ESTRATÉGIAS INTELIGENTES PARA DESENVOLVIMENTO DE APLICATIVOS MOBILE MULTIPLATAFORMA

Dércia Antunes de Souza
derciaantunes@uol.com.br
FATEC - Faculdade de

Jefferson Aparecido França
franca.ajef@gmail.com
FATEC - Faculdade de

Guilherme Forte
forte.182011@yahoo.com.br
FATEC - Faculdade de

Cristina Becker Matos Nabarro
profcrisbecker@gmail.com
FATEC - Faculdade de

Wakim Boulos Saba
wakim.saba@fatec.sp.gov.br
FATEC - Faculdade de

Resumo:O desenvolvimento de aplicativos Mobile utilizando o conceito “multiambiente” (ou multiplataforma) deve atender as principais plataformas: Android, Windows Phone e iOS utilizando o mesmo código fonte. Este trabalho tem como objetivo apresentar estratégias de desenvolvimento multiplataforma para a implementação de aplicativos que funcionem nas principais plataformas. Apresenta-se o conceito deste tipo de desenvolvimento utilizando o framework Cordova, suas principais vantagens e desvantagens. Esse framewok facilita o desenvolvimento de aplicativos, que pode ser utilizado por diversos sistemas operacionais Mobile, permitindo atingir um número muito maior de usuários, a partir da mesma implementação computacional, uma vez que o aplicativo mobile (como produto final de software) não se limita a uma única plataforma. Essa abordagem permite ainda, caso o projeto de software assim necessite, explorar um conjunto de funcionalidades de baixo nível de equipamentos mobile assim como é possível fazer utilizando recursos nativos de uma plataforma específica.

Palavras Chave: Aplicativos - Multiplataformas - Mobile - Cordova - Framework

1. INTRODUÇÃO

O desenvolvimento de aplicativos para mobile tem obtido destaque no mundo tecnológico, deixando de ser tendência e passando a ser uma realidade. São lançados inúmeros aplicativos e não é incomum ler uma reportagem de alguém que tenha se tornado milionário depois de ter desenvolvido um aplicativo para mobile, apresentando em geral, funcionalidades simples. Como toda nova tecnologia, esses casos apresentam vantagens e também problemas. Um dos maiores problemas envolve o suporte de uma característica de qualidade, denominada portabilidade pela ISO/IEC 9126 (2000), para atender uma diversidade de plataformas mobile disponíveis no mercado.

Desenvolver um aplicativo que atenda as principais plataformas (Android, Windows Phone e iOS) pode se tornar uma dor de cabeça devido ao alto custo de horas de programação para todas elas, considerando o tempo necessário para deixar o aplicativo funcionando sem nenhum problema. Segundo Nunes (2013), é comum a necessidade de muitas empresas em criar aplicativos ou mesmo páginas mobile, que atendam a uma boa parte do mercado e que funcionem corretamente nas mais diversas plataformas existentes como Android, IOS, Windows Phone, BlackBerry, entre outras.

O objetivo geral deste trabalho trata de uma questão central a respeito da diversidade de plataformas disponíveis para desenvolvimento de aplicativos Mobile no mercado. Para tanto, procurou-se elucidar a questão: como desenvolver um aplicativo Mobile que funcione em diversas plataformas utilizando o mesmo código fonte?

O delineamento metodológico adotado neste trabalho é a pesquisa bibliográfica, pois foi feito um levantamento em livros e artigos acadêmicos científicos realizados sobre as estratégias de desenvolvimento de aplicativos que funcionam nas principais plataformas a partir de uma única implementação computacional.

Como resposta para a pergunta-problema central, encontra-se disponível no mercado ferramentas, como o *framework*[1] *open-source*[2] chamado Cordova, fornecida pela Apache Community que permite empacotar uma aplicação que utiliza tecnologias web padrão (HTML, CSS e Java Script) para aplicativos Mobile, ou seja, com um único código fonte é possível gerar um aplicativo que funciona nas principais plataformas, podendo acessar recursos nativos de cada uma delas sem a necessidade de desenvolver nenhuma linha de código nativo. Exemplo: plataforma Android, código nativo Java.

