

Desenvolvimento de Aplicações para Dispositivos Móveis: Tipos e Exemplo de Aplicação na plataforma iOS

Alternative Title: Application Development for Mobile Devices: Types and Application Example on the iOS platform

Leandro Luquetti B. da Silva
Centro Universitário Municipal de Franca – Uni-FACEF
Av. Dr. Alonso y Alonso, 2400
Bairro São José, Franca-SP
55 16 3713 4688
leandroluquetti@gmail.com

Daniel Facciolo Pires
Centro Universitário Municipal de Franca – Uni-FACEF
Av. Dr. Alonso y Alonso, 2400
Bairro São José, Franca-SP
55 16 3713 4649
dfpires@gmail.com

Silvio Carvalho Neto
Centro Universitário Municipal de Franca – Uni-FACEF
Av. Dr. Alonso y Alonso, 2400
Bairro São José, Franca-SP
55 16 3713 4619
silviocarvalhoneto@gmail.com

RESUMO

Este artigo apresenta uma comparação entre os tipos de desenvolvimento de aplicações móveis e descreve conceitos sobre o sistema operacional e a arquitetura iOS. Tem ainda como objetivo, apresentar ferramentas de desenvolvimento para aplicativos em dispositivos móveis, com foco na tecnologia iOS, discutindo exemplos para ilustrar diferenças entre as aplicações. O presente estudo é relevante na medida em que é crucial para o desenvolvedor iniciante conhecer as diferentes formas de se construir aplicativos e soluções mobile.

Palavras-Chave

Desenvolvimento de Aplicativos móveis, iOS, Aplicativos Híbridos, Web e Nativos.

ABSTRACT

This paper compares types of mobile application development and describes concepts about mobile operating system and the iOS architecture. The objective is to present development tools for applications on mobile devices, with focus on iOS technology, discussing examples to illustrate differences between applications. This study is relevant because it is crucial for the beginner developer to know the different ways to build mobile applications and solutions.

Categories and Subject Descriptors:

C.5.3 [Computer System Implementation]: Microcomputers--- Portable devices

General Terms:

Design, Languages

Keywords

Development, iOS mobile Applications, Hybrid Applications, Web Applications, Native Applications

1. INTRODUÇÃO

Atualmente, o uso de aplicativos em dispositivos móveis é

pervasivo. De acordo com os dados da pesquisa do instituto Ipsos[1], é possível observar que 40% dos brasileiros utilizam o *smartphone* ao menos todo dia, 27% preferem ficar sem televisão que o *smartphone*, e 73% dos usuários da mobilidade não saem de casa sem seu dispositivo. “O mercado de desenvolvimento de aplicativos móveis está em alta e a demanda por bons profissionais aumenta a cada dia” [2]. Diversas empresas querem ter seus serviços e informações disponíveis e para isto necessitam de desenvolvedores [3]. Toda a demanda por informação, agilidade e disponibilidade em diversos dispositivos móveis, possibilita diversas formas de criar e desenvolver aplicativos para dispositivos portáteis. Este trabalho tem o objetivo de apresentar uma comparação entre tipos de desenvolvimento de aplicações para dispositivos móveis, com o propósito de auxiliar na escolha da forma de desenvolver uma aplicação. O intuito é trazer apontamentos sobre o uso desses aplicativos para dispositivos móveis tendo como foco a plataforma iOS, sistema operacional de código fechado, desenvolvido e controlado pela *Apple*, uma das empresas líderes neste mercado. A decisão de focar na plataforma iOS vem por meio de sua lucratividade e escalabilidade, pois o iOS da *Apple* se destaca como uma das principais plataformas de desenvolvimento móvel do mercado, uma vez que possui uma série de recursos que permitem criar aplicativos diferenciados que proporcionam uma ótima experiência ao usuário final [3].

2. MÉTODO DE PESQUISA

Este estudo tem como método a pesquisa bibliográfica, por meio de revisão literária, juntamente a atividade prática, em torno das fases do projeto para a elaboração de aplicativos móveis. Foram utilizados como base de dados livros, artigos científicos e sites na área de tecnologia e informação como fonte de publicações eletrônicas. As palavras-chave utilizadas para a busca foram: Desenvolvimento, iOS, Aplicativos móveis, Aplicativos Híbridos, Aplicativos WEB, Aplicativos Nativos. Os critérios de inclusão foram artigos publicados a partir do ano de 2011, com texto completo disponível, em inglês e português, e os de exclusão foram artigos anteriores ao ano 2011, com texto incompleto.

