

# Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas

Ezequiel Douglas Prezotto<sup>1</sup>, Bruno Batista Boniati<sup>1</sup>

<sup>1</sup>Tecnologia em Sistemas para Internet - Universidade Federal de Santa Maria(UFSM)  
Frederico Westphalen – RS – Brasil

prezotto.ezequiel5@gmail.com, bruno@cafw.ufsm.br

**Resumo.** Com o expressivo crescimento dos dispositivos móveis, suas variedades de modelos, fabricantes e sistemas operacionais aumentaram muito. Havendo a necessidade de se disponibilizar um software que atenda todas (ou mais de uma) plataforma, será necessário desenvolver uma aplicação específica para cada uma delas. Tal requisito eleva muito os custos e o tempo do desenvolvimento. Neste contexto surgem as aplicações híbridas, que permitem que uma mesma aplicação funcione em diferentes plataformas. Objetiva-se estudar e analisar os frameworks para desenvolvimento de aplicações híbridas, bem como características e desempenho.

**Abstract.** The significant growth of mobile devices, their varieties of models, manufacturers and operating systems has greatly increased. With the need for software that attends all platforms available, it will be necessary to develop a specific application for each of them. This requirement raises a lot the cost and the time spend on development. In this contest arise hybrid applications, which allow the same application running on different platforms. Objective is to study and analyze frameworks for developing hybrid applications as well as features and performance.

## 1. Introdução

Com o expressivo crescimento das tecnologias móveis, experimentado ao longo dos últimos anos, observa-se que a diversidade de tais equipamentos aumentou muito. Consequentemente o número de plataformas e ambientes/linguagens de programação para desenvolvimento de aplicações também foi incrementado. Tal situação encarece e complexifica o desenvolvimento de uma aplicação que venha a atender a todos dispositivos, pois é necessário conhecimentos específicos de cada plataforma, e também que seja desenvolvida uma aplicação diferente para cada plataforma.

Neste contexto surgem as aplicações híbridas, em que se desenvolve apenas uma aplicação, e esta pode ser utilizada em vários dispositivos com diferentes sistemas operacionais. Isso é possível graças aos *frameworks* para desenvolvimento de aplicações híbridas, que são responsáveis por empacotar o código-fonte para as diferentes plataformas, permitindo que elas sejam instaladas no dispositivo, e possam acessar os seus recursos, como câmera, GPS e contatos.

Este trabalho visa demonstrar características da programação de aplicações híbridas os seus pontos fortes e fracos, além de destacar os principais *frameworks* para desenvolvimento de aplicações híbridas, que habilitam acesso a recursos nativos, bem como os *frameworks* de UI (*User Interface* - interface do usuário).

Para melhor compreender o funcionamento dos *frameworks* para desenvolvimento e das aplicações híbridas, foi desenvolvido um estudo de caso. Neste estudo, uma aplicação (diário de classe escolar) foi desenvolvida, utilizando-se de *frameworks* para aplicações híbridas e disponibilizada em diferentes plataformas. Entende-se que além de servir como meio de análise do estudo, a aplicação também resulta em um software útil em meio acadêmico.

## 2. Dispositivos e Plataformas

Dispositivo móvel é todo aquele equipamento, que pode ser levado a qualquer lugar. Quanto menos dependente de características físicas, maior será o grau de mobilidade. Por exemplo, a bateria: quanto maior a duração e menor o tempo de recarga maior a mobilidade provida pelo dispositivo [MORIMOTO 2009].

São considerados dispositivos móveis os celulares, *tablets* e outros dispositivos do gênero que ofereçam mobilidade e autonomia. O presente trabalho não considera *laptops* (*notebooks* ou *netbooks*) como dispositivos móveis já que ao contrário de celulares, o usuário não os mantém o tempo todo funcionando e utilizando, apenas o transporta para o local onde irá utilizá-lo.

Hoje existem muitas plataformas para desenvolvimento móvel, como Android, iOS, Windows Phone, Firefox OS, BlackBerry, Ubuntu Touch, Fire OS, entre outros. Cada plataforma possui diferentes formas e linguagens para desenvolvimento, como por exemplo o Android utiliza Java como linguagem de programação, Firefox OS utiliza linguagens web (HTML, CSS, JavaScript), iOS utiliza Objective C.

