

# Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares

Marcelo Moro da Silva, Marilde Terezinha Prado Santos

**Resumo:** A popularização dos Smartphones desencadeou uma revolução no mercado de telefonia móvel. Este artigo objetiva apresentar as características e os paradigmas de desenvolvimento de aplicativos para aparelhos celulares. São ilustrados como as características do aplicativo para aparelho celular podem auxiliar na definição do melhor tipo de paradigma de desenvolvimento, tendo como estudo de caso o sistema GEPAT - Sistema de Gestão Patrimonial.

**Palavras Chave:** dispositivos móveis, características do aparelho celular, desenvolvimento de software

## *Application Development paradigms for cell phones*

**Abstract:** *The popularization of smartphones has triggered a revolution in the mobile phone market. This article aims at presenting the features and paradigms when developing applications for mobile devices. We illustrate how the app features for mobile devices can help defining the best kind of development paradigm presenting the GEPAT system - Asset Management System - as a case study.*

**Keywords:** *mobile devices, cell phone features, software development*

## I. INTRODUÇÃO

A evolução da tecnologia dos aparelhos celulares permitiu oferecer ao usuário recursos que vão muito além da realização de uma chamada ou do envio de uma mensagem. As melhorias de hardware dos aparelhos celulares permitiram o desenvolvimento de sistemas operacionais mais avançados. Com sistemas operacionais mais avançados foi possível desenvolver aplicativos melhores, com cada vez mais recursos e serviços ao usuário. Os acessos a serviços de instituições financeiras e redes sociais, por exemplo, podem ser facilitados pelo uso de aplicativos que são executados em aparelho celular. Devido a esta evolução, um aparelho celular se transformou em uma oportunidade de entretenimento, acesso a informação e solução de problemas, integrando-se assim ao cotidiano das pessoas e facilitando diversas tarefas do dia a dia.

Motivado por estes resultados, o mercado de aplicativos para aparelhos celulares mostra-se em crescimento rápido. Segundo Martins et al. (2013) este mercado é disputado por diferentes plataformas tecnológicas, incluindo sistemas operacionais e plataformas de desenvolvimento, gerando uma variedade de soluções no mercado. Frente a este fato, um dos principais desafios deste mercado é desenvolver aplicativos em um ambiente de tecnologia altamente fragmentado e em rápida evolução.

Limitações de plataforma para distribuição do aplicativo,

tempo e custo para o desenvolvimento, complexidade das tecnologias necessárias para a sua criação e manutenção são pontos problemáticos em um projeto voltado ao desenvolvimento deste tipo de aplicativo.

Neste artigo apresentamos informações sobre os tipos de paradigmas de desenvolvimento de aplicativos para aparelhos celulares. É discutida também a importância de acertar na escolha do tipo ideal de paradigma, para garantir a construção de um aplicativo com qualidade, utilizando as ferramentas apropriadas.

Este artigo está organizado da seguinte forma: na seção II será abordado o desenvolvimento de aplicativos nativos para aparelhos celulares; na seção III será abordado o desenvolvimento de aplicativos multiplataforma para aparelhos celulares e as diversas abordagens existentes; na seção IV será discutida a existência de um paradigma de desenvolvimento ideal; Na seção V é realizado um estudo de caso, apresentando o sistema GEPAT (Sistema de Gestão Patrimonial); Na seção VI serão apresentados os trabalhos relacionados a este estudo e na seção VII será apresentada a conclusão e a indicação para trabalhos futuros.

## II. DESENVOLVIMENTO DE APLICATIVO NATIVO PARA APARELHOS CELULARES

Conceituando aplicativos nativos destinados a serem executados em aparelhos celulares, podemos defini-los como

aplicativos que foram desenvolvidos para um tipo específico de plataforma. Essas plataformas são compostas de diversas tecnologias, tais como: sistema operacional, linguagens de programação e IDEs (Integrated Development Environment). O sistema operacional é responsável por gerenciar diversos recursos do aparelho celular, as linguagens de programação são utilizadas na programação do aplicativo e o IDE, também conhecido como ambiente de desenvolvimento integrado, fornece ferramentas que auxiliam na criação do aplicativo.