3. FUNDAMENTAÇÃO TEÓRICA

Os aplicativos móveis (Apps) são produtos projetados e desenvolvidos para serem executados especificamente em dispositivos eletrônicos móveis, tendo como comuns os PDA's também conhecidos como Palmtops, *Tablets*, leitores de mp3, telefones celulares, e *Smartphones* mais modernos e com larga

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SBSI 2015, May 26–29, 2015, Goiânia, Goiás, Brazil.
Copyright SBC 2015.

capacidade de armazenamento e processamento [4]. Um aplicativo móvel pode ser baixado diretamente do aparelho eletrônico, desde que o dispositivo possua conexão com a Internet. A gama de fornecedores que disponibilizam aplicativos para download através de lojas virtuais como a *Apple Store* - loja virtual da *Apple*, *Play store* - loja virtual do Google para o sistema operacional *Android*, *Windows Phone Store* - loja virtual da Microsoft para *Windows Phone* (entre outras diversas lojas para seus respectivos sistemas operacionais), é enorme e é possível encontrar todo tipo e gênero de aplicação. O número de download destas aplicações está em expansão em ritmo muito forte. Chegam a ser publicados em média trinta mil aplicativos novos por mês na loja virtual da *Apple*.

O mercado de dispositivos móveis é ramificado por diferentes fabricantes, o que inclui uma gama de plataformas de desenvolvimento, sistemas operacionais móveis, software e hardware. A existência de múltiplas plataformas cria uma grande variedade de aplicativos, cada um codificado para ser executado sob sua arquitetura específica, o que segundo Martins et al. [5] é atualmente um dos principais desafios da computação móvel. Em resumo, isto significa que um aplicativo desenvolvido para *iPhone* da *Apple* não funcionará nos sistemas operacionais da Black Berry e *Android*, assim como também não funcionará no *Windows Phone*, ou seja, para cada sistema operacional, deve haver uma nova aplicação. Em outro ponto, o grande fluxo de desenvolvimento causa um abandono na questão segurança da aplicação. Segundo Moraes [6], com as necessidades crescentes por mobilidade e o surgimento de soluções móveis de negócio ou a união entre os sistemas existentes, é cada vez mais difícil manter a segurança das informações. Empresas como *Apple* e Google tentam criar ações para minimizar os trabalhos e processos dos desenvolvedores. No sistema operacional iOS, desde sua versão 5 usa-se criptografia de hardware. A criptografia de hardware faz uso do algoritmo AES256 (*Advanced Encryption Standard*) e garante uma proteção efetiva dos dados armazenados fisicamente no dispositivo. Já no ambiente *Android*, desde sua versão 2.3, chamada de Gingerbread o uso de criptografia é opcional, cabendo ao usuário ativá-la quando assim o desejar.

A Trend Micro, empresa de segurança digital, realizou duas pesquisas em que escutaram 709 profissionais de TI, uma revela que 62% das empresas não estão aptas a proporcionar segurança nos dispositivos. Outro estudo aponta que 78% das empresas veem o mau uso da tecnologia por parte dos usuários, seja acidental ou proposital, como o maior perigo para a segurança da empresa [7]. O sistema operacional da *Apple* iOS roda em dispositivos sensíveis ao toque em aparelhos como *iPad*, *iPhone* e *iPod*, não operacional em hardware de terceiros. O iOS gerencia o hardware do dispositivo e fornece as tecnologias necessárias para programar aplicativos nativos [8]. A arquitetura do iOS é formada por quatro camadas, Cocoa Touch, Media, Core Service e Core OS [9]. Cada uma delas oferece um conjunto de frameworks que podem ser utilizados durante o desenvolvimento. A camada Cocoa Touch é a de mais alto nível e a Core OS a de mais baixo nível (Figura 1). A camada Core OS (núcleo do sistema operacional) é responsável pela gestão de energia, controle de segurança, certificados e sistema de arquivo, e faz a comunicação com os acessórios de hardware externo [10].

