WebApp utiliza as tecnologias web (JavaScript, HTML e CSS); a camada PhoneGap *Plug-ins* permitem o acesso aos componentes do dispositivos móvel; e o *HTML Rendering Engine* usa o *webview* do sistema operacional *mobile* nativo para a renderização do layout. A aplicação é distribuída em formato binário, após um processo de empacotamento de acordo com a política dos proprietários do sistema operacional móvel e pode ser vendido pelos canais de distribuição dos fornecedores de aplicativos móveis.

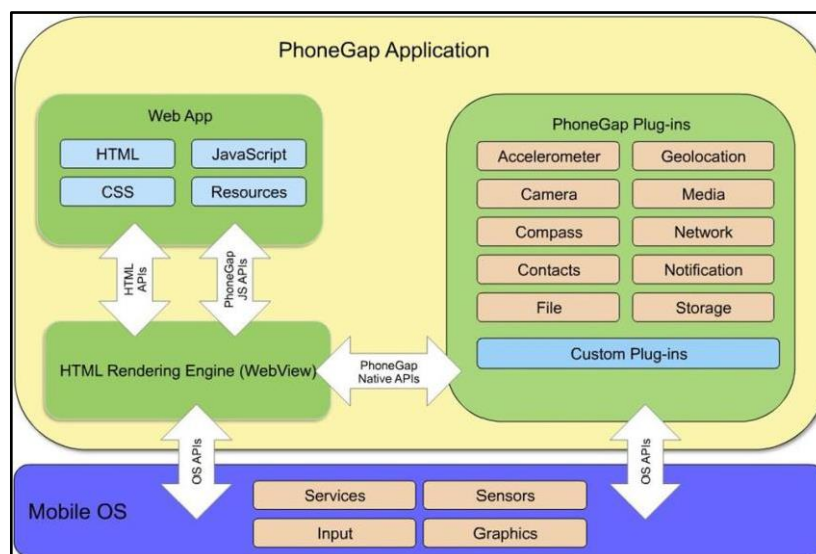


Figura 2. Arquitetura do Framework PhoneGap [Popa, 2013]

3.1.3 jQuery Mobile

O jQuery Mobile é um *framework* de interface gráfica para dispositivos móveis. Baseado em HTML5 e CSS3, e também faz uso do JavaScript para definição da estrutura de uma página e a navegação entre páginas. Atualmente está na versão 1.4.5. Tem compatibilidade com os navegadores desktop (jQuery UI), iPad, iPhone, Android, Blackberry, Windows Phone, Firefox Mobile, entre outros. Fornece um conjunto de *widgets* pré-configurados: botões, barra de ferramentas, busca, *list views*, abas, *sliders*, etc.

Segundo a definição de [Firtman, 2012]:

“O jQuery Mobile é um framework que proporciona a experiência de webapps para dispositivos móveis e tablets, principalmente com interfaces de toque, sem esforço, através de múltiplas plataformas, e usando apenas código padrão HTML5. Este framework utiliza o núcleo do JQuery UI, uma biblioteca JavaScript, CSS3 e algumas imagens de recurso.”

3.1.4 HyperText Markup Language

“*HyperText Markup Language* (HTML), é a linguagem de marcação para páginas *web*” [W3C, 2015b]. “Os documentos HTML são descritos por *tags* HTML” [W3SCHOOLS, 2015] (< e >), “cada *tag* descreve o conteúdo do documento diferente” [W3SCHOOLS, 2015], as quais correspondem um item da sintaxe. É responsável pela exposição do conteúdo, sua estruturação e pela semântica.

Atualmente o HTML está na versão 5, HTML5. Tem como objetivo a manipulação dos elementos, dando liberdade ao desenvolvedor para com a modificação destes, e que disponha ao usuário uma melhor interface ao usuário final. O HTML5 dispõe de recursos que facilitam a manipulação de CSS e JavaScript, e também traz: Canvas (permite o desenho elementos gráficos usando JavaScript), Local Storage (armazenamento de dados localmente no navegador do usuário), Geolocalização e WebGL (Baseado no OpenGL, fornece suporte para renderização de gráficos 2D e 3D sem a necessidade de plug-ins no navegador).

3.1.5 *Cascading Style Sheets*

“*Cascading Style Sheets* (CSS) é um mecanismo simples para adicionar estilo (por exemplo, fontes, cores, espaçamento) a documentos web” [W3C, 2015a]. Possui seletores que são associados às *tags* do HTML. São eles: o pré-seletor (exemplo: *div*, *p*, *a* e etc), o *id* (identificador único que pode ser adicionado a determinado elemento HTML) e *class* (pode ser definido para mais de um elemento HTML). O CSS3 é uma evolução das versões antigas, contém novas propriedades para uma melhor visualização do conteúdo web.

3.1.6 JavaScript

O JavaScript foi criado em 1995, é uma linguagem de programação que pode ser adicionada a páginas HTML. É uma linguagem orientada a objetos, fracamente tipada e dinâmica. Antigamente o JavaScript tinha uma má reputação, por causa do baixo desempenho, de *bugs* que surgiam sem resolução aparente, e de dependências com o *Document Object Model* (DOM – representa como as marcações em HTML, XHTML e XML são organizadas e lidas pelo navegador).

Com scripts JavaScript é possível inspecionar os valores digitados em um formulário para validação, ocultar elementos em uma página dependendo do clique do usuário e é possível também realizar consultas ao servidor de banco de dados sem atualizar a página da web atual [Stark, 2010].

Com o surgimento do *Asynchronous Javascript and XML* (AJAX – conjunto de tecnologias, a qual o desenvolvedor deve conhecer bem JavaScript/DOM, CSS, XML e XMLHttpRequest Object, usado para troca de dados com o servidor) o JavaScript passou a ser visto novos olhares devido a má reputação que tinha. Com isso, diversas APIs foram desenvolvidas, entre elas: JQuery, Noje.js, ExtJS e GWT.

3.2 ROTA URBANA MULTIPLATAFORMA

A aplicação da solução proposta visa o uso da versão multiplataforma do Rota Urbana⁶ para usuários com diferentes *smartphones* com sistemas operacionais Android, iOS e Windows Phone, esta seção trata justamente desta aplicação. Primeiramente serão apresentados os requisitos e funcionalidades (subseção 3.2.1), posteriormente será exposta a arquitetura por meio de um diagrama de componentes (subseção 3.2.2), e por fim a interface da aplicação (subseção 3.2.3).

3.2.1 Requisitos e Funcionalidades

Para a criação da Aplicação Rota Urbana Multiplataforma foi realizada a engenharia reversa. Este processo foi realizado com base nas versões Web e Android: por meio da criação de Casos de Uso, elicitação de requisitos e reutilização do código fonte.

O Rota Urbana é uma plataforma de criação e compartilhamento de rotas de regiões metropolitanas para ajudar os cidadãos a se locomoverem pela cidade e região. Faz uso do conceito de *Crowdsourcing*⁷, as rotas podem ser registradas por qualquer usuário que queira colaborar com informações para auxiliar a mobilidade urbana de sua comunidade. [ROTA URBANA, 2013].

Atualmente o Rota Urbana exibe as rotas de ônibus da cidade de Belém e Macapá, e com ele é possível saber quais linhas de ônibus levarão o usuário ao seu destino, indicando a parada de subida no ônibus e parada de descida. [ROTA URBANA, 2013]. A Figura 3 apresenta a tela do sistema Web. Na Figura 4 está a tela da aplicação Android.

