

Um estudo de caso com análise comparativa entre plataformas para aplicações móveis aberta e proprietária: Android e iOS

Marivaldo Mascarenhas¹, Mario Martins¹, Lucas Bulcão¹,
Jailson de Brito¹, Vaninha Vieira^{1,3}, Adolfo Duran^{2,3}

¹ Departamento de Ciência da Computação
Universidade Federal da Bahia, Salvador, BA, Brasil,

² Centro de Processamento de Dados
Universidade Federal da Bahia, Salvador, BA, Brasil,

³ Fraunhofer Project Center for Software and Systems Engineering
Universidade Federal da Bahia,
Salvador, BA, Brasil,

{marivaldojr, marioaugusto, lucasbulcao, jailson, vaninha}@dcc.ufba.br,
adolfo@ufba.br

Resumo. *As plataformas para desenvolvimento de aplicações móveis Android e iOS juntas detém a maior parte do mercado. Cada plataforma tem suas particularidades: ambiente de desenvolvimento, custo de desenvolvimento, ferramentas de apoio ao desenvolvedor, e distribuição das aplicações. Como desenvolver para as duas plataformas requer mais recursos, muitas vezes o desenvolvedor precisa decidir qual plataforma vai escolher. Este trabalho apresenta resultados de uma análise comparativa de desenvolvimento para as plataformas Android e iOS com um estudo de caso real. Os resultados mostram que a depender do contexto, uma plataforma pode ser mais adequada que a outra.*

1. Introdução

O desenvolvimento de aplicações para dispositivos móveis está cada vez mais difundido na indústria e na academia [Al-Khalifa and Al-Subaihin 2012, O'Rourke et al. 2010, Kurkovsky 2009, Heikkinen and Still 2008, Hill and Wesson 2008]. Entre as plataformas para aplicações móveis existentes, Google Android ¹ e Apple iOS² juntas detém cerca de 91.1% do mercado [IDC 2013]. Uma questão comum na concepção de uma aplicação móvel é a escolha da plataforma [Palme et al. 2010, Goadrich and Rogers 2011, Ribeiro et al. 2011]. Entre os critérios mais comuns na escolha da plataforma estão: público alvo da aplicação; afinidade dos desenvolvedores com a plataforma, ferramentas e linguagens; e custo de licenças e hardware para o desenvolvimento. Esses e outros fatores quando desconsiderados podem prejudicar o desenvolvimento, a evolução e a manutenção da aplicação. Esta decisão precisa ser bem pensada.

Este trabalho sucede um estudo exploratório sobre o uso de carona colaborativa [Vieira et al. 2012a] onde foram levantados requisitos para o desenvolvimento de uma aplicação móvel para carona colaborativa. Entre os requisitos estão aspectos de

¹<http://www.android.com>

²<http://www.apple.com/ios>

comunicação, coordenação, cooperação e percepção. Não encontramos evidências suficientes sobre como as plataformas iOS e Android suportam esses aspectos.

O projeto Ubibus[Vieira et al. 2012b] tem como objetivo investigar, especificar e desenvolver soluções tecnológicas de apoio a usuários de transporte público, facilitando o seu acesso a informações contextuais em tempo real sobre meios de transporte disponíveis e condições dos veículos e vias urbanas. Como parte do desenvolvimento de uma aplicação móvel, tivemos a chance de desenvolver a mesma aplicação nas plataformas Android e iOS. Este trabalho traz resultados da análise comparativa de desenvolvimento para as plataformas iOS e Android com um estudo de caso real no projeto UbiBus: um sistema de carona colaborativa. O objetivo deste trabalho foi de mostrar os resultados das particularidades de cada plataforma quanto a: ambiente de desenvolvimento; custos de equipamentos e licenças; documentação e comunidade de desenvolvimento; ferramentas de simulação; distribuição da aplicação. Esperamos que este trabalho sirva de insumo em projetos de desenvolvimento de aplicações móveis em contextos acadêmicos e industriais.

O restante deste artigo está organizado da seguinte forma. Na seção seguinte são apresentados os trabalhos relacionados. Na seção 3 é apresentada uma caracterização do cenário e ferramentas analisadas. Na seção 4 apresentamos o estudo de caso realizado e a análise dos resultados. Por fim, a Seção 5 temos a conclusão e os trabalhos futuros.