Na camada de serviços oferecidos pelo sistema (Core Services) são disponíveis os acessos como o banco de dados SQLite, manipulação de arquivos, preferências, livros de endereço e rede. Também contém tecnologias individuais para suportar recursos

como localização, iCloud, mídia social e acesso à rede [11]. A camada Media contém os gráficos, núcleo de animação, leitor de PDF OpenGL ES e OpenAL, áudio e tecnologias de vídeo para implementar experiências de multimídia. A tecnologia nessa camada facilita a construção de aplicativos [11]. Cocoa Touch é a camada responsável por eventos Multi-Touch e controles, notificações, câmera, alerta e sistema de localização, camada basicamente responsável pelas interações do usuário com sistema [11]. Os aplicativos desenvolvidos para o iOS raramente se comunicam diretamente com o hardware do dispositivo, ao invés disso, os aplicativos se comunicam com o hardware através de um conjunto de interfaces de sistema bem definidas que protegem seu aplicativo de alteração de hardware.

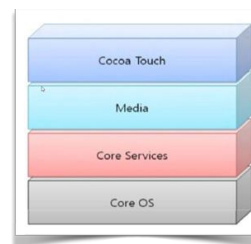


Figura 1: Camadas do iOS

Fonte: adaptado Ferreira [11]

4. RESULTADOS

Na construção de uma aplicação para dispositivo móvel, a escolha do tipo do aplicativo é um ponto muito importante no processo de desenvolvimento do projeto [5]. Ao iniciar o projeto é necessário realizar uma análise minuciosa e estratégica sobre a plataforma, sistemas, produtos e arquiteturas a serem utilizadas. Os principais tipos de aplicativos móveis a serem considerados no início de um projeto são: aplicativos Web Apps (sites móveis), aplicativos móveis nativos e aplicativos híbridos, que consistem respectivamente em soluções feitas para Web formatadas para serem acessadas através do browser dos dispositivos móveis, soluções que são desenvolvidas para um delimitado dispositivo móvel e sistema operacional e aplicações desenvolvidas com a junção do nativo e Web.

4.1. Nativa com foco no iOS

Aplicativos nativos são aplicações comumente encontradas em lojas de aplicativos. São normalmente desenvolvidas utilizando linguagens de programação de nível superior, como o Java para o *Android*, Objective-C para iOS, ou C# para *Windows Phone*. As APIs nativas são fornecidas para o desenvolvedor como parte do SDK da plataforma. As APIs da plataforma são normalmente concebidas para fornecer aplicativos nativos de acesso ideal para as capacidades de hardware, como a câmera do dispositivo e emparelhamento de Bluetooth. Além disso, os usuários podem ser capazes de usar esses aplicativos sem uma conexão com a Internet [12]. Execução nativa consiste essencialmente em um aplicativo desenvolvido para um dispositivo móvel específico e instalado diretamente sobre o próprio dispositivo com interpretação direta pelo sistema operacional. O pacote de desenvolvimento de software para iOS (SDK) contém as ferramentas e interfaces necessárias para desenvolver, instalar, executar e testar aplicativos nativos. O Xcode é responsável pelo ambiente de desenvolvimento integrado da *Apple* (IDE). O Xcode inclui um editor de código, um de interface gráfica do usuário, e muitos outros recursos. O iOS SDK contido no Xcode inclui as ferramentas, os compiladores e as estruturas necessárias

especificamente para o desenvolvimento iOS. Aplicativos nativos em iOS são construídos usando a estrutura de sistema na linguagem Objective-C [10]. A linguagem nativa é executada diretamente no sistema operacional. Ao contrário de aplicações Web, aplicativos nativos são fisicamente instalados em um dispositivo e, portanto, são sempre disponíveis para o usuário. A linguagem de programação recomendada pela *Apple* na plataforma iOS é o Objective-C, havendo possibilidade de utilização da linguagem C e C++. Por ser uma linguagem específica da *Apple*, o principal recurso de pesquisa da linguagem é a própria documentação disponibilizada pela *Apple*, e de acordo com Morandi [13], os conteúdos são completos, o suficiente para entender e desenvolver projetos nessa linguagem.