Atualmente, há diversas plataformas para aparelhos celulares, tais como: Android (Google), IOS (Apple Inc), Windows Mobile (Microsoft Corp), entre outros. Cada um destes sistemas operacionais necessita que o aplicativo nativo seja construído utilizando uma linguagem de programação específica como, por exemplo, Java para Android, Objective-C para IOS e C++ ou C# para Windows. (WHITE, 2013).

Através das APIs disponíveis para cada linguagem utilizada em cada plataforma, um aplicativo nativo pode acessar funcionalidades oferecidas por recursos nativos do sistema operacional, tais como GPS, banco de dados, SMS, email, gerenciador de arquivos, entre outros. Esses aplicativos

podem ser baixados, instalados e vendidos em lojas de aplicativos (SAMBASIVAN et al., 2011).

O desenvolvimento de um aplicativo nativo para aparelhos celulares exige conhecimentos específicos a respeito das tecnologias utilizadas pela plataforma na qual se deseja executar-lo.

A principal desvantagem de um aplicativo nativo está no fato de ser executado apenas na plataforma para a qual foi desenvolvido, aumentando o tempo, custo e o esforço para disponibilizar um mesmo aplicativo para mais de uma plataforma (SAMBASIVAN et al., 2011).

As diferenças entre as plataformas também são importantes, pois para cada uma há diferentes ferramentas de desenvolvimento, APIs e equipamentos com diferentes capacidades. Desenvolver aplicativos nativos para diversos sistemas operacionais requer uma equipe de trabalho com conhecimento em várias tecnologias (CHARLAND e LEROUX, 2011). A tabela 1 apresenta conhecimentos necessários para desenvolver aplicativos nativos para nove plataformas diferentes:

Tabela 1. Conhecimentos necessários para desenvolver aplicativos para nove plataformas.  
Adaptado de (CHARLAND e LEROUX, 2011)

CONHECIMENTOS NECESSÁRIOS PARA DESENVOLVER APLICATIVOS PARA NOVE PLATAFORMAS	
PLATAFORMA	CONHECIMENTOS NECESSÁRIOS
Apple IOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Windows 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Aplicativos nativos são desenvolvidos de acordo com um conjunto de especificações fornecidas pelo fabricante do sistema operacional. Esses fabricantes fornecem APIs que podem ser utilizadas pelos desenvolvedores, garantindo certo nível de consistência entre todos os aplicativos nativos. (WHITE, 2013). Oferecem ótima experiência de uso, pois possuem acesso a mais recursos do aparelho celular, tais como câmera, agenda de contatos, GPS, entre outros e, portanto, permitem ao usuário uma navegação mais imersiva. Com relação a sua interface, os componentes de um aplicativo nativo são iguais aos do sistema operacional utilizado pelo sistema operacional do aparelho celular, portanto é mais intuitivo ao usuário.

Aplicativos desenvolvidos através do paradigma nativo

possuem ótimo desempenho, tendo em vista não ser necessário qualquer interpretação de código durante a execução.

Outro aspecto importante dos aplicativos nativos é a independência de comunidades responsáveis por versões de frameworks de desenvolvimento de aplicativos para celular, pois em um aplicativo nativo as atualizações de API do próprio sistema operacional do aparelho celular serão utilizadas no desenvolvimento de novas funcionalidades do aplicativo nativo, portanto a partir do momento que a plataforma é atualizada, já é possível incorporar as novidades existentes da plataforma para desenvolver para o aplicativo nativo, sem a dependência da atualização de tecnologias adjacentes, como por exemplo, algum framework particular.

### III. DESENVOLVIMENTO DE APLICATIVO MULTIPLATAFORMA PARA APARELHOS CELULARES

Considerando um cenário com notável diferença entre sistemas operacionais e plataformas de programação existentes, desenvolver um aplicativo para aparelho celular uma vez e poder utilizá-lo em varias plataformas é um dos maiores desafios da computação móvel.