⁶ Seguem os endereços eletrônicos para download do aplicativo na AppStore: <https://itunes.apple.com/us/app/rota-urbana/id1021111389?l=pt&ls=1&mt=8>, e na Loja: <http://windowsphone.com/s?appid=97d0d046-b981-4bf4-82e1-a92302466b5b>.

⁷ *Crowdsourcing* é um modelo de negócio, baseado na Internet, no qual as empresas expõem um problema específico e com a contribuição de indivíduos externos a empresa são propostas soluções para o problema. [MUNHOZ et al, 2013].

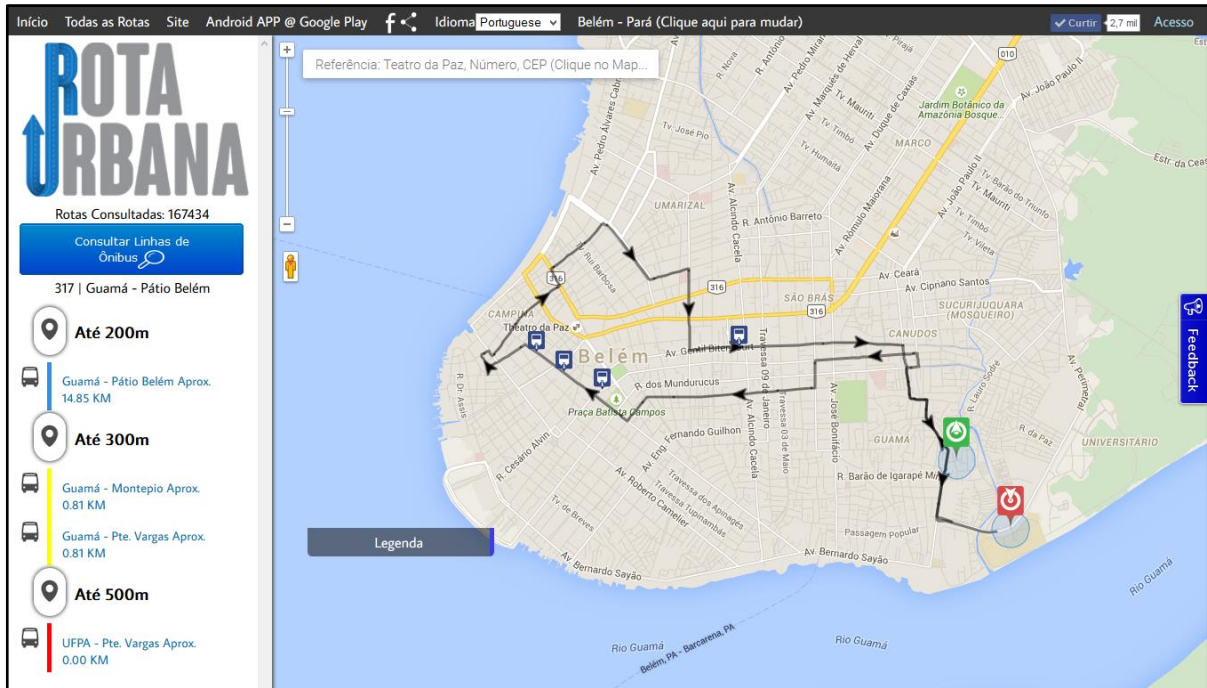


Figura 3. Rota Urbana (Web) – Fonte: ROTA URBANA 2015

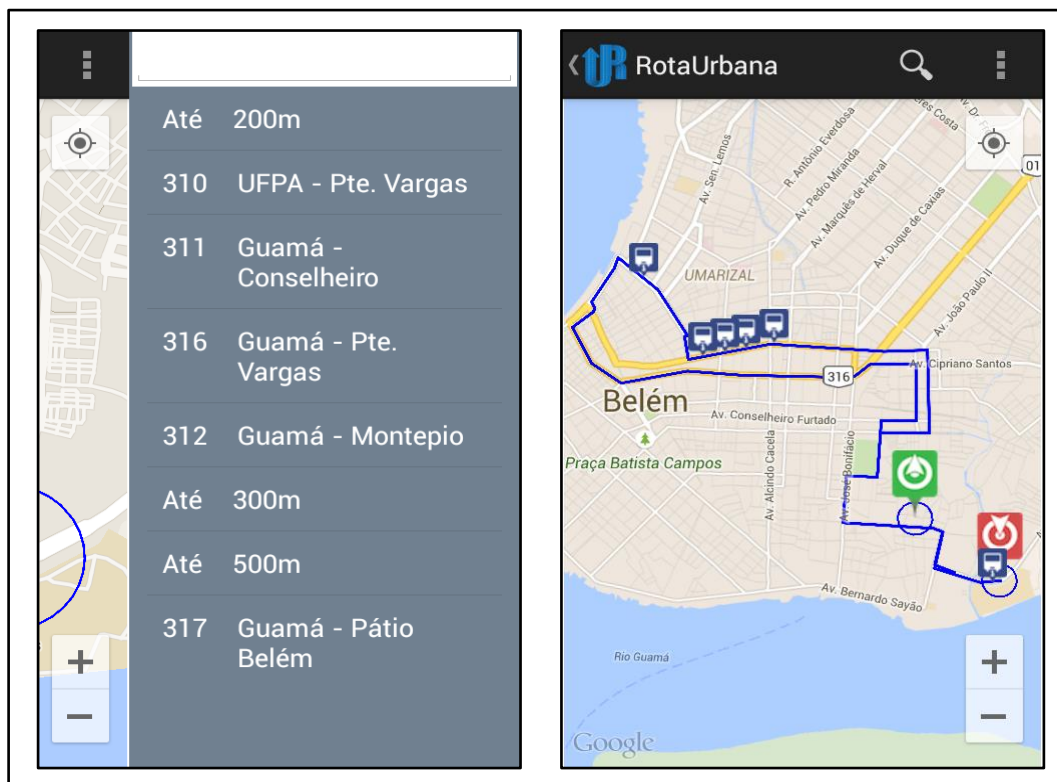


Figura 4. Rota Urbana (Android) – Fonte: ROTA URBANA 2015

Para implementação da aplicação são necessários os requisitos, estes que foram obtidos com base na utilização das versões Web e Android do Rota Urbana. Na Tabela 6 estão elicitados os requisitos da aplicação multiplataforma.

Tabela 6. Requisitos do Rota Urbana Multiplataforma

REQUISITOS FUNCIONAIS		
Nº	Descrição	Prioridade
RF 1	O aplicativo deve permitir a pesquisa de rotas de ônibus.	Alta
RF 2	O aplicativo deve exibir cada rota pesquisada com os seguintes campos: código do ônibus, o nome e a distância aproximada da rota.	Alta
RF 3	O aplicativo deve permitir a visualização de linhas (indicando o sentido da rota do ônibus) e paradas de ônibus que estão na linha exibida. Sendo a visualização de uma rota por vez.	Alta
RF 4	O aplicativo deve permitir a adição de marcadores inicial e destino no mapa, para a pesquisa de rotas de ônibus.	Alta
RF 5	O aplicativo deve exibir a geolocalização do usuário no mapa, quando disponível.	Média
RF 6	O aplicativo deve permitir a limpeza de marcadores inicial e destino, e linha de ônibus, e as paradas de ônibus do mapa.	Baixa
RF 7	O aplicativo deve exibir a legenda dos ícones no mapa: Posição Inicial, Posição Destino, Parada de Ônibus, Centralização com Posição do GPS, Posição do GPS e Limpar Mapa.	Baixa
RF 8	O aplicativo deve informar que a pesquisa está sendo realizada.	Alta
RF 9	O aplicativo deve informar quando a rota de ônibus escolhida para visualização esta sendo carregada no mapa.	Alta
RF 10	O aplicativo deve permitir a pesquisa de locais no mapa, onde o usuário informa o nome do local desejado.	Media

Na Figura 5 está o diagrama de casos de uso do Rota Urbana (versão web, apresenta mais funcionalidades). Para a migração multiplataforma foi implementado somente dois casos de uso (em destaque na Figura 5): Pesquisar Rotas e Visualizar Rotas. Estes casos de uso foram escolhidos por adequarem-se melhor ao desenvolvimento mobile e ao tempo para finalização da aplicação.