2. Trabalhos relacionados

Para reforçar a relevância do presente trabalho e enriquecer os critérios para a análise comparativa das plataformas para aplicações móveis, foram pesquisados trabalhos correlatos. Identificamos os trabalhos comparativos dos últimos 4 anos através dos sistemas de busca ACM Digital Library ³, IEEE Explorer ⁴, e Google Scholar ⁵ e selecionamos os mais detalhados. Como as plataformas iOS e Android estão em constante evolução, e os critérios de análise podem variar de acordo com o contexto, ainda há espaço para contribuição sobre o tema.

Em [Goadrich and Rogers 2011], os autores apresentam uma análise comparativa de desenvolvimento entre as plataformas móveis Android e iOS. A principal questão do trabalho é qual plataforma o setor acadêmico deve ensinar para alunos de graduação. Os autores comparam as plataformas com exemplos práticos e questões sobre hardware, interface gráfica e SDKs. As conclusões finais são que o desenvolvimento iOS em relação ao Android pode ser mais difícil devido ao hardware e ferramentas específicas e a linguagem de programação menos comum. Ambas plataformas ajudam os alunos a entenderem conceitos de desenvolvimento de aplicações móveis através de experiência da sala de aula excitante, o que corrobora para relevância da discussão.

Já Tracy [Tracy 2012] relata a experiência do desenvolvimento de uma ferramenta nas plataformas iOS e Android. As principais diferenças apresentadas foram particularidades de interface gráfica em cada plataforma, como por exemplo o botão de voltar presente no Android mas não no iOS. Tracy recomenda que o desenvolvedor se especi-

³<http://dl.acm.org>

⁴<http://ieeexplore.ieee.org>

⁵<http://scholar.google.com>

alize na plataforma que vai desenvolver, pois esse conhecimento aprofundado pode ser crucial para o desenvolvimento de determinadas funcionalidades na aplicação.

Em [Ribeiro et al. 2011], são comparados o desenvolvimento de aplicações iOS e Android apresentando detalhes de cada plataforma como seu histórico, arquitetura, componentes fundamentais, interface com o usuário e recursos. Como estudo de caso, um jogo é implementado utilizando as duas plataformas, analisando os sensores de movimento. A comparação entre as duas plataformas segue as métricas: ambiente de desenvolvimento, linguagem de programação e documentação. Os autores concluem que os pontos fortes do Android são: maior difusão da linguagem de programação Java em relação ao Objective-C do iOS; o emulador dispõe de mais funcionalidades; a licença ser aberta possibilita criação de aplicações mais robustas. Já os pontos fortes do iOS em relação ao Android são: o ambiente de desenvolvimento XCode é melhor integrado e possui ferramentas de interface superiores; a pequena variedade de dispositivos facilita a compatibilidade das aplicações desenvolvidas.

Esses trabalhos trazem informações acerca do desenvolvimento entre plataformas móveis, com apresentações de números comparativos, análises nos âmbitos software e hardware que são relacionados aos SO e análises comportamentais dos desenvolvedores. Ainda restam questões que não ficaram claras sobre ferramentas, custos de desenvolvimento e instrumentos de apoio ao desenvolvedor como documentação e simuladores. Este trabalho visa preencher esta lacuna.

3. Caracterização do cenário e plataformas analisadas

O projeto Ubibus [Vieira et al. 2012b], visa um sistema de transporte público mais inteligente, dinâmico e eficiente. O foco desse projeto é apresentar informações relevantes ao usuário do transporte público, como o tempo de chegada previsto dos ônibus e fatores que estão afetando o trânsito para auxiliar o usuário na tomada de decisões. No entanto, auxiliar apenas o sistema transporte público convencional pode não ser a única solução.