Cabe ao desenvolvedor ou a equipe de desenvolvimento optar pela plataforma e pela quantidade de sistemas diferentes que a aplicação deve dar suporte e assim escolher a forma de desenvolvimento que melhor vai se adequar. Ao fazer esta análise deve-se levar em conta, os conhecimentos dos desenvolvedores, a quantidade de plataformas que deverão ser suportadas, o tempo para desenvolvimento, e a quantidade de recursos que a aplicação vai consumir.

## 3. Tipos de Aplicações

Além das aplicações híbridas que são o foco do trabalho também serão demonstradas as aplicações nativas e as Web Apps, para que se possa observar as diferenças entre as mesmas, e identificar em quais situações determinado tipo de aplicação melhor se adequa.

### 3.1 Aplicações Nativas

As aplicações nativas são as que foram desenvolvidas especialmente para aquela plataforma e somente para ela, utilizando para isso as ferramentas disponibilizadas para a mesma, como a linguagem de desenvolvimento, ambiente e emulador. A tabela 1 demonstra as principais plataformas existentes bem como suas e características específicas.

**Tabela 1. Listagem de plataformas e suas características**

Plataforma	IDE	Loja de App	Linguagem
Android	Eclipse	Google play	Java
IOS	XCODE	Apple Store	Objective-C
Windows Phone	Visual Studio	WindowsPhone store	C#
Firefox OS	N/D	Mozilla Marketplace	Javascript,CSS,HTML
BlackBerry	Momentics	BlackBerry World	C+
Web OS	WebOS3.0 SDK	WebOS Nation	Javascript C/C++
Ubuntu Touch	Ubuntu SDK		HTML5, Javascript,CSS
Bada	Bada IDE	Bada Brasil	C++
Symbian	Symbian SDK		C++
Tizen	Tizen SDK		C++,HTML5 e Java
Fire os	SDK próprio		C, C++, Java

Aplicativos desenvolvidos na linguagem nativa do sistema tem um melhor desempenho, apesar das outras tecnologias de desenvolvimento terem evoluído bastante quase que se equiparando. Um ambiente já preparado para o desenvolvimento nativo, agiliza muito o processo, que é insignificante se for preciso desenvolver para inúmeros sistemas diferentes, pois a cada sistema é necessário se desenvolver uma aplicação específica, diferindo assim das outras formas de desenvolvimento mostradas a seguir.

### 3.2 Web App

Além das aplicações nativas de cada plataforma existe também as Web Apps. Basicamente são aplicações Web que rodam no *browser* do dispositivo, dessa forma não é necessário se preocupar com as diferentes linguagens, pois ela utiliza padrões web como HTML, CSS e JavaScript para o lado do cliente e PHP, Java ou outras linguagens para o lado servidor.

Mesmo se utilizando a mesma linguagem para todos os sistemas operacionais, estas aplicações podem se comportar de maneiras distintas, dependendo se o *browser* do dispositivo suporta ou não determinado componente da aplicação, ou mesmo dependendo da forma como o *browser* realiza o processamento gráfico da aplicação [W3C 2014].

Para estas aplicações não é necessário que a mesma seja desenvolvida especificamente para dispositivos móveis, pois apenas tendo *layout* responsivo, é possível fazer com que a mesma aplicação que foi desenvolvida para *desktop* seja utilizada em um *smartphone*. Existem aquelas que foram desenvolvidas pensando especificamente em dispositivos móveis, que podem ser melhor otimizadas para os mesmos.

### 3.3 Híbridas

Estas são aplicações que possuem como finalidade funcionar em qualquer que seja o dispositivo, sendo que para as diferentes plataformas, será utilizado o mesmo código-fonte. Aplicações híbridas ficam instaladas no dispositivo e podem funcionar independentemente de se ter ou não conexão com internet. Partindo do princípio de

utilizar uma mesma aplicação para diferentes plataformas, normalmente as aplicações híbridas são desenvolvidas em linguagens Web que são interpretadas pelo *browser* nativo do sistema. Possuem tanto características de aplicações nativas como de web apps.