Desta forma, a relevância deste trabalho consiste na importância de se desenvolver aplicativos Mobile que atenda as principais plataformas do mercado, Android, iOS e Windows Phone, de uma forma mais viável. Pois, desenvolver o mesmo aplicativo para cada plataforma desejada em sua respectiva linguagem de programação, assim como são construídos, os Apps nativos, gera trabalho excessivo e alto custo no desenvolvimento.

2. REFERENCIAL TEÓRICO

Antes de se desenvolver um aplicativo Mobile, devemos primeiramente, analisar qual o público alvo e em quais plataformas ele deverá atuar. Ao pensar em desenvolver aplicativos mobile, é importante pensar em quais plataformas ele deverá ser disponibilizado. Android é

[1] É um conjunto de ferramentas que dispõe funcionalidades a fim de acelerar o desenvolvimento de aplicações empregando reuso de código.
[2] Quando um software possui seu código fonte aberto. Dependendo da licença, em geral, é permitido fazer alterações diretamente no código fonte.

uma das plataformas mais utilizada no mundo dos smartphones, principalmente no Brasil. iOS é bastante usado, tendo seu principal público, as classes sociais mais altas, o que inclui usuários de maior poder aquisitivo. Windows Phone é uma boa terceira opção, em franco crescimento. E ainda existem outras plataformas no mercado, como BlackBerry, Tizen e outros.

A maioria dos aplicativos Mobile visa atender a grande massa de usuários, desta forma, recomenda-se que o aplicativo funcione pelo menos nas três principais plataformas do mercado: Android, iOS e Windows Phone.

Segundo Lopes (2015 p. 01), "o desenvolvimento para plataformas diferentes tem sido um grande problema, pois, cada plataforma suporta uma linguagem de programação diferente. Por exemplo: a plataforma Android apresenta muitos recursos em Java, plataforma iOS utiliza código nativo Objective C e a plataforma Windows Phone suporta o *framework* .Net, em geral, a linguagem C#. Cada plataforma tem sua combinação de linguagem e, principalmente, APIs específicas".

Desenvolver um aplicativo Mobile que atenda as principais plataformas do mercado pode se tornar uma dor de cabeça devido ao alto custo de mão de obra especializada (programadores) com competência técnica em cada uma das linguagens de programação necessárias. Outro aspecto relevante é o tempo necessário para deixar o aplicativo Mobile funcionando sem nenhum problema (bug).

Isso nos leva direto a questão: como desenvolver um aplicativo Mobile que funcione em diversas plataformas utilizando o mesmo código fonte?

É nesta hora que entram os aplicativos multiplataforma. A empresa Avanti! Tecnologia & Marketing (2015), publicou e defendeu a ideia que a construção de um aplicativo híbrido se torna mais rápido e mais barato do que o desenvolvimento de aplicativos nativos. A redução de tempo se deve à possibilidade de execução do aplicativo híbrido em diferentes plataformas. Devido a essa característica, não existe a necessidade de desenvolver o aplicativo várias vezes para adapta-lo a diferentes plataformas, permitindo assim, menor impacto no orçamento. Em situações que não exigem um alto desempenho do aplicativo, muitas empresas também optam pelo desenvolvimento de aplicativo híbrido ou quando o público-alvo é heterogêneo. Nesses casos, soluções mais genéricas que podem ser utilizadas em múltiplas plataformas, apesar do alto custo de desenvolvimento, apresentam vantagens significativas.

O desenvolvimento de um aplicativo multiplataforma pode ser feito utilizando-se determinados *frameworks* de mercado, como por exemplo, o Cordova (open-source) que permite empacotar uma aplicação desenvolvida a partir das tecnologias web HTML, CSS e Java Script para aplicativos Mobile.

Atualmente, o Cordova é uma das soluções mais comuns para desenvolvimento de aplicativos multiplataforma. Para construir e executar aplicativos, ele usa uma das maiores vantagens da web, ter linguagens padronizadas e o navegador como ambiente de execução. Segundo Lopes (2016 p. 05), "[...] São Apps instaláveis que você pode publicar nas lojas, e pode usar recursos nativos da plataforma, mas são escritas em HTML, CSS e JavaScript".

Desta forma a partir do mesmo código fonte é possível gerar um aplicativo mobile que funciona nas principais plataformas do mercado, podendo acessar recursos nativos de cada uma delas. Isso sem a necessidade de desenvolver nenhuma linha de código especifica da plataforma desejada.