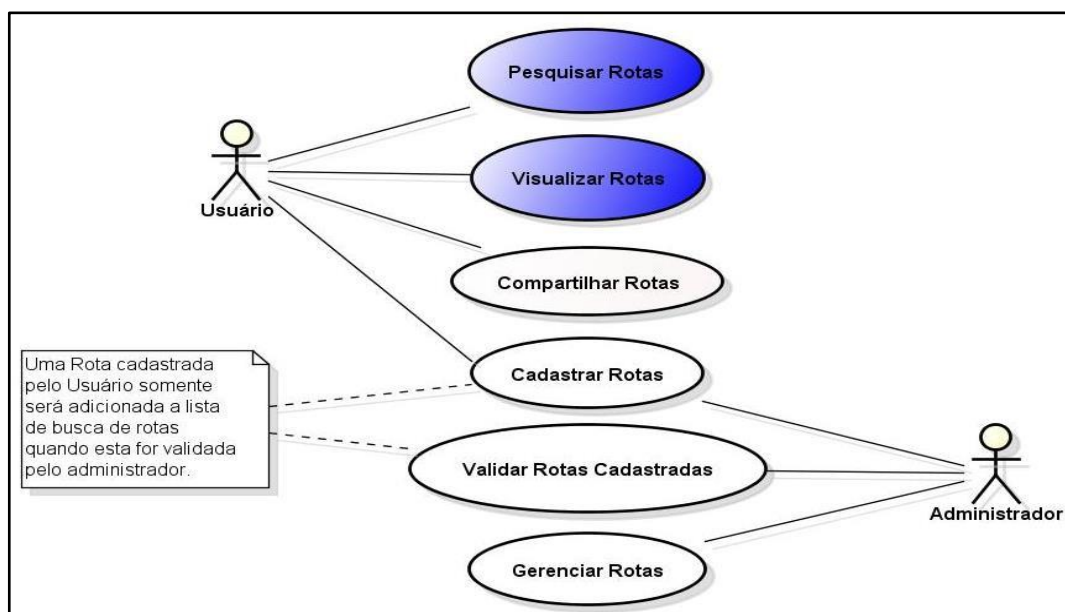


Figura 5. Caso de Uso do Rota Urbana

O caso de uso Pesquisar Rotas representa a ação que o usuário irá realizar para encontrar rotas que passam pelos pontos inicial e destino, definidos pelo usuário. O caso tem início quando o usuário define com o clique no mapa os pontos: inicial e destino. Após a definição dos marcadores o usuário deve clicar no botão com o ícone em formato de lupa para pesquisar uma rota. Após a pesquisa é exibido uma lista das rotas que foram encontradas. Cada rota listada apresentará o código do ônibus, o nome e a distância aproximada da rota.

O caso de uso Visualizar Rotas é a ação posterior à pesquisa, com as rotas listadas o usuário poderá visualizá-las. Nas rotas listadas o usuário deve clicar em uma rota para visualizá-la. No mapa serão exibidos além dos marcadores de início e destino, a linha do ônibus (indicando o sentido da rota do ônibus) e as paradas de ônibus que estão na linha exibida. Para a visualização de outras rotas o usuário deve clicar novamente na rota desejada, que o mapa será atualizado com a nova rota indicada.

3.2.2 Arquitetura da Solução

Para apresentação da arquitetura da aplicação será exposto um diagrama de componentes. Este diagrama ilustra a estrutura física do software, seus relacionamentos e implementação, representa um empacotamento físico de elementos relacionados logicamente. Apresenta as interfaces que os componentes dispõem e quais componentes utilizam essas interfaces. As interfaces facilitam a comunicação entre componentes por meio de pontos de acesso. Abaixo, na Figura 6, está o diagrama de componentes do sistema.

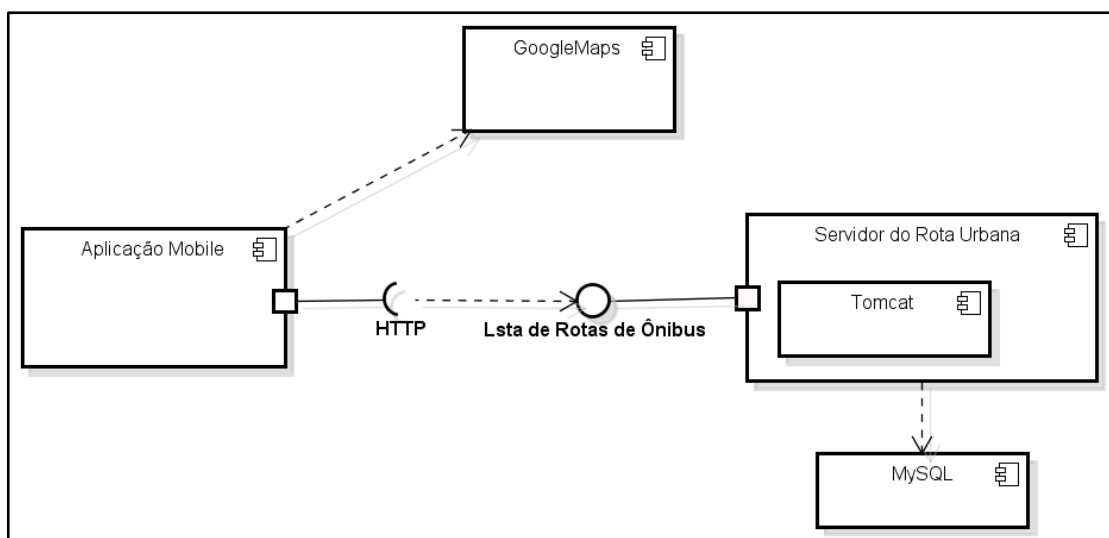


Figura 6. Diagrama de Componentes

Referente à Figura 6, o sistema cliente-servidor está organizado através de 4 componentes principais, os quais são: a Aplicação Mobile; o Servidor do Rota Urbana que contém a interface para fornecer a lista de rotas de ônibus consultadas pela aplicação; o Banco de Dados (MySQL) que armazena os dados do sistema; e por fim o Google Maps, utilizado para interação do usuário com o mapa e visualização de rotas de ônibus. A comunicação se dá por portas com o protocolo HTTP que estão representados no diagrama.

3.2.3 Interface da Aplicação

Nesta subseção será apresentada a interface desenvolvida da aplicação Rota Urbana Multiplataforma. Utilizando as ferramentas descritas na seção 3.1, e o código fonte de ambas as versões do Rota Urbana.