Apesar de serem feitas em linguagens web estas aplicações não são Web App, elas ficam instaladas no dispositivo e quando são acionadas chamam o *browser*, onde serão executadas, independente de estarem conectadas à internet. Como são executadas no *browser* e este está sujeito à padronização pela W3C, as aplicações que são desenvolvidas e testadas em uma plataforma devem funcionar da mesma forma nas outras.

Através do estudo destas diferentes formas de desenvolvimento de aplicações móveis, foi elaborado um comparativo representado pela figura 1, demonstrando a quais quesitos cada tipo de aplicação oferece suporte. Aplicações nativas tem acesso a todos recursos do aparelho como câmera, GPS, contatos, por exemplo. Da mesma forma as aplicações híbridas que graças ao *framework* é possível acessar estes recursos, já as web app tem acesso limitado.

	Acesso ao Dispositivo	Performance	Tempo Desenvolvimento	App Store	Multi Plataforma
Nativo	Sim	Sim	Caro	Sim	Não
Web	Parcial	Sim*	Ótimo	Não	Sim
Híbrido	Sim	Sim*	Ótimo*	Sim	Sim*

**Figura 1. Comparativo dos diferentes tipos de aplicações**

Aplicações nativas e híbridas podem ser disponibilizadas através das lojas de aplicativos, uma vez que elas ficam instaladas no aparelho, ao contrário de uma web app que nada mais é que um site para dispositivos móveis. Apenas as aplicações nativas é que são desenvolvidas para plataformas específicas, as web app e aplicações híbridas em geral independem da plataforma.

#### 4. Frameworks

Os *frameworks* para desenvolvimento de aplicações híbridas separam-se em dois tipos, os responsáveis pela interface (UI – *User Interface*), e aqueles que empacotam a aplicação para as diferentes plataformas e permitem que ela acesse os recursos do aparelho.

##### 4.1 Frameworks de UI

Os *frameworks* de UI, são os responsáveis por definir a aparência que vai ter a aplicação, sendo ela uma aplicação nativa, Web App ou híbrida, todas tem uma interface com o usuário. A parte visual da aplicação é aquela que dá a primeira impressão e que pode causar maior aceitação ou maior rejeição da aplicação. Um aplicativo com interface amigável e fácil de usar, tem maiores chances de ganhar a preferência dos usuários se comparado a um que, mesmo tendo a mesma funcionalidade

ou até outras, apresenta-se com uma interface difícil de usar e pouco intuitiva [MANZOTTI, 2013].

Segundo Prates e Barbosa (2010), o design de interfaces em sistemas interativos é uma tarefa tão relevante que se tornou uma das subáreas da interação humano-computador (IHC) que, por sua vez, visa estudar, planejar e entender como as pessoas e dispositivos computacionais podem interagir de forma que as necessidades delas sejam contempladas da forma mais efetiva possível.

Para aplicações nativas existe certa padronização das interfaces, já que elas obedecem as regras da plataforma, já para as web app e híbridas, podem ser utilizados os *frameworks* para desenvolvimento web, como jQuery ou Bootstrap entre outros. Não é necessário que o *framework* seja específico para dispositivos móveis, embora um específico poderia proporcionar melhor experiência, tanto em e performance quanto em usabilidade.

#### 4.1 *Frameworks* para Acesso a Recursos Nativos

Segundo Ribeiro e Freire (2013), definimos um *framework* multiplataforma como um conjunto de arquivos de código-fonte, bibliotecas e ferramentas que oferece suporte a mais de uma plataforma, pelo menos duas plataformas diferentes. Nesta definição, iOS e Android são duas plataformas diferentes, iOS 5 e iOS 6 não são, e permite o desenvolvimento sem ramificações do código-fonte, ou seja o mesmo código para todas plataformas [RIBEIRO; FREIRE 2013].

Desta forma os *frameworks* devem possuir uma linguagem unificadora, ou seja, a mesma para todas as plataformas, APIs que permitam que se possa acessar da mesma forma um recurso em um Android ou em um Firefox OS, e que possibilite acesso da aplicação aos recursos nativos do sistema. Durante o estudo foram encontrados vários *frameworks*, como o Titanium [TITANIUM, 2014], o Trigger.io [TRIGGER, 2014], o Adobe Phonegap [ADOBE, 2014] e o Apache Cordova [APACHE, 2014].