Diversas ferramentas de desenvolvimento (frameworks) de aplicativos multiplataforma para aparelhos celulares foram desenvolvidas com abordagens, propósitos e características distintas. De acordo com Palmieri et al. (2012), estas ferramentas trouxeram os seguintes benefícios: redução da complexidade; redução de código; redução do tempo de desenvolvimento e custo de manutenção; diminuição de conhecimento necessários sobre a API; maior facilidade no desenvolvimento e aumento de participação de mercado.

Dentre os pontos importantes a serem considerados na escolha de uma boa ferramenta de desenvolvimento de aplicativos multiplataforma para celular. Palmieri et al. (2012) destacam as seguintes características: quantidade de sistemas operacionais suportados, licença que apresente de forma clara os termos e condições para uso, linguagem de programação utilizada pela ferramenta de desenvolvimento, API com acesso a muitos recursos nativos do aparelho, arquitetura e integração a ambientes de desenvolvimento.

Palmieri et al. (2012) sugerem que com o crescimento dos aplicativos multiplataforma, o mercado de comercialização de aplicativos para aparelhos celulares sofrerá alterações importantes em seu modelo de negócio. Entre as vantagens destaca-se maior retorno de ROI (Retorno sobre

Investimento), pois uma vez que o aplicativo é multiplataforma e pode ser empacotado como um aplicativo nativo, este é desenvolvido uma única vez e poderá ser comercializado em várias lojas de aplicativos (app stores).

Hartmann et al. (2011) divide as ferramentas de desenvolvimento nos seguintes grupos:

- **Biblioteca:** Pequeno kit de ferramentas que oferece funcionalidades muito específicas ao usuário. Normalmente utilizada em conjunto com outras bibliotecas e ferramentas para desenvolver uma aplicação para celular completa. Exemplos incluem UI Widgets e bibliotecas de gráficos 3D;
- **Framework:** Conjunto de bibliotecas, componentes de software e diretrizes de arquitetura que fornecem ao desenvolvedor um conjunto de ferramentas abrangente para criar um aplicativo para celular completo;
- **Plataforma:** Um conjunto de frameworks, ferramentas e serviços que não apenas permitem ao desenvolvedor construir um aplicativo para aparelho celular completo, como também empacotá-lo para distribuição nas lojas de software ou na nuvem. Geralmente, incluem algum tipo de ambiente de desenvolvimento integrado para facilitar a construção, documentação completa, apoio e ferramentas de automação;
- **Produto/Serviço:** Fornece funcionalidade ou serviço pronto para ser usado por um aplicativo em um aparelho celular.

Na tabela 2 são apresentados alguns dos mais populares frameworks de desenvolvimento de aplicativos multiplataforma para aparelhos celulares.

Tabela 2. Ferramentas de desenvolvimento multiplataforma para aparelhos celulares mais populares. Adaptado de (HARTMANN et al., 2011)