A Figura 7 mostra a tela inicial da aplicação. Esta tela é composta pelo mapa (API Google Maps); que apresenta um botão para limpeza do mapa (retirar marcadores e a rota desenhada no mapa); outro para centralização da localização do usuário com o GPS (o marcador vermelho exibe esta localização); e um botão para zoom. Além de um campo para pesquisa de lugares na Cidade, visto que muitas vezes o usuário desconhece certas localidades de Belém. Existem mais três botões no cabeçalho da tela, o primeiro (da esquerda para direita) é para exibir o Menu, o segundo para pesquisa de Rotas (o usuário deve definir marcadores de início e destino), e o último mostra o *side bar* direito que exibe a listagem de rotas pesquisadas.

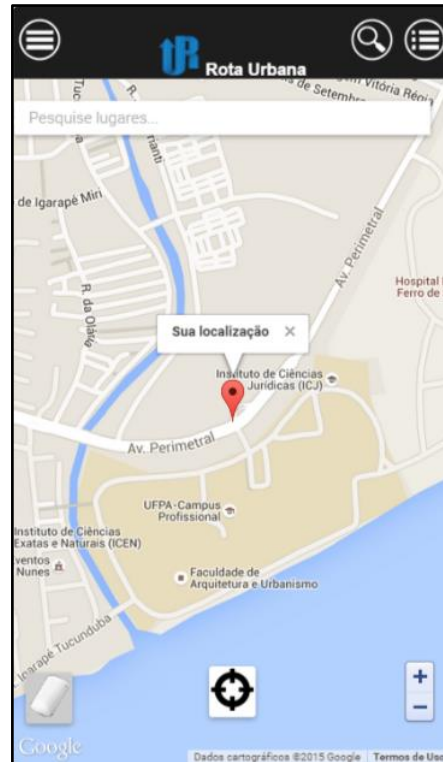


Figura 7. Tela Inicial do Rota Urbana Multiplataforma

Abaixo segue a Figura 8 com o Menu da aplicação: Botão “Home” (para recolher o Menu); Botão “Legenda do Mapa”; os botões, “Sobre o Sistema”, “Quem Somos” e “Termos” exibem informações adicionais sobre a aplicação; o botão “Sair”, para sair da aplicação; e o botão “Feedback”, para auxiliar no melhoramento da aplicação.

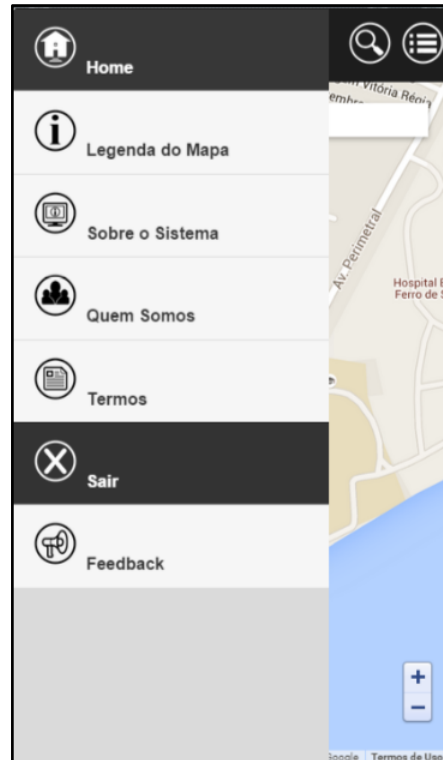


Figura 8. Menu do Rota Urbana Multiplataforma

A Figura 9 mostra o mapa com os marcadores de início (marcador verde) e destino (marcador vermelho), para representar uma busca de rotas do usuário. O círculo azul indica o raio para pesquisa de rotas.

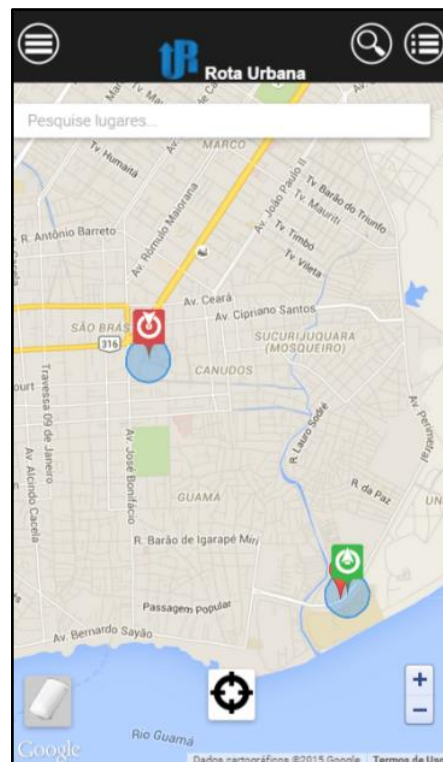


Figura 9. Marcadores de Início e Fim para Pesquisa de Rotas de Ônibus

Abaixo na Figura 10 está ilustrada a tela com a listagem de rotas de ônibus encontradas. As rotas são listadas de acordo com três parâmetros: “até 200m” (rotas que estão aproximadamente 200 metros de distancia do usuário), “até 300m” (rotas que estão aproximadamente 300 metros de distancia do usuário) e “até 500m” (rotas que estão aproximadamente 500 metros de distância do usuário). Cada rota listada apresenta os seguintes campos: Código do Ônibus, Nome do Ônibus, e Distância Aproximada da Rota do Ônibus.

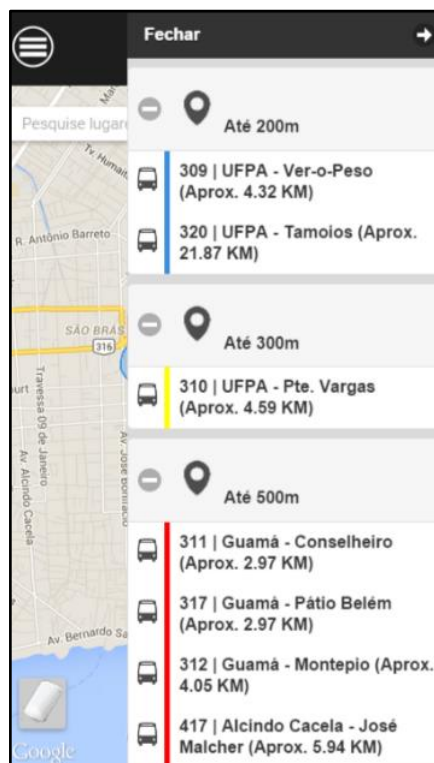


Figura 10. Lista de Rotas de Ônibus Encontradas

Quando o usuário escolhe uma rota de “até 200m” o raio dos marcadores de início e destino ficam de cor azul, quando é uma rota de “até 300m” o raio fica amarelo, e em uma rota de “até 500m” o raio fica vermelho. A Figura 11 ilustra como fica a rota de “até 500m” é desenhada no mapa. Os marcadores de cor azul representam as paradas de ônibus cadastradas no sistema. Como recurso do zoom, o usuário pode ter uma visualização melhor desta rota, e pode visualizar todas as rotas listadas (uma rota por vez) e pesquisar outras também.