Após analisar as formas de desenvolvimento de cada plataforma, sua linguagem e outras características optou-se, para o estudo de caso, o *framework* do Apache Cordova. A decisão baseou-se no fato de o mesmo utilizar linguagens web como HTML, CSS e JavaScript, além disso possuir um maior número de plataformas suportadas, por possuir maior popularidade (tendo um maior número de desenvolvedores), além de boa documentação. O Apache Cordova teve origem do *framework* Phonegap, que foi adquirido pela Adobe e posteriormente doado à fundação Apache sendo chamado assim de Cordova, a Adobe deu continuidade ao projeto Phonegap, adicionando outros produtos como o Phonegap Build.

O *framework* faz a ponte entre a aplicação e o dispositivo, através de suas APIs, também com ele é possível empacotar a aplicação, ou gerar o executável para cada plataforma diferente. Ele é compatível com todos os sistemas operacionais e por não possuir interface gráfica, todas as operações são feitas por linha de comando:

- Criar projeto: *cordova create nomeDaPasta br.com.meu.projeto "nome do projeto"*.
- Adicionar plataforma: *cordova platform add android*(ou iOS, wp7, wp8, firefoxs).
- Adicionar plugins: *cordova plugin add org.apache.cordova.geolocation*.
- Compilar: *cordova build*.

## 5. Estudo de Caso

De forma a validar as ideias apresentadas por este trabalho, pensou-se em desenvolver uma aplicação útil, de objetivo prático e funcional, que permite fazer o maior número possível de testes, para que se possa revelar quais seriam os pontos fortes e fracos das aplicações híbridas e dos respectivos *frameworks* utilizados. Considerando tais requisitos optou-se pelo desenvolvimento de uma aplicação para controle de presenças na sala de aula.

A aplicação desenvolvida independe de conexão com internet, já que utiliza banco de dados local e fica instalada no dispositivo, permitindo assim que usuários possam utilizá-la em locais desprovidos de conexão com a internet. Também se buscou desenvolver uma aplicação fácil e prática de se utilizar, com bom desempenho e boa condução ao usuário. Para comprovar o funcionamento das aplicações híbridas, ela também deve ser testada em mais de uma plataforma, e acessar recursos nativos do aparelho.

Foram utilizadas para desenvolvimento as linguagens HTML, CSS e JavaScript, e como editor de código o SublimeText, por se tratarem linguagens web, os testes de interface puderam ser realizados diretamente no *browser*. Já testes a outros recursos como câmera foram feitos diretamente em um dispositivo Android, e os teste multiplataforma em emuladores como do Firefox OS, Windows Phone e BlackBerry.

Por se tratarem muitas vezes de dispositivos com pouca capacidade de armazenamento e processamento, foi estruturada uma base de dados simples, que se encarregue de responder a todas as requisições de forma rápida, para isso foi utilizado o Web SQL [WEBSQL, 2014], além disso, para armazenamento de informações adicionais, foi utilizado o LocalStorage [LOCALSTORAGE, 2014], principalmente para variáveis de controle.

Na aplicação desenvolvida é possível cadastrar alunos e turmas, para em seguida criar uma aula e marcar os alunos presentes. No cadastro do alunos é possível tirar uma foto do mesmo para uma melhor identificação durante o processo da chamada. Para obter acesso às funcionalidades da aplicação o professor deve primeiramente fazer o *login* na aplicação, este acesso ficara disponível a ele até que seja feito o *logout*.

Os testes da aplicação foram feitos em dispositivos Android e no emulador dessa plataforma, além disso, também foram feitos testes nos emuladores do Firefox OS e Windows Phone 7, para assim demonstrar o comportamento da aplicação nestas diferentes plataformas.

Para admitir um resultado mais preciso, os testes realizados durante o desenvolvimento da aplicação, foram feitos em plataforma Android, e após concluído o desenvolvimento, foi então feito o teste nas demais plataformas. Durante os testes foi percebido que a interface da aplicação sofre poucas mudanças nas diferentes plataformas. As mesmas mudanças que sofreriam as aplicações web, por exemplo, em diferentes *browsers*.