4.2. WEB APP

No desenvolvimento para dispositivos móveis é possível utilizar aplicações Web com o mesmo objetivo de uma aplicação nativa, porém com limitações de recursos. O Web App refere-se a aplicativos baseados em navegador criados para dispositivos móveis, como *smartphones* ou *tablets*, que podem ser conectados sem fio. Web App é basicamente uma aplicação em marcação HTML e CSS, podendo ser programado em JavaScript [14], ou seja, são sites projetados para telefones móveis baseados na Web. Eles geralmente são parecidos em todas as plataformas e não aproveitam os recursos da plataforma para personalizar a experiência do usuário [12]. Este modelo é recomendado para projetos cujo custo e tempo sejam reduzidos e a necessidade de várias plataformas seja um requisito essencial. Por ser uma aplicação interpretada pelo navegador do dispositivo, a mesma passa a ser visualizada em sistemas operacionais diferentes, tornando assim as aplicações Web Apps àquelas com o menor tempo de projeto. A capacidade de atualizar e manter aplicações Web sem distribuição e instalação de software, em potencialmente milhares de dispositivos, é uma das principais razões para a sua popularidade, e a compatibilidade dos navegadores entre plataformas torna a aplicação Web mais prática e dinâmica. Uma desvantagem dessa abordagem é que o download desses ativos como a biblioteca *Jquery Mobile* para o dispositivo pode não só aumentar o custo associado com o uso de dados, mas também afetar a experiência do usuário devido à latência das redes [12].

4.3. APP HÍBRIDO

A criação de aplicativos para cada dispositivo como *iPhone*, *Android*, *Windows Mobile*, etc. requer diferentes frameworks e linguagens. A aplicação híbrida resolve esta questão usando tecnologias Web baseadas em padrões nativos. Aplicações "híbridas" consistem em uma categoria especial de aplicações web que ampliam o ambiente do aplicativo baseado na Web através da utilização de APIs nativas da plataforma disponível em um determinado dispositivo. O padrão de projeto aplicação híbrida é igualmente aplicável a ambos os ambientes móveis e de desktop [12]. Assim como aplicativos nativos, a aplicação híbrida é executada dentro de um ambiente de processo nativo no dispositivo. Esses aplicativos tipicamente envolvem o conteúdo HTML dentro de um controle de navegador em modo de tela cheia, sem a barra de endereços visível ou outros controles embutidos do navegador. Aplicativos híbridos usam uma camada web-para-nativa abstração (também conhecida como camada de ponte) que permite que o Javascript acesse muitos recursos específicos do dispositivo e APIs nativas que não são geralmente acessíveis a partir do navegador Web móvel sozinho [12]. Uma síntese das vantagens e desvantagens de cada tipo de desenvolvimento de aplicativos mobile está descrita no Quadro 1.

4.4 Exemplo de Aplicação

Como exemplo de aplicação móvel (e seus tipos), nesta seção é apresentado um projeto que consiste em um exemplo básico de visualização baseado nos conceitos de experiência de usuário e consumo de serviço online através de APIs.

Quadro 1: Tipos de Aplicativos Móveis

	Principais Vantagens	Principais Desvantagens
WebApps	- Executados pelo <i>browser</i> , proporcionando o uso em outras plataformas; Atualização e distribuição rápida e abrangente, não precisam ser baixados ou atualizados; Acesso rápido e fácil, os usuários têm acesso imediato pelo <i>Smartphone</i> .	- Pouca ou quase nenhuma integração com o <i>hardware</i> do dispositivo em que está sendo executado; Mais lentos, dependendo da conexão com a <i>Internet</i> ; Interação entre o usuário e o aplicativo menos rica em funcionalidades.
Nativo	Interação entre o usuário e o aplicativo mais rica em funcionalidades e recursos; Velocidade na execução. Independente da <i>Internet</i> .	Uma nova aplicação escrita para cada plataforma diferente Distribuição e atualização dependentes de lojas on-line; (<i>Apple Store</i> , <i>Google play</i>).
Híbrido	Compartilhamento de boa parte do código entre plataformas; Possibilidade do uso de recursos da plataforma com código nativo; Pode ser distribuído lojas on-line; (<i>Apple Store</i> , <i>Google play</i>) Interoperabilidade.	Performance. Limitação de design.