Ferramentas de desenvolvimento	URL	Licença	Tipo
Rhodes	<a href="http://rhone.com/products/rhodes/">http://rhone.com/products/rhodes/</a>	Open Source	Plataforma
Phonegap	<a href="http://www.phonegap.com">http://www.phonegap.com</a>	Open Source	Framework
FeedHenry	<a href="http://developer.feedhenry.com/">http://developer.feedhenry.com/</a>	Commercial	Plataforma
Appcelerator	<a href="http://www.appcelerator.com/">http://www.appcelerator.com/</a>	Open Source	Plataforma
Grapple	<a href="http://www.qrapplemobile.com/">http://www.qrapplemobile.com/</a>	Commercial	Framework
MotherApp	<a href="http://www.motherapp.com/">http://www.motherapp.com/</a>	Commercial	Framework
Corona	<a href="http://www.anscamobile.com/corona/">http://www.anscamobile.com/corona/</a>	Commercial	Produto
Sencha Touch	<a href="http://www.sencha.com/products/touch/">http://www.sencha.com/products/touch/</a>	OS/Commercial	Biblioteca
MoSync	<a href="http://www.mosync.com/">http://www.mosync.com/</a>	Open Source	Plataforma
Resco	<a href="http://www.resco.net/">http://www.resco.net/</a>	Commercial	Plataforma
CouchOne	<a href="http://www.couchone.com/products">http://www.couchone.com/products</a>	Commercial	Plataforma
MobileIron	<a href="http://mobileiron.com/">http://mobileiron.com/</a>	Commercial	Plataforma
WidgetPad	<a href="http://widgetpad.com">http://widgetpad.com</a>	Open Source	Plataforma
AML	<a href="http://www.amlcode.com">http://www.amlcode.com</a>	Open Source	Framework
Jo	<a href="http://joapp.com">http://joapp.com</a>	Open Source	Biblioteca
xui	<a href="http://xuijs.com">http://xuijs.com</a>	Open Source	Biblioteca
jQuery Mobile	<a href="http://jquerymobile.com">http://jquerymobile.com</a>	Open Source	Biblioteca
JQTouch	<a href="http://jqtouch.com">http://jqtouch.com</a>	Open Source	Biblioteca
QT	<a href="http://qt.nokia.com/products/qt-for-mobile-platforms/">http://qt.nokia.com/products/qt-for-mobile-platforms/</a>	Open Source	Framework
QuickConnectFamily	<a href="http://www.quickconnectfamily.org/">http://www.quickconnectfamily.org/</a>	Open Source	Framework
Bedrock	<a href="http://www.metismo.com">http://www.metismo.com</a>	Commercial	Plataforma
WebApp.net	<a href="http://webapp-net.com/">http://webapp-net.com/</a>	Open Source	Framework



Conceituar as diferentes abordagens utilizadas pelas ferramentas de desenvolvimento de aplicativos multiplataforma para aparelhos celulares é fundamental, pois para cada tipo de abordagem há distintas limitações e podem atender a diferentes propósitos.

#### A) *Aplicativo Web Puro*

Segundo Hartmann et al. (2011) a abordagem Web Puro é cada vez mais popular e tem como objetivo construir uma aplicação web móvel que será executada no navegador do aparelho celular. Para tanto, são utilizadas tecnologias Web Padrão como HTML, CSS e Javascript para construir a aplicação com comportamento muito similar a um aplicativo nativo. Isto é possível devido às capacidades avançadas de HTML5 e CSS3, incluindo locais para armazenamento, animações, telas, soquetes Web e reprodução de vídeo.

Smutny (2012) divide esta abordagem em dois tipos: Aplicativo Web Dedicado, que pode ser executado apenas em algumas plataformas e Aplicativo Web Genérico, que pode ser executado em qualquer plataforma;

Segundo Smutny (2012) a maioria dos aparelhos celulares possuem navegadores que suportam HTML5 e CSS3 e este tipo de abordagem nos frameworks de desenvolvimento multiplataforma para aparelhos celulares, visam tirar proveito das especificações W3C para oferecer uma melhor experiência ao usuário. Smutny (2012) cita o framework Sencha Touch como um exemplo de ferramenta de desenvolvimento que emprega esta abordagem.

O Sencha Touch é um framework para aplicativos móveis que utiliza recursos da tecnologia HTML5. Este framework permite ao desenvolvedor criar aplicativos para as plataformas IOS, Android, BlackBerry, Windows Phone, entre outros. (SENCHA TOUCH, 2013)

Segundo Smutny (2012), o Sencha Touch é um framework open source construído especificamente para alavancar HTML5, CSS3 e Javascript ao seu mais alto nível de poder e flexibilidade, tais como local de armazenamento e suporte a vídeo e áudio. Também oferece integração entre uma variedade de fontes, tais como Ajax, JSON e YQL.

A execução deste tipo de aplicativo é classificada por Hartmann et al. (2011) como web puro, pois a execução é realizada por completo no navegador do aparelho celular.

White (2013) classifica este tipo de aplicativo como sites otimizados para telas menores. O sistema operacional do dispositivo não tem controle sobre seu conteúdo ou funcionalidade. Os desenvolvedores podem fazer alterações a qualquer momento, com efeito imediato aos usuários.