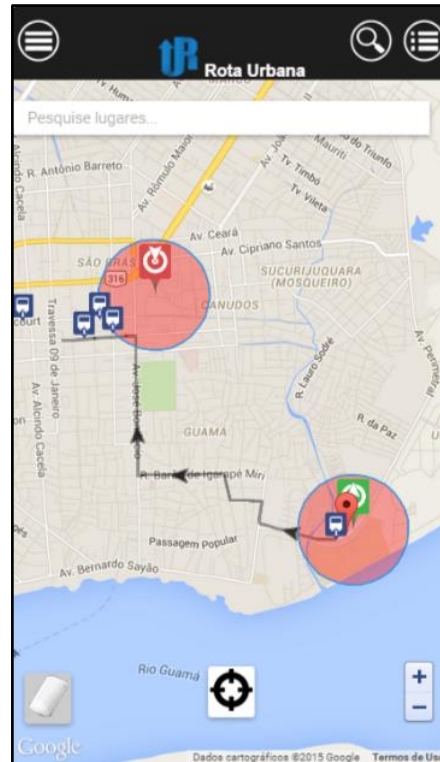


Figura 11. Tela com a Rota Escolhida desenhada no Mapa

3.3 TESTES REALIZADOS

Teste é o processo pelo qual se experimenta o software com a intenção de encontrar e corrigir erros [Pressman, 2011]. É uma etapa fundamental no processo de desenvolvimento, sendo uma das causas de aplicações e sistemas defeituosos e com pouca qualidade.

Como o foco deste trabalho deu-se no desenvolvimento multiplataforma (com uso de tecnologias web), foi utilizado técnicas de teste para *webapps*. Não foram empregados testes automatizados, pois não foram encontradas ferramentas para testes de interação no mapa da aplicação. Portanto, foi necessário realizar os testes diretamente por um usuário interagindo com o mapa.

Inicialmente foi realizado o teste de unidade. A unidade é o componente do software que não pode ser dividido (variáveis e funções). Para testar as unidades foram realizados testes por meio da interface da unidade. Onde foram verificados se as funções respondem às expectativas de sucesso testando os mecanismos da interface (referente aos requisitos da aplicação, a navegação entre paginas, entre outras). Verificou-se também a usabilidade da

aplicação, considerando a interatividade, a disposição e tamanho dos botões para diferentes plataformas e a *Affordance* para realização de ações. Após a integração das unidades da aplicação, realizou-se o teste de compatibilidade. A ferramenta Intel XDK apresenta um emulador que simula diferentes plataformas mobile, e também permite a depuração em dispositivos moveis (segue ilustração na Figura 12).

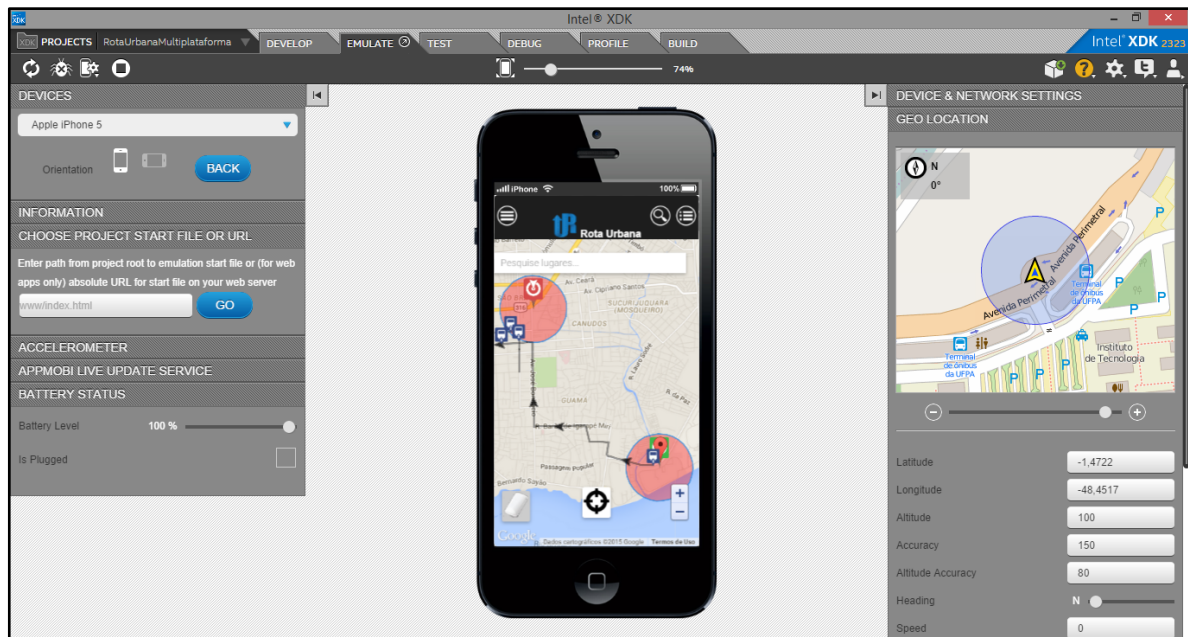


Figura 12. Emulador do Intel XDK

É difícil prever como o usuário final irá usar a aplicação e descobrir todos os tipos de erros possíveis, para tornar a aplicação livre de erros. É custoso executar testes de aceitação para cada cliente (a fim de validar os requisitos), como alternativa é comum utilizar os testes alfa e beta [Pressman, 2011].

O teste alfa é executado em um ambiente controlado pelo desenvolvedor. Em relação a este trabalho, o aplicativo Rota Urbana Multiplataforma foi executado em diferentes dispositivos móveis para as plataformas Android, iOS e Windows Phone. Houve problema de incompatibilidade com o Windows Phone (principalmente em dispositivos inferiores), pois a aplicação apresentou baixo desempenho na interação com o mapa do Google Maps. Não foi utilizado o Bing Maps (recomendável para a plataforma Windows Phone), pois o cadastro de linhas de ônibus foi realizado no Google Maps, o qual apresenta um melhor mapeamento das cidades, tornando a aplicação dependente desse sistema.

O teste beta é executado em um ambiente não controlado, nas instalações de um ou mais usuários finais. Em nível de teste beta, a aplicação foi publicada nas lojas de aplicativos a fim de obtenção de *feedback* dos usuários. No menu do aplicativo tem-se um botão que

serve para obtenção de *feedback* dos usuários, o qual é um formulário com perguntas avaliando a satisfação dos usuários no uso da aplicação e aplicações multiplataforma. Na sessão seguinte serão apresentados os resultados obtidos deste questionário.

3.4 RESULTADOS

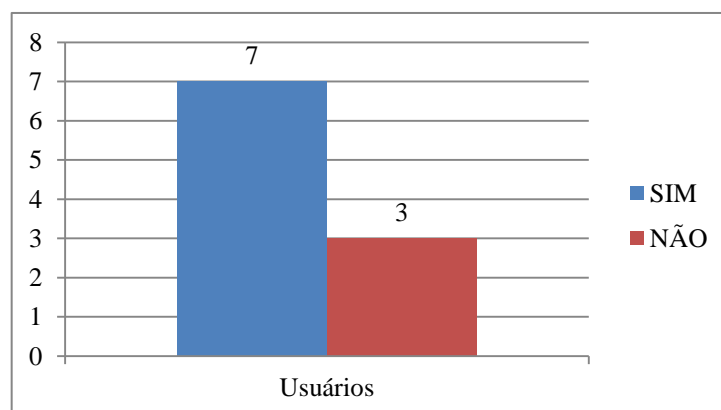
Na sessão anterior foram descritos os processos para os testes alfa e beta. Pela aplicação era possível o envio de *feedback* do usuário que será utilizado para validação dos requisitos. No Apêndice I, estão as perguntas do formulário. Por um período de três semanas obtiveram-se respostas de dez usuários (seguem as respostas no Apêndice II, nas Tabelas 7 e 8). Houveram questões de múltipla escolha e discursivas, as quais visam listar quais erros foram encontrados e qual o entendimento sobre aplicações multiplataforma.