Foi notado que cada plataforma, interpreta os componentes da aplicação de formas distintas, como um campo data que no Firefox OS é exibido uma espécie de calendário para selecioná-la, da mesma forma que o Android, já no Windows Phone 7 este campo é interpretado como uma campo de texto normal obrigando ao usuário a digitar a data. Pode-se observar na figura 2 um exemplo disso, em uma das telas do aplicativo desenvolvido, existe um campo de data, que foi interpretado de formas distintas, nas diferentes plataformas.



**Figura 2. Comparativo da Aplicação em Diferentes Plataformas**

Além da interface, outros componentes da aplicação também se comportaram de formas distintas nas diferentes plataformas, como por exemplo a base de dados que utiliza Web SQL, a plataforma Windows Phone 7 não oferece suporte. Nestes casos deve se pensar em opções alternativas como o IndexedDB [INDEXEDDB, 2014], que são suportadas por um maior número de plataformas. Por se tratar de uma aplicação de testes, a utilização do Web SQL dentre outros componentes foi proposital.

## 6. Conclusões

Ao longo do trabalho buscou-se demonstrar de forma imparcial, dando enfoque nas aplicações híbridas, os diferentes tipos de desenvolvimento de aplicações para dispositivos móveis, relatando as características de cada tipo de desenvolvimento. Destaca-se que não existe uma opção melhor que a outra, mas sim que situações onde uma opção se adapta melhor do que as outras, cabendo a desenvolvedor decidir qual melhor se enquadraria em sua necessidade.

Pode-se concluir com base no estudo que comprovadamente as aplicações híbridas, podem através de um mesmo código fonte originar aplicações para diferentes plataformas. Também se pode constatar que uma aplicação híbrida possui acesso a todos recursos do dispositivo, e pode também ser disponibilizadas juntamente com as aplicações nativas, nas lojas de aplicativos. Porém também pode-se identificar que elas não se comportam da mesma forma, por isso é muito importante fazer uma análise dos requisitos, para identificar se a plataforma dará ou não suporte a eles.

Como a aplicação não tem fins lucrativos, todos os códigos-fonte serão disponibilizados, a partir do Github. Além disso, pretende-se publicar a aplicação nas lojas oficiais do maior número de plataformas possíveis, para possibilitar o acesso a ela por todos, além de dar continuidade melhorando-a e adicionando novas funcionalidades.

## 7. Referências

ADOBE. Disponível em: <http://phonegap.com/>, Acessado em: Maio/2014.

APACHE. Disponível em: <http://cordova.apache.org/>, Acessado em: Maio/2014.

INDEXEDDB. Disponível em <http://www.w3.org/TR/IndexedDB/>, Acessado em: Agosto /2014.

LOCALSTORAGE. W3C. Web Storage. Disponível em: <http://www.w3.org/TR/webstorage/>, Acessado em: Maio/2014.

- MANZOTTI, C. S. Design de Interface em Dispositivos Móveis. 2013.
- MORIMOTO, C. E. Smartphones Guia Prático. [S.l.]: Sul Editores, 2009.
- PRATES, R. O.; BARBOSA, S. D. J. Avaliação de Interfaces de Usuário: conceitos e métodos. 2010.
- RIBEIRO, R.; FREIRE, P. Frameworks de Desenvolvimento Móvel Multiplataforma. In: CONFERÊNCIA DA ASSOCIAÇÃO PORTUGUESA DE SISTEMAS DE INFORMAÇÃO, 13., 2013. Anais. . . [S.l.: s.n.], 2013.
- TITANIUM. Mobile Development Environment. Disponível em: <http://www.appcelerator.com/titanium/>, Acessado em: Maio/2014.
- TRIGGER. The Simplest Way to Build Amazing Mobile Apps. Disponível em: <http://trigger.io>, Acessado em: Maio/2014.
- W3C. W3C Brasil. Disponível em: <http://www.w3c.br/>, Acessado em: Maio/2014.
- WEBSQL. W3C. Web SQL Database. Disponível em: <http://www.w3.org/TR/webdatabase/>, Acessado em: Maio/2014.