#### B) *Aplicativos Web Híbrido*

Hartmann et al. (2011) cita uma variação da abordagem de aplicativos Web Puro denominada como aplicativo web híbrido.

Segundo Charland e Leroux (2011), a única coisa em comum entre os sistemas operacionais para aparelhos celulares está no fato de todos possuírem um navegador móvel que é acessível através de programação por código

nativo. Cada plataforma permite que seja aberta uma instância do navegador, denominada WebView, interagindo com sua interface Javascript de código nativo. Este tipo de abordagem permite a criação de aplicativos Web Híbridos utilizando HTML, CSS e Javascript e ainda utilizar recursos nativos do dispositivo móvel e sensores através de uma API Javascript comum.

Na definição de Hartmann et al. (2011) um aplicativo Web Híbrido consiste em um aplicativo Web incorporado em um aplicativo nativo que fornece uma ponte para o sistema operacional e os serviços nativos do dispositivo. A comunicação entre o aplicativo Web e o nativo normalmente ocorre através de APIs escritas em Javascript. Esta técnica visa reunir o melhor dos dois mundos em uma única solução integrada: a flexibilidade de aplicativos Web com a velocidade e riqueza de recursos de um aplicativo nativo.

Para a abordagem denominada Web Híbrido, Ng Moon et al. (2013) destacam o framework PhoneGap. Charland e Leroux (2011) também apontam o PhoneGap como uma ferramenta de desenvolvimento para aplicativos Web Híbridos.

PhoneGap é um framework de desenvolvimento de aplicações multiplataforma para aparelhos celulares que permite a utilização de tecnologias Web Padrão como HTML5, CSS3 e Javascript evitando linguagem de desenvolvimento nativa das plataformas móveis. Os aplicativos são executados em pacotes direcionados para cada plataforma e dependem de padrões compatíveis com sua API para acessar recursos de cada dispositivo, dados e status de rede. Esta ferramenta de desenvolvimento é compatível as plataformas Android, IOS, BlackBerry OS, Windows Phone, Bada, WebOS e Symbian. (PHONEGAP, 2013).

Na figura 1 é apresentada a arquitetura do framework PhoneGap. A camada superior representa o código fonte da aplicação. A camada central é composta por bibliotecas Javascript e nativas, sendo esta camada responsável pela interface entre a aplicação Web e a aplicação nativa.

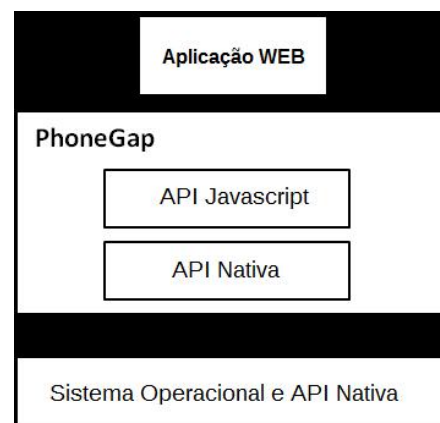


Figura 1. Camadas da arquitetura do framework PhoneGap (PALMIERI et al. 2012)

A tabela 3 mostra quais APIs estão disponíveis no PhoneGap para cada sistema operacional suportado.

Tabela 3. Recursos suportados pelo framework PhoneGap para cada plataforma. Adaptado de (PHONEGAP, 2013)

	iPhone/ iPhone 3G	iPhone 3GS e Novos	ANDROID	BlackBerry OS 5.x	BlackBerry OS 6.0+	WebOS	Windows Phone 7 e 8	Symbian	BADA
ACELERÔMETRO	✓	✓	✓	✓	✓	✓	✓	✓	✓
CÂMERA	✓	✓	✓	✓	✓	✓	✓	✓	✓
BÚSSOLA	x	✓	✓	x	x	✓	✓	x	✓
CONTATOS	✓	✓	✓	✓	✓	x	✓	✓	✓
ARQUIVO	✓	✓	✓	✓	✓	x	✓	x	x
GEOLOCALIZAÇÃO	✓	✓	✓	✓	✓	✓	✓	✓	✓
MÍDIA	✓	✓	✓	x	x	x	✓	x	x
REDE	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICAÇÃO (ALERTA)	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICAÇÃO (SOM)	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICAÇÃO (VIBRAÇÃO)	✓	✓	✓	✓	✓	✓	✓	✓	✓
ARMAZENAMENTO	✓	✓	✓	✓	✓	✓	✓	x	x