Os 10 *feedbacks* obtidos foram suficientes para avaliação do aplicativo. Pois, Segundo um estudo de [Nielsen, 2000], cinco usuários são necessários para testar uma aplicação. Pois:

Conforme você adicione mais e mais usuários, você aprende cada vez menos, porque você vai continuar vendo as mesmas coisas repetidas vezes... Após o quinto usuário, você está desperdiçando seu tempo observando os mesmo resultados repetidamente, e não aprendendo nada novo.

A primeira pergunta é se os usuários conseguiram fazer pesquisas de rotas na aplicação. De 10 usuários 7 responderam “SIM” e 3 responderam “NÃO”. No Gráfico 1 abaixo segue a ilustração.

Gráfico 1. Resultado da Primeira Pergunta do Formulário



A segunda pergunta é relativa a dificuldades enfrentadas no uso da aplicação (seis usuários tiveram dificuldades). Problemas com o Windows Phone sobre a fluidez do mapa

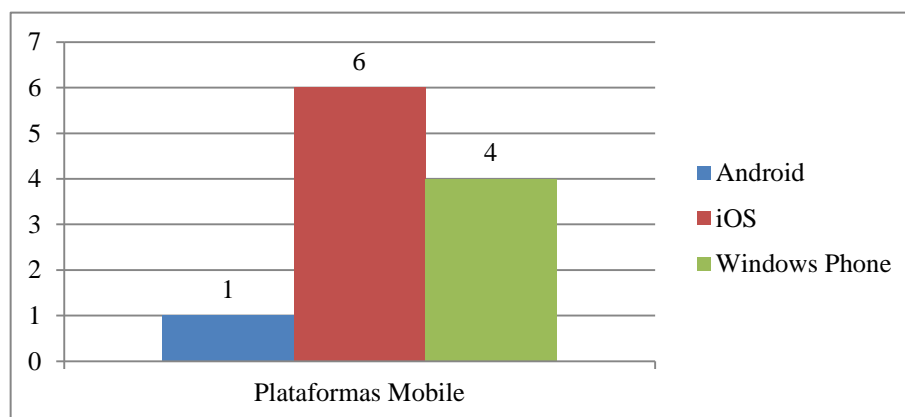
eram esperados, o que foi confirmado pelos *feedbacks*. Também se verificou problemas quanto a lentidão da barra de rolagem quando listava varias rotas, com a barra de menu do iOS que estava sobre a aplicação, dificuldades em entender certos ícones, e em não saber como usar o aplicativo. A aplicação migrada para o multiplataforma foi baseada nas versões web e Android existentes, visto que houveram dificuldades em usar a aplicação e em entender o sentido dos botões, deduziu-se que são novos usuários, é sugerido à postagem de um vídeo tutorial mostrando como utilizar o aplicativo.

A terceira pergunta é quanto à aplicação de nota para o aplicativo, o qual obteve média de 3,1 pontos em uma escala de 1(ruim) a 5 (excelente).

A quarta pergunta refere-se ao motivo da nota aplicada. Os problemas relatados nas dificuldades encontradas foram uns dos motivos para a aplicação da nota pelos usuários, onde as notas variaram de 1 a 3. Para as notas 4 e 5 obtidas, foi levado em consideração a utilidade do aplicativo e atendimento da requisição do usuário.

A quinta pergunta é sobre quais plataformas móveis os usuários possuem. A quantidade usuários com Android foi 1, para iOS foram 6 e Windows Phone foram 4 (tiveram usuários com mais de uma plataforma). Para melhor exposição do resultado segue representação no Gráfico 2.

Gráfico 2. Resultado da Quinta Pergunta do Formulário



A sexta e última pergunta refere-se à importância de aplicações que estejam disponíveis para mais de uma plataforma. Com exceção de um usuário que não respondeu esta pergunta, o restante destacou a importância de aplicações multiplataforma: levando em consideração a abrangência e acessibilidade a mais de um grupo de usuários no mercado.

4 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi apresentado como aplicação prática o desenvolvimento multiplataforma de uma aplicação com uma solução para mobilidade urbana, Rota Urbana. Foi realizado um levantamento bibliográfico e comparativo entre *frameworks*, a fim de encontrar o que melhor se adequa as especificações da solução proposta. Entre eles o PhoneGap foi selecionado em conjunto com o Intel XDK (IDE para desenvolvimento da aplicação). Avaliou-se a aplicação em diferentes dispositivos móveis (Android, iOS e Windows Phone), a qual se configurou nos testes alfa e beta.

Em relação aos resultados obtidos do formulário repassado aos usuários beta, a adequar as reclamações feitas na aplicação. Contudo, no ambiente controlado (teste alfa) a aplicação demonstrou lentidão no Windows Phone com dispositivos inferiores. Durante esse processo houve dificuldades, são elas: dificuldades de desenvolvimento e dificuldades de implantação.

As dificuldades de desenvolvimento refletem-se no período de estudo das linguagens, JavaScript, HTML e CSS com o foco no desenvolvimento *mobile* multiplataforma. Cada plataforma tem sua particularidade: formas de interação com aplicativos, disposição dos componentes no layout, dependendo da funcionalidade que se deseja implementar, certos componentes ou mesmo o framework para interface com o usuário, podem ser inúteis para determinadas plataformas ou dispositivos mobile.

O desenvolvimento do aplicativo foi realizado com as ferramentas (expostas na subseção 3.1), com o auxílio dos emuladores do Intel XDK verificou-se como a aplicação poderia se comportar em um dispositivo real. No entanto é necessário testar em um dispositivo real.

Para implantação na plataforma Android, não se obtiveram dificuldades, sendo gerados *releases* sem problemas. Para implantação nas plataformas Windows Phone e iOS foram encontradas barreiras. Foi necessário criar uma conta para publicação do aplicativo e adicionar os identificadores da conta no arquivo de configuração. No Windows Phone precisou-se fazer a publicação na loja virtual, para que os usuários betas pudessem utilizar a aplicação.

Na plataforma iOS, gerou-se os certificados necessários para o construção do *release*, mas no processo de *build* do aplicativo no Intel XDK exibia erro com os certificados adicionados, sendo que eles estavam em ordem. Muito tempo foi perdido na resolução desse problema, a solução encontrada foi em utilizar o Adobe Phonegap Build⁸, não havendo problemas com os certificados para iOS. O projeto teve que ser modificado tanto na estrutura quanto em retirar as dependências das bibliotecas do Intel XDK para a submissão nesse *build*.

É estimulante codificar uma aplicação uma vez e poder executá-la independentemente da plataforma. Todavia, é importante verificar como a aplicação irá se comportar para cada plataforma, não somente quanto ao sistema operacional como também a robustez do dispositivo.

As empresas criam ou buscam cada vez mais soluções para smartphones objetivando o lucro e liderar a concorrência. Logicamente, quanto mais usuários a empresa alcançar mais próximo do sucesso ela estará, e “aplicativos multiplataforma deixaram de ser um mero capricho e passaram a ser a solução dos problemas destas empresas” [Café, 2012]. E [Palmieri, Singh e Cicchetti, 2012] complementam:

No futuro, cross platform tools podem trazer uma mudança drástica no modelo de negócios de sistemas operacionais móveis, especialmente devido ao fato da redução dos custos de desenvolvimento de novas aplicações.