Desenvolver um aplicativo Mobile utilizando as tecnologias web padrão (HTML, CSS e JavaScript) apresenta-se de forma descomplicada a partir da utilização de um framework

como o Cordova, afinal é ele o responsável por encapsular, transformar o código fonte para as plataformas de aplicativos Mobile. Neste contexto, Lopes (2016, p. 5) afirma que:

> Só escrever HTML, CSS e JS não é suficiente para ter um aplicativo no fim. Então, o que o Cordova faz é prover uma casca nativa para o nosso aplicativo responsável por subir um browser que fará a execução do nosso código. O papel do Cordova é apenas criar essa janela de navegador para nós, e fazer a comunicação das nossas chamadas de código para chamadas nativas quando necessário.

Mas como funciona esse *framework*? Será que é simples desenvolver uma aplicação mobile utilizando apenas uma linguagem de programação que atende as principais plataformas do mercado? Fica claro que essas questões passam pela cabeça de quem procura desenvolver um aplicativo multiplataforma.

Primeiramente precisamos entender um pouco mais sobre os aplicativos híbridos. Os aplicativos híbridos são aplicativos que utilizam tecnologias web padrão (HTML, CSS e JavaScript) e podem acessar recursos nativos de cada plataforma mobile, como por exemplo: câmera, GPS, acelerômetro, etc.

São considerados como aplicativos híbridos, pois, ao mesmo tempo em que eles são desenvolvidos para web, eles acessam recursos nativos dos dispositivos em que estão sendo executados. Como foi descrito na matéria da IBM 2013, os aplicativos híbridos contêm dois elementos: um componente web, baseado em linguagem de programação para web, e um container ou “bridge” nativo, que permite acessar os recursos nativos da plataforma e dispositivo.

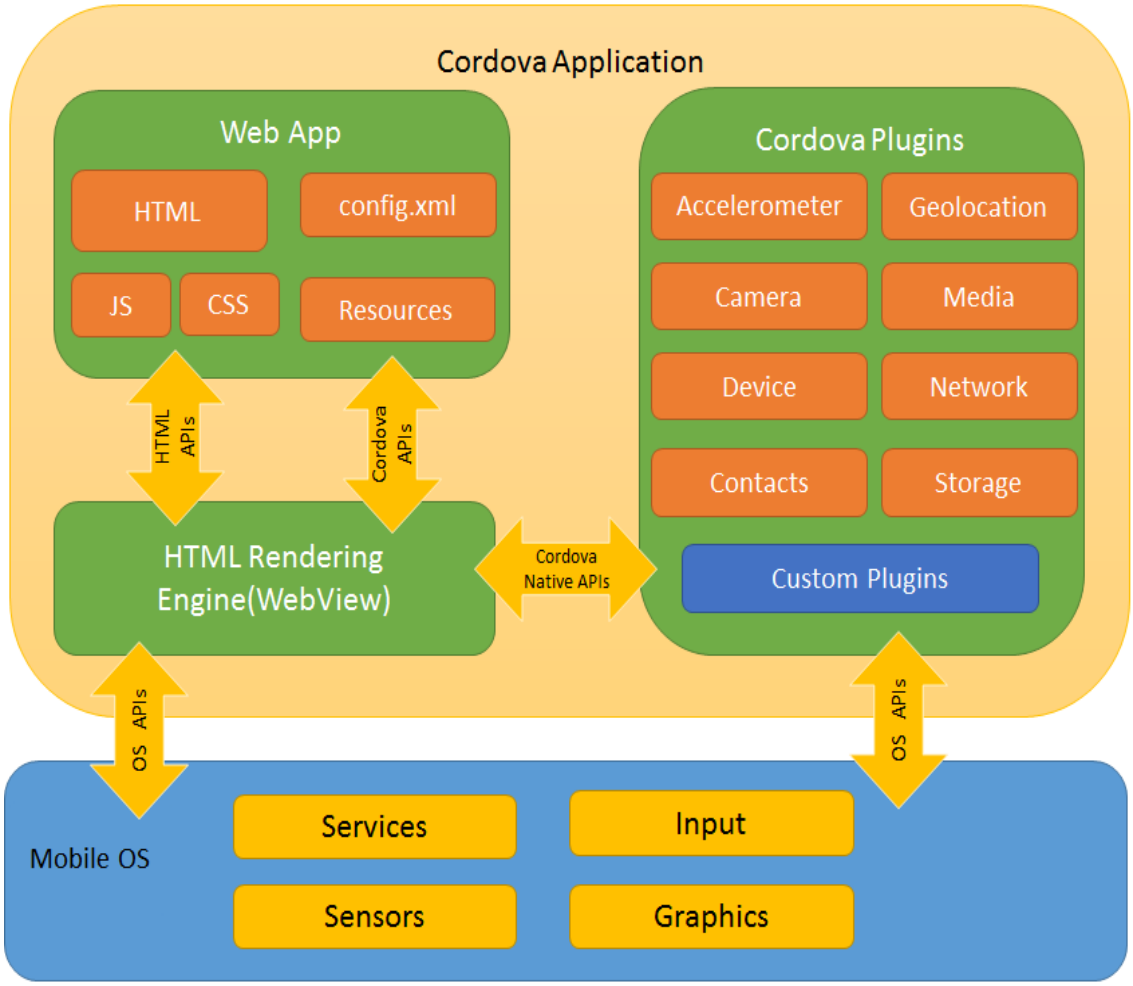
O projeto Open Source Apache Cordova é o container mais utilizado e consiste em um conjunto ferramentas de apoio que permite o aplicativo web acessar os recursos nativos do dispositivo. O aplicativo híbrido tem seu código principal desenvolvido em HTML5 e é envelopado em um container, empacotado como um app nativo e, portanto, residindo em uma app store. (IBM, 2013)