Fonte: Elaborados pelos Autores

O serviço de consulta utilizado neste projeto é fornecido pela Google. O Google Maps possui uma API difundida no mercado, que é utilizada para consultas de geolocalização assim como outros dados disponíveis em seu sistema [15]. O consumo deste serviço é baseado no modelo REST, modelo de transferência de estado representacional que é um padrão de estilo de arquitetura ou desenho utilizado como um conjunto de diretrizes para a criação de serviços Web, e que permitem que qualquer coisa ligada a uma rede se comunique com outros dispositivos através de um ponto comum compartilhado no protocolo de comunicações HTTP (*Hypertext Transfer Protocol*). No projeto além do serviço na Web do Google, também foram utilizadas bibliotecas e frameworks, descritas no Quadro 2.

Quadro 2: Tipos de aplicação e tecnologias utilizadas

Tipo de Aplicação	Linguagem, Bibliotecas, Framework, SDK, IDE.
WEB	HTML5, JQUERY, CSS3, JAVASCRIPT, JSON, NOTEPAD++.
NATIVA	OBJECTIVE-C, JSON, XCODE, IOS SIMULATOR.
HÍBRIDA	OBJECTIVE-C, XCODE, IOS SIMULATOR, CORDOVA, HTML5, JQUERY, CSS3, JAVASCRIPT, JSON.

Estas aplicações foram executadas em um simulador de sistema operacional iOS conhecido como iOS Simulator. Nos casos de aplicação nativa e aplicação híbrida houve a necessidade de compilar a aplicação para iOS utilizando o Xcode, já a aplicação Web apenas tornou-se disponível em uma página da Internet. Na Figura 2 é possível perceber uma aparência muito semelhante principalmente nos tipos Web e Híbrido (2a e 2b), pois possuem a mesma interface, já que somente a aplicação Web exibe a barra de endereço e de navegação do navegador Safari. Na Figura 2c (da aplicação nativa) nota-se algumas pequenas mudanças visuais, como o campo de busca e a barra de navegação mais alargada. Para a codificação Web e Híbrida a única mudança de código além do estrutural do Cordova, foram as chamadas externas do HTML. O Quadro 3 apresenta o código com a mudança.

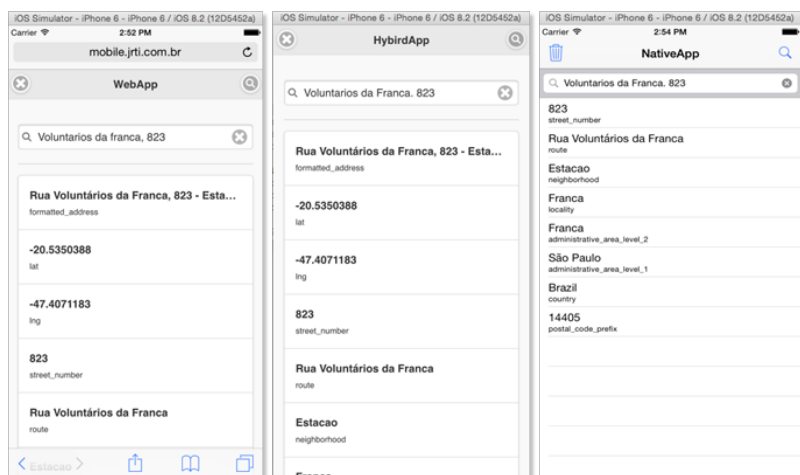


Fig. 2a Web Fig. 2b Híbrida Fig. 2c Nativa
Figura 2 – Exemplos de Telas Aplicação Web, Híbrida e Nativa

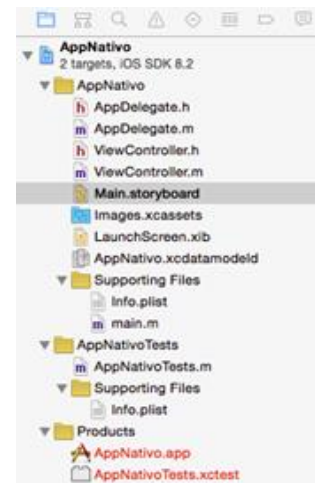


Figura 3: Estrutura do projeto da aplicação nativa

Quadro 3: Exemplos de códigos dos tipos Web e Híbrido

Chamadas externas da aplicação híbrida

```
<link rel="stylesheet" href="css/jquery.mobile.min.css"/>
<script src="js/jquery.min.js"/>
<script src="js/jquery.mobile.min.js"/>
```