Um dos subprojetos do Ubibus é o Ubibus-Cars, um sistema de caronas solidárias que visa apoiar e incentivar indivíduos que compartilham de um mesmo trajeto a utilizar o mesmo veículo para chegar até o destino ou uma localidade que facilitará sua locomoção, um ponto de ônibus, por exemplo. Como estudo de caso para aplicação do Ubibus-Cars consideramos o ambiente da Universidade Federal da Bahia, haja vista que a utilização da universidade como local de testes traz alguns benefícios: aumentar a confiabilidade entre os usuários, já que frequentam o mesmo ambiente; (ii) disponibilidade de uma rede sem fio com acesso a internet disponível em todos os campi; (iii) apresentar uma demanda real dos usuários, com o aumento de estudantes devido a programas de incentivo ao acesso ao ensino superior como o REUNI⁶, surgiram problemas como falta de vagas para estacionamento, super-lotação dos ônibus e indisponibilidade de ônibus no horário noturno próximo à Universidade.

Através de estudos e questionários [Vieira et al. 2012a], constatou-se que se a prática da carona solidária fosse largamente usada pelos alunos e funcionários da instituição, esses problemas poderiam diminuir. Foram escolhidas como plataformas de desenvolvimento do Ubibus-Cars as plataformas de aplicações móveis Android e IOS, os quais são descritos na próxima seção.

⁶<http://reuni.mec.gov.br>

3.1. Plataformas móveis Android e IOS

Segundo uma pesquisa feita pelo International Data Corporation (IDC) [IDC 2013], as plataformas Android e iOS representaram, em 2012, 87,6% dos 722,4 milhões de dispositivos vendidos, contra 68,1% das 494,5 milhões de unidades vendidas no mesmo período de 2011. Esses números evidenciam o crescimento e o domínio desses sistemas. Ainda considerando o ano de 2012, os dispositivos Android simbolizaram 70,1% das vendas, foram 497,1 milhões de aparelhos vendidos, com um crescimento de 104,1% em relação a 2011, enquanto que smartphones com iOS representaram 18,8% das vendas com 135,9 milhões de unidades vendidas, apresentando um crescimento de 46,0% em relação ao ano anterior.

3.2. Ferramentas utilizadas

As ferramentas utilizadas no desenvolvimento para Android foram o Android SDK, o IDE Eclipse com o plugin ADT (Android Development Tools) e a linguagem de programação Java. A escolha dessas ferramentas deve-se a maior familiaridade da equipe com elas e pela escolha por fazer uma aplicação nativa. O Android SDK é o kit de desenvolvimento provido pela Google que contém as ferramentas e bibliotecas necessárias para construir e testar aplicações para Android. Junto com o SDK é disponibilizado o simulador de um dispositivo com Android que, apesar das limitações, através dele dá para fazer a maior parte dos testes, com a vantagem de poder simular os diversos tamanhos de tela, resoluções e versões do sistema.

O Eclipse foi o IDE escolhido para deixar o trabalho mais rápido. Com o auxílio do plugin ADT desenvolvido pela Google, é uma ferramenta que dispõe de diversos recursos para o desenvolvimento de aplicações para Android. Além dessa integração com as ferramentas da Google, o Eclipse ainda tem um bom suporte à linguagem de programação utilizada, o Java.

O desenvolvimento de aplicativos para plataforma IOS são necessários um computador Apple Macintosh com processador Intel, a ferramenta Xcode e o SDK da Apple. O Xcode suporta a API Cocoa Touch, as linguagens Objective-C e Apple Script, entre outros recursos necessários para o desenvolvimento de aplicativos. Apesar do seu uso ser gratuito, o XCode está disponível apenas para o sistema operacional Macintosh, o que restringe o desenvolvimento.

Análogo ao Eclipse para Android, o Xcode possui um emulador de iPhone e um emulador de iPad, denominados de IOS Simulator. Esses emuladores permitem que diversos testes sejam feitos no ambiente desktop, isto é, sem a necessidade da presença real de dispositivos com IOS.

4. Estudo de caso

4.1. Caracterização do estudo de caso

Para analisar as plataformas para aplicações móveis Android e iOS conduzimos um estudo de caso a partir da metodologia proposta por Wohlin [Wohlin et al. 2000]. Dessa forma, o estudo de caso tem as seguintes características:

- Objeto de estudo: o objeto de estudo são as plataformas de desenvolvimento para aplicações móveis Android e a iOS;

- Propósito: o propósito deste estudo é comparar as duas plataformas em questões de ambiente de desenvolvimento, programação e custos de desenvolvimento.
- Perspectiva: a perspectiva de análise é de alunos de Iniciação Científica do Curso de Ciência da Computação da Universidade Xis, que utilizam as plataformas no desenvolvimento da aplicação UbiBus-Cars.
- Foco: o foco do estudo é a análise comparativa através do levantamento de pontos fortes e fracos das plataformas.
- Contexto: As plataformas serão utilizadas no desenvolvimento da aplicação UbiBus-Cars e outras aplicações do projeto UbiBus.