Os aplicativos Web Híbridos podem ser empacotados de acordo com a plataforma destino, portanto é possível comercializá-los nas lojas de aplicativos juntamente aos aplicativos nativos.

C) *Compilação cruzada*

Hartmann et al. (2011) cita a abordagem Compilação Cruzada que tem por objetivo converter a codificação do framework em código nativo. Para esta abordagem, o framework fornece uma API independente de plataforma utilizando uma linguagem de programação convencional. Os desenvolvedores utilizam esta API para criar o código do aplicativo, incluindo a interface do usuário, persistência de dados e regras de negócio. Este código é depois compilado por um Compilador Cruzado, responsável pela conversão para código nativo da plataforma.

Segundo Hartmann et al. (2011) o framework AppCelerator Titanium utiliza esta abordagem, pois uma única base de código fonte escrito em Javascript é convertido em código nativo em tempo de compilação e empacotado para diferentes plataformas destino.

O AppCelerator Titanium é ambiente de desenvolvimento de aplicativos multiplataforma para aparelhos celulares, incluindo IOS, Android e Mobile Web Sites. (TITANIUM MOBILE, 2013)

De acordo com Smutny (2012), o AppCelerator Titanium possui uma interface com módulos nativos baseada em Javascript. Pode armazenar preferências do usuário, salvar arquivos de dados ou implementar a versão móvel de um cookie utilizando, por exemplo, SQLite.

Estes módulos nativos baseados em Javascript são APIs que permitem o acesso a diversos recursos nativos do dispositivo móvel.

A arquitetura do framework AppCelerator Titanium é apresentada na figura 2. Na parte inferior da arquitetura está o sistema operacional cliente: Android, IOS ou navegador. No topo está o aplicativo, construído em Javascript. No meio está o Titanium SDK e as APIs por ele disponibilizadas. O aplicativo será desenvolvido invocando as APIs do Titanium SDK que permitirá o acesso aos recursos nativos do dispositivo móvel (TITANIUM MOBILE, 2013).

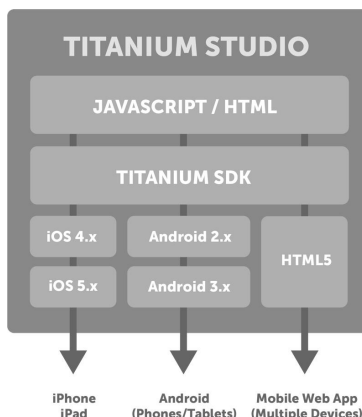


Figura 2. Ilustração da arquitetura do framework AppCelerator Titanium. (TITANIUM MOBILE, 2013)

A tabela 4 mostra as APIs que estão disponíveis no AppCelerator Titanium para cada sistema operacional suportado:

Tabela 4. Acesso aos recursos nativos do aparelho celular pelo framework AppCelerator Titanium. Adaptado de (HARTMANN et al. 2011)

	IPHONE	Windows Mobile	BlackBerry	Symbian	Android	Palm
GEOLocalização	SIM	N/A	N/A	N/A	SIM	N/A
CONTATOS PIM	SIM	N/A	N/A	N/A	PARCIAL	N/A
CÂMERA	SIM	N/A	N/A	N/A	SIM	N/A
MENU NATIVO/ TAB BAR	SIM	N/A	N/A	N/A	SIM	N/A
CÓDIGO DE BARRAS	NÃO	N/A	N/A	N/A	NÃO	N/A
CAPTURA DE ÁUDIO/VÍDEO	SIM	N/A	N/A	N/A	SIM	N/A
BLUETOOTH	NÃO	N/A	N/A	N/A	NÃO	N/A
SMS	PARCIAL	N/A	N/A	N/A	PARCIAL	N/A
CALENDÁRIO	NÃO	N/A	N/A	N/A	SIM	N/A
ROTAÇÃO DA TELA	SIM	N/A	N/A	N/A	SIM	N/A
MAPAS	SIM	N/A	N/A	N/A	SIM	N/A
TOQUES	NÃO	N/A	N/A	N/A	NÃO	N/A
ARMAZENAMENTO	SIM	N/A	N/A	N/A	SIM	N/A