Chamadas externas da aplicação Web

```
<link rel="stylesheet" href="//ajax.googleapis.com/ajax/libs/jquerymobile/1.4.2/jquery.mobile.min.css"/>
<script src="//ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"/>
<script src="//ajax.googleapis.com/ajax/libs/jquerymobile/1.4.2/jquery.mobile.min.js"/>
```

Por sempre estar disponível no aplicativo mesmo quando sem conexão com a Internet, a aplicação híbrida exige que as bibliotecas fiquem armazenadas localmente no dispositivo por isto a necessidade de alterar somente as chamadas externas. Em comparativo, a aplicação nativa possui outro paradigma e estrutura. A linha visual da aplicação não é desenvolvida em linguagem de marcação HTML e sim visualmente em Interface Builder pela storyboard no Xcode. A Figura 3 apresenta a estrutura do projeto nativo observando a quantidade de arquivos e parâmetros disponíveis (as aplicações têm código fonte e toda a estrutura do projeto, disponível no repositório GitHub [16]).

5. CONCLUSÃO

O presente trabalho apresentou uma abordagem sobre as possíveis formas de desenvolvimento com foco na plataforma iOS para dispositivos móveis. Foi apresentada uma breve reflexão sobre as dificuldades enfrentadas pelos desenvolvedores em relação à diversidade de plataformas existentes, os tipos de desenvolvimento em iOS com suas características e um exemplo de aplicação. Para pequenos projetos que demandam maior amplitude no que se refere à compatibilidade de dispositivos, a utilização de aplicações híbridas ou Web torna o tempo de desenvolvimento mais curto além de tornar a aplicação multi-plataforma com pouco trabalho de adaptação.

6. REFERÊNCIAS

- [1] Ipsos. Google. Our Mobile Planet Brazil: understanding the mobile consumer. May, 2013.
- [2] Lecheta, R. Desenvolvendo iPhone e iPad. 1.ed. São Paulo: Novatec Editora, 2012.
- [3] Milani, A. Programando para iPhone e iPad: aprenda a construir aplicativos para iOS 1.ed. SP: Ed Novatec, 2012.
- [4] Janssen, Cory. Mobile Application. Disponível em: <http://www.techopedia.com>. Acesso em: 20/2/2015.
- [5] Martins, C.; Antonio, A.; Oliveira, C. A. Os desafios para a mobilização de aplicações baseadas em plataforma Web. Catalão: Enacom, 2013. P. 294-300.
- [6] Moraes, M. E. Mobilidade corporativa desafia CIOs: Gerenciamento de dispositivos móveis pode ser a solução para segurança empresarial. Disponível em: <http://itforum365.com.br/>. Acesso em: 17 dez. 2013.
- [7] Trendmicro. Is your business at risk of losing data? Disp. em: <http://www.trendmicro.com.br>. Acesso em: 10/8/2013.
- [8] Apple. iOS Technology Overview. Disponível em: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>. Acesso em: 18/12/2013.
- [9] Ferreira, M. Aplicativo de gestão energética visando redes elétricas inteligentes para dispositivo móvel. Disponível em: <http://www.lactec.org.br/mestrado/dissertacoes/arquivos/MauricioFerreira.pdf> Acesso em: 27/8/2014.
- [10] Apple. Programming with Objective-C Disponível em: <https://developer.apple.com/library/prerelease/ios/documentation/n/Cocoa>. Acesso em: 18/12/2014.
- [11] Apple. Start Developing iOS Apps Today Disponível em: <https://developer.apple.com/>. Acesso em: 15/6/2014.
- [12] Gok, N; Khanna, N. Building Hybrid Android Apps with Java and JavaScript. California: O'Reilly Media, 2013.
- [13] Morandi, P. Desenvolvimento Mobile nas Plataformas Android e iOS. Disponível em: <http://www.mobiltec.com.br/blog/index.php> Acesso em: 17/12/2013.
- [14] Hales, W. HTML5 and JavaScript Web Apps. California: O'Reilly Media, 2012.
- [15] Google. Serviços da Web da API do Google Maps. Disponível em: <https://developers.google.com/maps/documentation/geocoding/#JSON> Acesso em: 06/1/2015.
- [16] Luquetti, L. Projetos do artigo: Desenvolvimento Móvel: diversas interfaces com enfoque na plataforma iOS Disp. em: <https://github.com/leandroluquetti> Acesso em: 14/1/2015.