O *framework* Cordova deve ser utilizado pelo desenvolvedor para construir uma aplicação web (WebApp). A WebApp pode ser acessada a partir de um browser fornecido pelo *framework,* denominado WebView, que permite acessar os recursos nativos de um dispositivo Mobile a partir de um conjunto de Plugins fornecidos por Cordova. Tanto o aplicativo WebView quanto o conjunto de plug-ins tem características multiplataforma.

De acordo com Silva (2016), os aplicativos móveis híbridos são construídos com uma combinação de tecnologias web como HTML, CSS e JavaScript. A principal diferença, utilizando Cordova, é que os aplicativos híbridos são acessados pelo aplicativo WebView, que por sua vez, tem acesso aos recursos nativos de uma plataforma móvel. Essa abordagem permite acessar recursos do dispositivo, como acelerômetro, câmera, contatos e muito mais. Estes recursos são, em geral, de acesso restrito por navegadores móveis. Além disso, o *framework* Cordova permite incluir outros elementos de interface nativos (plugins) quando for necessário (ANDRADE, 2016).

A WebView é a forma utilizada para que uma aplicação web seja visualizada como um aplicativo em dispositivos móveis representando uma camada relevante da aplicação híbrida. Como podemos ver na figura 1, o Cordova utiliza da WebView para acessar tanto o código fonte (Web App) quanto os recursos nativos do dispositivo (Cordova plugins) através de APIs específicas, obtendo assim comunicação com a plataforma desejada.

Figura 1 - Arquitetura do aplicativo híbrido utilizando Cordova



Fonte: Cordova (2016)

Também é possível ver que todo acesso aos recursos das diferentes plataformas existentes no mercado é suportado pelo conjunto de APIs que o Framework Cordova disponibiliza para o desenvolvedor. São vários plug-ins de códigos nativos onde cada um deles tem uma função de acesso a recursos de diferentes plataformas. Lembrando que a utilização dessas APIs sempre obedece ao mesmo padrão de implementação, ou seja, não é necessário alterar a implementação das de acesso aos recursos nativos do dispositivo seja qual for a plataforma utilizada para executar o aplicativo desenvolvido.

Lopes (2016) relata que, ao utilizar Cordova, o código fonte web é empacotado para se tornar um aplicativo normal. Esses Aplicativos são mais próximos de aplicativos nativos que aplicativos web. Neste sentido, Lopes (2016, p.7) indica que:

> [...] Elas têm as mesmas vantagens e deficiências de Apps normais: precisam ser geradas para cada plataforma, precisam ser disponibilizadas na loja de cada fabricante, e estão submetidas às regras de cada plataforma. Não são navegáveis, não estão na internet, e não têm URLs. Porém, estão totalmente integradas ao dispositivo. Podem ser instaladas e ser usadas offline. Podem usar APIs da plataforma e usar recursos de hardware avançados. Podem ser divulgados nas lojas e ser vendidas facilmente para os usuários.