4.2. Critérios para análise

Selecionamos os critérios com base nas experiências que tivemos nas plataformas e analisando outras estruturas presentes nos trabalhos relacionados, adicionando assim os critérios relevantes à compreensão das divergências que consideramos significativas, no âmbito de desenvolvimento entre as já mencionadas plataformas móveis, para o nosso estudo de caso. Para cada plataforma avaliamos características do ambiente de desenvolvimento necessário e facilidade de uso de suas ferramentas (hardware e software), suporte dado pelas respectivas comunidades de desenvolvimento, os softwares para simulação das aplicações, custos para desenvolvimento em ambas, aspectos de distribuição de aplicações nos meios disponibilizados pelas plataformas da aplicação desenvolvida e adaptabilidade. Sendo tais critérios já mencionados temas com uma certa gama de conteúdo, são compostos de pequenas subdivisões que são apresentadas no decorrer do artigo com o intuito de melhorar o entendimento.

4.2.1. Ambiente de desenvolvimento

Através da análise da Tabela 1 observamos que o Android provê maior flexibilidade quanto ao ambiente e desenvolvimento. Enquanto o desenvolvimento Android é possível através dos sistemas operacionais Windows, Linux e MacOS, o desenvolvimento iOS só é possível através do MacOS. Isto leva também a necessidade de um hardware específico já que apenas os computadores desenvolvidos pela Apple suportam o sistema operacional MacOS. Outra questão é a IDE, para desenvolvimento iOS só é possível através da ferramenta XCode, já no desenvolvimento Android que apesar da IDE mais recomendada ser o Eclipse, o desenvolvimento pode ser com outras IDEs. No caso do Android, é possível também desenvolver sem uma IDE, utilizando editor de texto e compilador em linha de comando. Um ponto em comum é que as linguagens de programação utilizadas no desenvolvimento Android e iOS estão sob licenças livres.

Tabela 1. Ambiente de desenvolvimento

Nº	Critério	Android	iOS
1.1	Multi-plataforma (SO)	Sim	Não
1.2	Multi-IDE	Sim	Não
1.3	Linguagem livre	Sim	Sim

Em resumo, Tabela 2 mostra as características do ambiente de desenvolvimento Android e iOS. O Android utiliza Java como linguagem de programação, e o iOS,

Objective-C. Para o desenvolvimento Android, além do Eclipse é preciso obter a SDK disponibilizada pela Google. O XCode já vem com todas ferramentas necessárias para o desenvolvimento iOS.

Tabela 2. Ambiente de desenvolvimento

Nº	Critério	Android	iOS
2.1	Plataformas	Windows, Linux, MacOS	MacOS
2.2	IDEs	Eclipse, NetBeans, etc	Xcode
2.3	Linguagem	Java, XML	Objective-C

4.2.2. Comunidade de desenvolvimento

O Android possui uma grande comunidade de desenvolvimento, com fóruns e listas de discussão. Além disso, o Google disponibiliza a documentação Android com exemplos no site Android Developers⁷. O iOS também possui uma documentação bem estruturada com guias e exemplos para facilitar o trabalho do desenvolvedor, esta documentação pode ser vista no site iOS Dev Center⁸.

Tabela 3. Comunidade de desenvolvimento

Nº	Critério	Android	iOS
3.1	Documentação gratuita	Sim	Sim
3.2	Códigos de exemplo	Sim	Sim
3.3	Fóruns de discussão	Sim	Sim

4.2.3. Simulação

No desenvolvimento para dispositivos móveis nem sempre o desenvolvedor tem em mãos dispositivos para testar a aplicação. Nesse contexto uma ferramenta para visualizar a aplicação em desenvolvimento é muito importante. A SDK Android provê a ferramenta Android Emulator e o XCode tem o iOS Simulator. Analisamos numa escala de 0 a 10 ambas ferramentas, estas notas correspondem à experiência da equipe de desenvolvimento quanto ao uso dos dois simuladores, levando em conta os critério citados na 4. Apesar das duas ferramentas apresentarem semelhanças o simulador do iOS se mostrou superior em todos os aspectos, seguido de perto pelo Android Emulator, sendo dada assim uma nota menor para ele.