#### D) Mobile Widget

Há uma outra abordagem citada por Hartmann et al. (2011) denominada como Mobile Widget. O conceito de Widget foi introduzido muito antes dos aplicativos para aparelhos celulares e da revolução das Lojas de Aplicativos (App Store) e pode ser visto como a primeira tentativa de entrega de pequenas funcionalidades em uma maneira leve e intuitiva para o usuário final. Um Widget é uma ferramenta interativa que oferece um serviço de finalidade única para o usuário, como mostrar notícias, data e hora atual, previsão do tempo, calendário, dicionário, calculadora, mapa ou até mesmo um tradutor de linguagem.

Hartmann et al. (2011) define os Mobile Widgets como pequenos aplicativos para aparelhos celulares que normalmente são escritos utilizando as tecnologias HTML, Javascript, CSS e XML.

Segundo Ngu Phuc e Do Van (2012), os Mobile Widgets são aplicativos leves e para propósitos específicos que são executados em uma plataforma de execução específica e com a engenharia necessária para execução deste tipo de aplicativo. Diferentes tipos de Mobile Widgets necessitam de diferentes ambientes para serem executados.

Alguns exemplos de plataformas para execução de Widgets móveis são Nokia S60, WRT, Windows Mobile Widget, Opera Widget e JIL Widget. Todavia, cada um deles possui uma especificação diferente. Devido a esta característica, um Mobile Widget pode funcionar bem em um aparelho, enquanto que em outro não. (BIN et al. 2011).

Alguns Mobile Widgets já vêm instalados no telefone, no entanto, os usuários podem baixar e instalar outros Mobile Widgets em seus aparelhos. Estes tipos de aplicativos são valiosos aos desenvolvedores quando há o propósito de desenvolver um aplicativo leve, simples e funcional voltado para aparelhos celulares. (NGU PHUC; DO VAN, 2012).

#### IV. A MELHOR SOLUÇÃO: EXISTE UM PARADIGMA DE DESENVOLVIMENTO IDEAL?

Como apresentado em sessões anteriores, há diversos tipos

de técnicas e abordagens para o desenvolvimento de aplicativos para aparelhos celulares. A definição do paradigma de desenvolvimento ideal é resultado de um conhecimento abrangente sobre as funcionalidades do aplicativo a ser desenvolvido.

White (2013) apresenta um estudo de caso citando a experiência da empresa Facebook no desenvolvimento de um aplicativo para aparelhos celulares. Segundo White, o propósito da empresa era desenvolver um aplicativo flexível e a decisão foi por uma aplicação multiplataforma, selecionando a abordagem web híbrida, apostando fortemente nos recursos da tecnologia web HTML5. De acordo com White, o aplicativo resultante foi problemático, sendo apontado universalmente como lento, repleto de erros, inconsistente e propenso a frequentes falhas. White sugere provável inferioridade dos aplicativos web híbridos frente aos genuinamente nativos, ou simplesmente, erros da equipe de desenvolvimento da aplicação web híbrida como motivos para os problemas ocorridos. O aplicativo Facebook foi completamente reescrito na abordagem nativa.

O estudo de caso apresentado por White demonstra consequências de uma escolha errada de técnica de desenvolvimento e abordagem utilizada na criação de um aplicativo para este mercado, portanto, podemos concluir que esta é uma etapa fundamental.