Lopes deixa claro que embora os aplicativos sejam desenvolvidos em linguagem de programação para web, eles podem ser executados normalmente sem a necessidade de ter uma conexão com a internet. Com isso pode-se afirmar que um aplicativo híbrido, desenvolvido utilizando o *framework* Cordova, está muito mais próximo de um aplicativo nativo do que de um aplicativo web.

A citação acima também descreve que para que o aplicativo híbrido seja compatível com as plataformas desejadas, embora trate-se do mesmo código fonte, é necessário realizar o empacotamento, com o auxílio do Cordova, para cada uma delas.

A construção de um aplicativo híbrido é mais fácil do que parece. Isso se deve ao uso de um formato de desenvolvimento bem conhecido de aplicações, o desenvolvimento web, que utiliza tecnologias web padrão (HTML, CSS e JavaScript). Inclusive para acessar a recursos nativos do dispositivo.

O Apache Cordova disponibiliza seu framework para várias ferramentas de desenvolvimento. Facilitando ainda mais a sua utilização e aumentado à velocidade de produção do aplicativo. Uma das ferramentas mais utilizadas é o Visual Studio a partir da versão 2013.

Um exemplo de ferramenta para desenvolvimento com utilização do Cordova, é o Visual Studio 2013. As ferramentas do Cordova são lançadas como uma versão de visualização. Elas serão embutidas como parte do Visual Studio 2015. A Microsoft (2016) recomenda e disponibiliza para download o Visual Studio 2015 RTM para desenvolver aplicativos usando o plug-in Visual Studio Tools for Apache Cordova.

Agora que conhecemos algumas características associadas ao funcionamento e ás dificuldades para o desenvolvimento de um aplicativo multiplataforma, é possível analisar algumas das principais vantagens e desvantagens quanto a construção de aplicativos híbridos e nativos.

Figura 2 - Aplicativos Nativos vs Híbridos

Fonte: Ádames (2016)

Como destacado na figura 2, o aplicativo híbrido exige menos conhecimento técnico do programador para seu desenvolvimento, uma vez que utiliza sempre programação web independentemente da plataforma em questão.

De acordo com Total Cross (2016), a grande vantagem é exigir apenas conhecimento de desenvolvimento web e, portanto, apresentar custo menor de desenvolvimento.

Ele se destaca também no quesito tempo de produção, pois é construído muito mais rápido e uma única vez, diferentemente dos nativos. Traz maior flexibilidade, no sentido de atender as principais plataformas de mercado, e esse com certeza é seu maior atrativo. Outra grande vantagem do aplicativo híbrido é a facilidade para se disponibilizar futuras atualizações.

O Aplicativo híbrido é mais adequado, porque parte do seu código poderá estar online e ser atualizado pelo sistema web dentro do aplicativo, sem precisar atualizar o aplicativo todo ou enviar novas versões para as lojas (GOUVÊIA, 2015).

Quando se trata de um aplicativo teste, também fica em primeiro lugar a utilização de um aplicativo mobile híbrido, pois, como se trata de um teste, se torna mais viável um investimento menor, tanto em tempo quanto em custos, para analisar a reação do público e dependendo dela, investir mais no aplicativo. Neste contexto, Gouveia (2015) indica afirma que “não gaste todas suas moedas em um aplicativo para ver no que vai dar”. Essa afirmação defende a ideia de que não vale a pena desenvolver um aplicativo teste em linguagem nativa quando não se sabe o nível de aceitação dos usuários para um novo Aplicativo. A melhor opção neste caso é desenvolvê-lo como aplicativo híbrido, e caso receba aceitação no mercado, é possível inclusive construí-lo como nativo (quando o foco também é desempenho).

Esse conjunto de vantagens apresenta uma atratividade interessante associada ao desenvolvimento de um aplicativo híbrido no competitivo mercado atual, o baixo custo no desenvolvimento. Essa, sem dúvida é a maior vantagem desse tipo de aplicativo, pois o que as empresas mais buscam hoje é a redução de custos.