O Android Emulator apresentou um tempo de resposta maior que o IOS Simulator. Ambas ferramentas possuem suporte para sensores de movimento, no entanto o sensor de toque de tela do Android Emulator não suporta multi-toque enquanto no iOS Simulator isto é possível de testar. Devido a diversas opções de configuração, o Android Emulator a principio não foi fácil de utilizar comparado ao iOS Simulator que além disso, tivemos uma experiência de uso próxima ao dispositivo real.

⁷<http://developer.android.com>

⁸<https://developer.apple.com/devcenter/ios/index.action>

Tabela 4. Simulação

Nº	Critério	Android	iOS
4.1	Desempenho	7	10
4.2	Suporte a sensores	8	10
4.3	Facilidade de uso	8	10
4.4	Experiência de usuário	7	10

4.2.4. Custos de desenvolvimento

O custo para desenvolvimento iOS é mais elevado que o desenvolvimento Android. O computador mínimo para desenvolvimento iOS pode custar até 3 vezes mais do que o mínimo para o desenvolvimento Android. O modelo mais simples disponível na Apple Store custa por volta de R\$4.000,00, enquanto com R\$2.000,00 já é possível comprar um computador de outra marca com configurações semelhantes. Devido a variedade de fabricantes, temos mais opções de dispositivos com Android, podendo optar por um aparelho mais simples e barato que já é suficiente para fazer a maioria dos testes. Com menos de R\$400,00 já é possível adquirir um aparelho com Android, já um dispositivo com iOS não é encontrado por menos de R\$1.000,00.

4.2.5. Distribuição

Após a aplicação pronta é preciso distribuir para os usuários e existem particularidades para cada plataforma. Apesar de ser a loja oficial de aplicativos da Google, o Google Play não é o único canal de distribuição de aplicativos Android⁹. O desenvolvedor é livre para gerar o executável do seu projeto e distribuí-lo da forma que quiser. Essa liberdade não está presente no iOS, dependendo do seu tipo de licença, o aplicativo só pode ser instalado em número limitado de dispositivos, apenas para testar antes de mandar para a loja de aplicativos da Apple.

Por ser uma plataforma livre, o Android está presente em centenas de aparelhos de diversas marcas, inclusive com versões próprias de cada fabricante, que modifica o sistema a fim de deixar os seus aparelhos com uma aparência singular. O iOS está restrito aos produtos da Apple, todas as versões do iPhone, iPad e iPod Touch utilizam o sistema. O pacote de desenvolvimento disponibilizado pela Apple é gratuito, no entanto, caso o desenvolvedor queira publicar seus aplicativos na AppStore (Loja Virtual de Aplicativos da Apple) ou, testar suas aplicações em dispositivos reais, é necessário que o mesmo adquira a licença de desenvolvedor que custa a partir de USD\$99/ano e, dependendo da necessidade de desenvolvimento, pode chegar até USD \$299/ano. Já no Android, o desenvolvedor precisa pagar uma taxa de USD \$25 (taxa única) para criar uma conta e ter acesso ao console de desenvolvedor Android, no entanto é livre para testar os projetos em um dispositivo real sem nenhum tipo de cobrança.

⁹<http://support.google.com/googleplay/android-developer/answer/142471?hl=pt-BR>

¹⁰<http://play.google.com>

¹¹<http://www.apple.com/br/osx/apps/app-store.html>

¹²<http://developer.android.com/distribute/googleplay/publish/register.htm>

Tabela 5. Distribuição

Nº	Critério	Android	iOS
5.1	Fabricantes de dispositivos	mais de 50	1
5.2	Tipos de dispositivos	mais de 1500	10
5.3	Canal de distribuição oficial	Google Play ¹⁰	Apple Store ¹¹
5.4	Distribuição standalone	Sim	Não
5.5	Licença de desenvolvedor	\$25 ¹²	\$99,00 ao ano

4.3. Resultados

Nesta seção, serão discutidas as diferenças de cada plataforma (Android e IOS) tendo como base para a comparação o desenvolvimento da aplicação Ubibus-Cars. Critérios como interface e facilidade de desenvolvimento serão levados em conta nesta análise.