Como trabalhos futuros propostos, recomenda-se o desenvolvimento de um aplicativo no ambiente multiplataforma, nativo no Android, no iOS e no Windows Phone, com o objetivo de avaliar o desempenho deste. Também se propõe a implementação de um aplicativo com pelo menos dois *frameworks* e realizar uma comparação entre a aplicação desenvolvida com eles para gerar uma matriz comparativa mais aprofundada.

⁸ Adobe Phonegap Build: utiliza serviços da nuvem para compilar aplicações PhoneGap. Segue o endereço eletrônico: <https://build.phonegap.com/>.

REFERÊNCIAS

- ALLEN, S.; GRAUPERA, V.; LUNDRIGAN, L. *Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution*. New York: Apress, 2010.
- AMADEO, R. *How Google Inbox shares 70% of its code across Android, iOS, and the Web*. 2014. Disponível em: <<http://arstechnica.com/information-technology/2014/11/how-google-inbox-shares-70-of-its-code-across-android-ios-and-the-web/>>. Acesso em: 11 fev. 2014.
- CAFÉ, A. **Desenvolvimento de Cross-Platform Mobile Apps Utilizando o Titanium Mobile**. 2012. Disponível em: <<http://adrielcafe.com/images/adrielcafe/artigos/2012/12/tcc/adriel-tcc.pdf>>. Acesso em: 25 ago. 2014. Monografia – Faculdade Zacarias de Góes, Bahia, 2012.
- DESISTO, R. et al. *Predicts 2015: Mobile Apps and Development*. 2014. Disponível em: <<https://www.gartner.com/doc/2926322/predicts--mobile-apps-development>>. Acesso em: 21 jul. 2015.
- DINCER, A.; URAZ, B. **Google Maps JavaScript API Cookbook**. Birmingham: Packt Publishing, 2013.
- DUBROVSKÝ, J. *Cross Platform Mobile Development*. In: *10th Summer School of Applied Informatics*, Masarykova, p 1-6, Janeiro de 2014.
- FALK, M.; BADE, S. *Mobile Frameworks Comparison Chart*. Disponível em: <<http://mobile-frameworks-comparison-chart.com/>>. Acesso em: 20 fev. 2015.
- FIRTMAN, M. **jQuery Mobile: up and running**. Sebastopol: O'Reilly, 2012.
- GIFFORD, M. **PhoneGap Mobile Application Development Cookbook**. Birmingham: Packt Publishing, 2012.
- HAMANN, R. **iOS, Android e Windows Phone: números dos gigantes comparados [infográfico]**. 2014. Disponível em: <<http://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm>>. Acesso em: 29 mai. 2015.
- HARTMANN, G.; STEAD, G.; DEGANI, A. *Cross-platform mobile development*. 2011. Disponível em: <<https://wss.apan.org/jko/mole/Shared%20Documents/Cross-Platform%20Mobile%20Development.pdf>>. Acesso em: 19 ago. 2014.
- INTEL XDK. **Intel XDK**. 2015. Disponível em: <<https://software.intel.com/en-us/intel-xdk>>. Acesso em: 30 set. 2014.
- JONES, N. *Top 10 Mobile Technologies and Capabilities for 2015 and 2016*. 2014. Disponível em: <<https://www.gartner.com/doc/2665315/top--mobile-technologies-capabilities>>. Acesso em: 21 jul. 2015.
- JQUERY MOBILE. **jQuery Mobile**. 2015. Disponível em: <<http://jquerymobile.com/>>. Acesso em: 30 set. 2014.

KAPETANAKIS, M. *Developer Economics 2012 – The new app economy*. London: VisionMobile, 2012. Disponível em: <<http://www.visionmobile.com/blog/2012/06/report-developer-economics-2012-the-new-app-economy/>>. Acesso em: 21 ago. 2014.

KARADIMCE, A.; BOGATINOSKA, D. *Using hybrid mobile applications for adaptive multimedia content delivery*. In: Information and Communication Technology, Electronics and Microelectronics (MIPRO), *37th International Convention on*, Opatija, Maio de 2014. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6859654>. Acesso em: 30 set. 2015.

LOUIS. T. *How Much Do Average Apps Make?*. 2013. Disponível em: <<http://www.forbes.com/sites/tristanlouis/2013/08/10/how-much-do-average-apps-make/>>. Acesso em: 29 mai. 2015.

MAOZ, M.; DESISTO, R. *The Future of Enterprise Applications Is Mobility*. 2014. Disponível em: <<https://www.gartner.com/doc/2793917/future-enterprise-applications-mobility>>. Acesso em: 21 jul. 2015.

MUNHOZ, A. et al. *Coworking e Crowdsourcing: Como Modelos de Negócios Inovadores Influenciam no Desenvolvimento de Start-Ups*. SEMINÁRIOS EM ADMINISTRAÇÃO, 2013.

MYER, T. *Beginning PhoneGap*. Indianapolis: John Wiley & Sons, Inc., 2012.

NEWMAN, B. *Are Cross-Platform Mobile App Frameworks Right for your Business?* 2011. Disponível em: <<http://mashable.com/2011/03/21/cross-platform-mobile-frameworks/>>. Acesso em: 22 ago. 2014.

NIELSEN, J. *Why you only need to test with 5 users*. 2000. Disponível em: <<http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>>. Acesso em: 27 set. 2015.

PALMIERI, M.; SINGH, I.; CICCHETTI, A. *Comparison of Cross-Platform Mobile Development Tools*. In: *Intelligence in Next Generation Networks (ICIN)*, Berlin, p 179 - 186, Outubro de 2012.

PHONEGAP, **PhoneGap**. 2015. Disponível em: <<http://phonegap.com/>>. Acesso em: 30 set. 2014.

PRESSMAN, S. *Software Engineering: A Practitioner's Approach*. McGraw Hill, 7ª edição. 2011.

POPA, M. *Considerations Regarding the Cross-Platform Mobile Application Development Process*. In: *Economy Informatics*, v. 13, n. 1/2013, p 40-52, Outubro de 2013.

REDDA, Y. *Cross platform Mobile Applications Development – Mobile Apps Mobility*. 2012. Disponível em: <<https://daim.idi.ntnu.no/masteroppgaver/008/8111/masteroppgave.pdf>>. Acesso em: 25 ago. 2014. Dissertação – Norwegian University of Science and Technology, Noruega, 2013.

ROCHA, K. *Como será 2015 para o mercado mobile?*. 2015. Disponível em: <<http://startupi.com.br/2015/01/como-sera-2015-para-o-mercado-mobile/>>. Acesso em: 28 mai. 2015.

ROTA URBANA. **Rota Urbana**. 2013. Disponível em: <<http://www.rotaurbana.net.br/>>. Acesso em: 12 jun. 2015.

SAMBASIVAN, D. et al. *Generic framework for mobile application development*. In: Internet (AH-ICI), *Second Asian Himalayas International Conference on*, Kathmandu, Novembro de 2011. Disponível em: <<http://ieeexplore.ieee.org/ielx5/6104277/6113924/06113938.pdf?tp=&arnumber=6113938&isnumber=6113924>>. Acesso em: 29 set. 2015.

SILVA, M.; SANTOS, M. **Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares**. In: T.I.S., São Carlos, v. 3, n. 2, p. 162-170, mai./ago. 2014.