Comparado a outras modalidades de aplicativos, os nativos possuem um custo de desenvolvimento muito maior, uma vez que necessitam de desenvolvedores com conhecimentos específicos para cada plataforma.

Mesmo com todas essas vantagens, o mundo dos aplicativos híbridos é perfeito. Como demonstrado na figura 2, uma das maiores desvantagens do aplicativo híbrido é seu desempenho. Quando um aplicativo exige muito da capacidade de processamento de um dispositivo móvel as aplicações nativas ficam a frente.

Devido ao uso de APIs para acessar os recursos nativos dos dispositivos, ele se torna mais lento. Mas essa diferença só é perceptível ao usuário quando é utilizado um alto índice de processamento.

Outra de suas desvantagens é não ter acesso a todos os recursos nativos do dispositivo. São eles: execução em segundo plano, notificações do sistema operacional, informações adicionais do acelerômetro (além da detecção dos eixos de coordenadas nas direções vertical e horizontal) e gestos complexos. Isso também vale para componentes visuais, ou seja, os componentes de tela responsáveis pelo layout (parte gráfica) do aplicativo, por causa disso, os aplicativos híbridos não seguem o padrão de telas conhecidos pelos usuários de aplicativos nativos, ou seja, existe uma variação considerando a experiência de usuário e usabilidade.

Figura 3 – Tabela de comparação Aplicativos Híbrido vs Nativo

	Híbrido	Nativo (android/iOS)	Melhor
Gráfico	HTML, Canvas, SVG	APIs Nativas	Nativo
Performance	Lenta	Rápida	Nativo
Aparência "natural"	Emulado	Aparência real	Nativo
Recursos do equipamento	Muita	Total	Nativo
Publicação nas lojas	Quase normal	Normal	Nativo
Reutilização do código	Total	Nenhuma	Híbrido
Custo de desenvolvimento	Médio	Alto	Híbrido
Tempo de desenvolvimento	Baixo	Alto	Híbrido
Facilidade de atualização	Fácil	Médio	Híbrido
Conhecimentos requeridos	HTML5, CSS e Javascript	Java, ObjectiveC e Swift	
Curva de aprendizado	Média	Lenta	Híbrido

Fonte: Gouveia (2015)

Depois de apresentados os conceitos de desenvolvimento de aplicativos multiplataformas e das suas principais vantagens e desvantagens, é interessante analisar e comparar uma aplicação híbrida com uma aplicação nativa, para que se possa escolher a melhor opção dependendo da sua necessidade.

De acordo com a figura 3, é possível fazer a comparação dos principais itens levados em consideração o desenvolvimento de um aplicativo mobile. Analisando os itens acima, se um aplicativo híbrido atender as suas necessidades, então ele é uma ótima opção para seu desenvolvimento, mas caso seja algo muito especifico e ele não atenda às suas necessidades então a opção seria o desenvolvimento de um aplicativo nativo.

3. CONSIDERAÇÕES FINAIS

Este trabalho objetivou conceituar as estratégias de desenvolvimento de aplicativos multiplataforma e identificar as principais vantagens e desvantagens, deste tipo de desenvolvimento. Foi realizada uma pesquisa bibliográfica e análise de artigos científicos sobre o tema em questão visando compreender melhor os aspectos envolvendo desenvolvimento de aplicativos Mobile que funcione em diversas plataformas sem a necessidade de codificação adicional.

Os resultados da pesquisa desse artigo apontam que o Cordova, é um dos principais frameworks utilizados como ferramenta de base para o desenvolvimento de aplicativos mobile multiplataforma. O emprego desse *framework* permite que aplicativos Mobile funcionem nas principais plataformas, como por exemplo, Android, iOS e Windows Phone.

As principais vantagens da utilização de aplicativos híbridos podem ser resumidas nos seguintes aspectos: exige menos conhecimento técnico; desta forma é necessário menor tempo de aprendizado da linguagem de programação utilizada no desenvolvimento (Desenvolvimento web); é flexível; atendendo as principais plataformas do mercado utilizando o mesmo código fonte; facilidade de se disponibilizar futuras atualizações e ideal

para a construção de aplicativos testes (protótipos). Essas vantagens resultam em baixo custo de desenvolvimento.