Em relação a interface, há uma dificuldade maior em adequar as aplicações aos diferentes tipos e tamanhos de telas no Android, haja vista que o sistema está presente em diversos aparelhos das mais variadas marcas. Por isso, é importante o uso do simulador para ver como o aplicativo se comporta em dispositivos de tamanhos diferentes. No nosso aplicativo, conseguimos fazer essa adaptação usando as próprias ferramentas do SDK, impedindo a rotação da tela, através do código *android:screenOrientation="nosensor"* no *AndroidManifest* (arquivo XML presente em todos os projetos Android, ele possui as configurações necessárias para integrar o aplicativo e o sistema operacional, como permissões, uso de sensores, compatibilidade com as versões do sistema, entre outras), e adicionando *ScrollView* onde necessário, item de configuração de layout que permite aos usuários percorrer uma área que ocupa mais espaço do que o display físico do dispositivo, ou simulador, deslizando a tela através do touchscreen. A Figura 1a mostra o simulador emulando a tela cadastro de caronas em um aparelho com 2.3 polegadas onde é necessário deslizar a tela para ver o botão de cadastro, já a Figura 1b mostra uma tela de 3.7 polegadas com todos os elementos já presentes na tela. Os smartphones com o sistema IOS não apresentam essa grande variação de tamanho, fazendo com que o tempo para criar uma tela seja menor em relação ao Android.

Por ter uma liberdade maior na customização, o programador precisa ter um pouco mais de trabalho se quiser fazer uma interface gráfica elaborada, já que o Kit de desenvolvimento oferece apenas as versões básicas de elementos como botões e campos de texto.

Um problema no desenvolvimento Android é que nem sempre é possível testar todas as funcionalidades do aplicativo no simulador. No Ubibus-Cars, por exemplo, não foi possível testar as funcionalidades relacionadas ao uso de mapas, sendo necessário um aparelho real para fazer os testes. Isso acontece porque, com o lançamento da nova versão da API do Google Maps para Android, o simulador não dá mais suporte a aplicações que utilizam este recurso, sendo que no iOS é um serviço integrado da Apple, que não tem necessidade de plugins, nem no Xcode, para o simulador, nem no dispositivo real.

5. Conclusões e trabalhos futuros

Este trabalho apresentou um estudo de caso com análise comparativa entre plataformas para aplicações móveis aberta e proprietária - Android e iOS. Conduzimos o estudo de

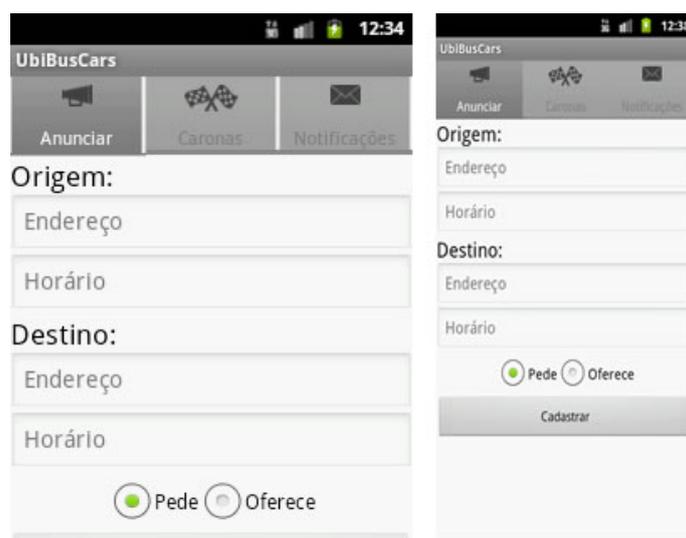


Figura 1. Comparação entre tamanhos de tela do UbiBus-Cars no simulador. (a) Tela de 2.3 polegadas. (b) Tela de 3.7 polegadas

caso com o propósito de comparar as duas plataformas nos quesitos referentes ao desenvolvimento de uma aplicação sob a perspectiva de alunos de Iniciação Científica do curso de Ciência da Computação da Universidade Xis. O foco foi o estudo da análise comparativa através dos pontos fortes e fracos das plataformas, no contexto do desenvolvimento da aplicação UbiBus-Cars em ambas plataformas. Definimos critérios diferentes para avaliação das plataformas com base nos requisitos levantados em um estudo exploratório anterior.