SILVA, M. et al. **Desenvolvimento de um Caderno de Campo para Plataformas Móveis utilizando PhoneGap**. 2013. Disponível em: <http://200.129.241.80/sbiagro/sbianais/paginas/trabalhos/118650_1.pdf>. Acesso em: 14 jun. 2015.

SINGH, I.; PALMIERI, M. *Comparison of Cross-Platform Mobile Development Tools*. In: IDT *Mini-conference on Interesting Results in Computer Science and Engineering (IRCSE '11)*, Suécia, Outubro de 2011.

SHOTTS, K. **PhoneGap 2.x Mobile Application Development HOTSHOT**. Birmingham: Packt Publishing, 2013.

SMUTNÝ, P. *Mobile development tools and cross-platform Solutions*. In: Carpathian Control Conference, p. 653-656, Maio de 2012.

SOMMERVILLE, I. **Engenharia de Software**, 8ª edição. São Paulo: Pearson Addison-Wesley, 2007.

STARK, J. **Building iPhone Apps with HTML, CSS, and JavaScript**. Sebastopol: O'Reilly, 2010.

OHRT, J.; VOLKER, T. *Cross-Platform Development Tools for Smartphone Applications*. In: *Computer*, High Tatras, v. 45, n. 9, p. 72-79, Setembro de 2012.

VAKULENKO, M. *Mobile Platforms: The Clash of Ecosystems*. London: VisionMobile, 2011. Disponível em: <<http://www.visionmobile.com/blog/2011/11/new-report-mobile-platforms-the-clash-of-ecosystems/>>. Acesso em: 21 ago. 2014.

W3C. *Cascading Style Sheets*. 2015. Disponível em: <<http://www.w3.org/Style/CSS/>>. 30 set. 2014a.

_____. **HTML**. 2015. Disponível em: <<https://docs.webplatform.org/wiki/html>>. 30 set. 2014b.

W3SCHOOLS. **HTML**. 2015. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 30 set. 2014.

ZIMMERMAN, B.; Arizona State University. *Evaluation of Cross-Platform Mobile Frameworks*. 2013. Disponível em: <<http://pooh.poly.asu.edu/Lindquist/Students/pubs/BrandonZimmermanProjectReport.pdf>>. Acesso em: 26 ago. 2014.

APÊNDICES

APÊNDICE I – QUESTIONÁRIO PARA *FEEDBACK*

QUESTIONÁRIO

QUESTIONÁRIO REPASSADO AOS USUÁRIOS BETA.

1. **Você conseguiu pesquisar uma ou mais rotas e visualizá-las?**
 - a. **SIM**
 - b. **NÃO**

2. **Você teve dificuldades em utilizar a aplicação? Em caso positivo, quais?**

3. **De uma nota para a sua experiência com esta nova versão do aplicativo.**
 - a. **1 (Ruim)**
 - b. **2**
 - c. **3**
 - d. **4**
 - e. **5 (Excelente)**

4. **O que você levou em consideração para a aplicação de nota deste aplicativo?**

5. **Quais plataformas para dispositivos móveis você possui?**
 - () **Android**
 - () **iOS**
 - () **Windows Phone**

6. **Com suas palavras, diga qual a importância de uma aplicação que seja disponível para mais de uma plataforma? Plataforma: Android, iOS, Windows Phone e etc.**

APÊNDICE II – RESULTADOS DOS *FEEDBACKS*

Tabela 7. Respostas da primeira, segunda e terceira perguntas do formulário

Você conseguiu pesquisar uma ou mais rotas e visualizá-las?	Você teve dificuldades em utilizar a aplicação? Em caso positivo, quais?	De uma nota para a sua experiência com esta nova versão do aplicativo
NÃO	<p>ROTAS: Após marcar os dois pontos no mapa, e clicar nas distâncias, nenhuma delas exibiu a lista de linhas e nem foi traçada nenhuma rota no mapa.</p> <p>FEEDBACK: Quando clica no botão de feedback o aplicativo encerra instantaneamente.</p> <p>MENUS E BOTÕES EM GERAL:</p> <p>1. Nos aplicativos iOS, a barra superior com hora/carga de bateria/etc sempre fica visível nos apps. A barra de menu do rota fica sobre essas informações, e a visualização fica "misturada".</p> <p>2. Os botões têm que ser pressionados mil vezes até funcionarem, acho que ou por serem muito pequenos, ou por não estarem clicáveis nos lugares certos, ou os dois. Parece quando um div tá no lugar errado, diferente do lugar do ícone.</p>	1
SIM	Não.	3
SIM	Carregamento do mapa.	4
SIM	A rolagem pela lista de rotas é um pouco difícil quando tem muitos resultados.	3
NÃO	Ainda não carrega as listas das rotas. O menu superior ainda continua em cima das informações do iOS (bateria, hora, etc).	3
SIM	Não.	5
NÃO	Extremamente lento. Não é intuitivo. Não tem movimentos com os dedos. Parece site de internet. Não tem cara de aplicativo. Treme as telas quando rolamos. As janelas ficam cortadas as vezes.	1
SIM	Sim. Os ícones do sistema não são tão claros quanto a sua função. Também não consegui definir um ponto final no mapa, somente o ponto de início. O mapa poderia ser mais fluído.	2
SIM	Nenhum.	5
SIM	Não	4

Tabela 8. Respostas da quarta, quinta e sexta perguntas do formulário

O que você levou em consideração para a aplicação de nota deste aplicativo?	Quais plataformas para dispositivos móveis você possui?	Com suas palavras, diga qual a importância de uma aplicação que seja disponível para mais de uma plataforma?
---	---	--

Não funcionou a rota, aí não tem como avaliar bem.	iOS	Apesar de ser popular, nem todo mundo possui Android. E quem tem iPhone também pode ser pobre e pegar ônibus (eu).
A demora na iniciação do aplicativo; O movimento do mapa poderia ser mais fluido, no momento o movimento do mapa é travoso; A demora do loading do mapa; Com exceção desses erros o aplicativo funciona bem, e fácil de utilizar com design e ferramentas bem intuitivas e práticas. Além de ser um aplicativo bem interessante com uma função bem útil para o dia-a-dia.	Windows Phone	Para alcançar todos os usuários e manter uma interação entre diferentes plataformas onde todos podem usar o serviço oferecido pelo aplicativo, deixar uma plataforma de fora é perder clientes e usuários importantes.
O uso devido da aplicação que conseguiu atender minha requisição.	Windows Phone	Portabilidade e atender todos os usuários.
A experiência na utilização do aplicativo e a integração com os padrões da interface gráfica do sistema operacional iOS.	iOS	É muito importante, pois assim pois a aplicação pode ser usada por um número maior de usuários.
Os ícones e aparência geral melhoraram em relação a primeira versão!	iOS	Já respondi da outra vez. Parabéns pela atualização, vamos melhorando.
Sua utilidade.	iOS	Versatilidade e acessibilidade.
–	Android, iOS	Abranger a todos os grupos de pessoas.
O aplicativo não é intuitivo e às vezes as ações que fazemos não são reconhecidas.	Windows Phone	Uma aplicação disponível para várias plataformas é extremamente importante, pois todo o tipo de público pode utiliza-la, principalmente um aplicativo com cunho social, como este. Lançar um aplicativo somente para uma plataforma X ou Y, além de limitar a quantidade de seu uso, pode fazer com que alguém seja obrigado a comprar um aparelho que não é de seu interesse somente para usar uma aplicação.
Rapidez.	Windows Phone	–
Com várias plataformas, pode ser utilizado por todos.	iOS	Praticidade para chegar em diferentes lugares.