As principais desvantagens apresentadas são: baixo desempenho ao se exigir maior esforço computacional; não apresenta acesso a todos os recursos nativos do dispositivo, como por exemplo, execução em segundo plano e notificações do sistema operacional. Um aspecto desejado também é o acesso a um número maior de componentes nativos para construção de telas, já que os recurso fornecidos pelo framework Cordova não seguem um padrão de telas familiar par os usuários de aplicativos nativos, o que acaba por comprometer de alguma forma a usabilidade.

Apesar disso é possível afirmar que o desenvolvimento de aplicativos multiplataforma, possui um grande aliado, o Cordova, que facilitou o desenvolvimento de aplicativos Mobile para diversas plataformas. Dessa forma é possível fornecer aplicativos Mobile para um número muito maior de usuários, já que essa abordagem permite o uso de múltiplas plataformas.

Por fim, conclui-se que, primeiramente deve ser realizada uma análise de todas as necessidades associadas ao aplicativo antes de decidir qual a melhor opção para o desenvolvimento: um aplicativo híbrido ou um aplicativo nativo.

REFERÊNCIAS

ISO/IEC. **International Standard ISO/IEC 9126-1 - Information technology - Software product quality - Part 1: Quality model**. 2000.

ÁDAMES, Alexandre. **Aplicativos Nativos vs Híbridos.** Disponível em <http://guiapraticoapp.com.br/aplicativos-nativos-vs-hibridos/> Acesso em 24 de julho de 2016.

ANDRADE, Tirso. **Opa ! NativeScript ! Outra linguagem para aprender?** Disponível em <https://www.linkedin.com/pulse/opa-nativescript-outra-linguagem-para-aprender-tirso-andrade> Acesso em 24 de julho de 2016.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). **NBR 20121:** informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2016.

AVANTI! TECNOLOGIA & MARKETING. **Aplicativo nativo ou aplicativo híbrido: qual a melhor solução?.** 2015. Disponível em <http://blog.penseavanti.com.br/aplicativo-nativo-ou-aplicativo-hibrido-qual-a-melhor-solucao/> Acesso em 24 de julho de 2016.

CORDOVA. **Overview.** Disponível em <https://cordova.apache.org/docs/en/latest/guide/overview> Acesso em 24 de julho de 2016.

GOUVÊA, Tiago. **Aplicativos mobile híbridos e nativos – qual a diferença?.** 2015. Disponível em <http://www.tiagogouvea.com.br/diferenca-entre-aplicativos-mobile-hibridos-e-nativos>. Acesso em 24 de julho de 2016.

IBM. **Desenvolvimento de apps – Parte 2: híbrido, nativo ou web?.** 2013. Disponível em <https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/desenvolvimento_de_apps-parte_2_hibrido_nativo_ou_web?lang=en>. Acesso em 24 de julho de 2016.

Artigo I. LOPES, Sérgio. **Aplicações mobile híbridas com Cordova e PhoneGap.** Série Caelum. Editora: Casa do Código. 1. ed. São Paulo, 2016. Pág. 04 a 07.

Artigo II. LOPES, Sérgio. **A Web Mobile**. Editora: Casa do Código. 2. ed. São Paulo, 2015.

ICROSOFT. **Instalar o Visual Studio Tools for Apache Cordova.** Disponível em <https://msdn.microsoft.com/pt-br/library/dn757054.aspx> Acesso em 24 de julho de 2016.

NUNES, Flávio. **Desenvolvendo aplicativos móveis multipltaforma**. 2013. Disponível em <http://imasters.com.br/desenvolvimento/desenvolvendo-aplicativos-moveis-multiplataforma/> Acesso em 14 de outubro de 2015.

SILVA, Patrícia Gomes dos Santos. A**plicativos móveis híbridos com ionic framework.** Disponível em <http://www.matera.com/br/2016/03/08/aplicativos-moveis-hibridos-com-ionic-framework/> Acesso em 24 de julho de 2016.

TOTALCROSS. **Conheça as diferenças entre aplicativos nativos, mobile e híbridos.** Disponível em <http://www.totalcross.com/blog/conheca-as-diferencas-entre-aplicativos-nativos-mobile-e-hibridos/> Acesso em 24 de julho de 2016.