De acordo com os resultados obtidos podemos concluir que a plataforma Android provê um ambiente desenvolvimento mais flexível do que iOS. Além disto, o custo para desenvolvimento iOS pode chegar a 3 vezes o valor do custo do desenvolvimento Android, considerando aquisição de computador, dispositivos e licença de desenvolvedor. Por outro lado, a plataforma iOS dispõe de ferramentas de simulação de mais fácil uso e mais versáteis do que as ferramentas Android. A distribuição da aplicação Android e os testes em dispositivos reais são mais acessíveis do que no desenvolvimento iOS. Ambas plataformas dispõem de vasta documentação e a comunidade de desenvolvedores é bastante engajada, o que auxilia quem está começando a desenvolver para esta área.

Como trabalhos futuros, percebemos a necessidade de realizar uma nova análise comparativa para avaliar outros aspectos como, por exemplo, experiência de usuário, adaptabilidade e performance. Um outro aspecto que poderemos analisar no futuro a partir deste trabalho é a evolução da aplicação com cada plataforma, destacando aspectos de: atualização da aplicação; compatibilidade com novas versões das plataformas; e manutenibilidade do código fonte.

Agradecimentos

Esse trabalho é apoiado pelo CNPq (projeto UbiBus, processo 560135/2010-6), RNP-CTIC (projeto SIMTUR), PIBIC e [parcialmente] apoiado pelo INES¹³.

¹³<http://www.ines.org.br>

Referências

- Al-Khalifa, H. S. and Al-Subaihin, A. A. (2012). Introducing mobile widgets development in an advanced web technologies course. In *Proceedings of the 13th annual conference on Information technology education, SIGITE '12*, pages 61–64, New York, NY, USA. ACM.
- Goadrich, M. H. and Rogers, M. P. (2011). Smart smartphone development: ios versus android. In *Proceedings of the 42nd ACM technical symposium on Computer science education, SIGCSE '11*, pages 607–612, New York, NY, USA. ACM.
- Heikkinen, M. T. and Still, J. (2008). Benefits and challenges of new mobile service development in r&d network. *Personal Ubiquitous Comput.*, 12(1):85–94.
- Hill, R. and Wesson, J. (2008). Using mobile preference-based searching to improve tourism decision support. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology, SAICSIT '08*, pages 104–113, New York, NY, USA. ACM.
- IDC (2013). Android and ios combine for 91.1market in 4q12 and 87.6
- Kurkovsky, S. (2009). Engaging students through mobile game development. In *Proceedings of the 40th ACM technical symposium on Computer science education, SIGCSE '09*, pages 44–48, New York, NY, USA. ACM.
- O'Rourke, J., MacDonald, I., and Goldschmidt, D. (2010). Learning computer science concepts using iphone applications. *J. Comput. Sci. Coll.*, 25(6):121–128.
- Palme, E., Tan, C.-H., Sutanto, J., and Phang, C. W. (2010). Choosing the smart phone operating system for developing mobile applications. In *Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business, ICEC '10*, pages 146–152, New York, NY, USA. ACM.
- Ribeiro, F. I. N., Ferraz, C. A. G., and Ferraz, F. S. (2011). Uma abordagem comparativa do desenvolvimento de aplicações para dispositivos móveis.
- Tracy, K. (2012). Mobile application development experiences on apple's ios and android os. *Potentials, IEEE*, 31(4):30–34.
- Vieira, V., Fialho, A., Martinez, V., Brito, J., Brito, L., and Duran, A. (2012a). An exploratory study on the use of collaborative riding based on gamification as a support to public transportation. In *Collaborative Systems (SBSC), 2012 Brazilian Symposium on*, pages 84–93. IEEE.
- Vieira, V., Salgado, A., Tedesco, P., Times, V., Ferraz, C., Huzita, E., Chaves, A., and Steinmacher, I. (2012b). The ubibus project: Using context and ubiquitous computing to build advanced public transportation systems to support bus passengers. *SBSI 2012*.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). Experimentation in software engineering: An introduction. 2000.