Não há uma solução que atenda a todas as necessidades deste mercado. Aplicativos nativos, por exemplo, tem bom desempenho e proporcionam uma experiência de uso rica ao usuário, entretanto, são desenvolvidos para serem executados em apenas uma plataforma acarretando maiores custos para o desenvolvimento. Para as abordagens multiplataforma, também existem limitações como acesso a recursos nativos do dispositivo, dependência da comunidade que mantém a ferramenta de desenvolvimento e experiência de uso limitada, entretanto, sua capacidade multiplataforma pode ser um diferencial importante, além da redução de complexidade, custo e manutenção oferecidos por estas ferramentas. Entre as abordagens multiplataforma também existem diferenças de



técnica de desenvolvimento, acesso a recursos nativos do aparelho, empacotamento, entre outras, portanto estas diferenças também devem ser analisadas.

#### V. ESTUDO DE CASO: SISTEMA DE GESTÃO PATRIMONIAL (GEPAT)

O Sistema de Gestão Patrimonial (GEPAT) é um software que possibilita o gerenciamento de patrimônio. Desenvolvido para utilização pelo Departamento de Gestão Patrimonial da Universidade Federal de São Carlos, este software oferece as seguintes funcionalidades:

- Cadastramento de informações auxiliares (Descrição padrão, Unidade, Fornecedor, Funcionário, Conta Contábil, Grupo de Contas e Convênio);
- Relatório de bens disponíveis;
- Controle de depreciação e valor de mercado;
- Cadastramento de entrada de bens;
- Realização de inventário de bens - Através da tecnologia NFC (Near Field Communication), etiquetas fixadas em cada bem são lidas na realização do inventário;
- Geração de termos de responsabilidade;
- Listagem de todos os bens cadastrados;

Através destas funcionalidades, é possível realizar o controle dos bens da instituição bem como gerenciar os processos inerentes a este negócio como por exemplo, a geração dos termos de responsabilidade, assinatura dos termos pelos responsáveis, arquivamento dos termos assinados para consultas futuras e controle dos termos com assinaturas pendentes.

O GEPAT pode ser acessado através de um navegador existente em um desktop ou através de um aparelho celular ou tablet. Denominado como GEPAT Móvel, o aplicativo desenvolvido para aparelhos celulares é um aplicativo nativo, desenvolvido para ser executado no sistema operacional Android.

Dentre as funcionalidades existentes no GEPAT Móvel, destacamos a realização de inventário de bens. No desenvolvimento desta funcionalidade foram utilizados os recursos da tecnologia Near Field Communication (NFC).

A tecnologia Near Filed Communication é emergente para aparelhos eletrônicos e permite a comunicação através de um simples toque ou da aproximação dos aparelhos. A comunicação através de NFC é possível através de dois aparelhos ativos, como aparelhos celulares, ou através de um aparelho celular e um passivo (ou sem alimentação) como uma etiqueta. (YAQUB e SHAIKH, 2012).

As características da tecnologia NFC resolvem os problemas para a realização de inventário de bens através do GEPAT Móvel. Para tanto, através do NFC foi utilizada a forma de comunicação entre um aparelho celular e uma etiqueta, onde informações sobre um bem são registradas em uma etiqueta que pode ser lida através de um aparelho com a tecnologia NFC, facilitando a atividade de inventário de bens.

O aparelho celular selecionado para execução da funcionalidade de inventário de bens na instituição através do GEPAT Móvel foi o Samsung Galaxy S3, tendo em vista possuir a tecnologia NFC e utilizar o sistema operacional

Android, entretanto, outros aparelhos também oferecem esta tecnologia, tais como: Samsung Galaxy Note 2, Motorola RAZR HR, Motorola RAZR i, Nokia 701, Nokia Lumia 920, Sony Xperia S, LG Optimus 4X HD P880 e BlackBerry Bold 9900.

A necessidade de utilização da tecnologia NFC foi determinante para a decisão do tipo de aplicativo a ser desenvolvido, portanto o GEPAT Móvel é um aplicativo Nativo, pois precisa acessar recursos nativos específicos do aparelho para a utilização dos recursos da tecnologia NFC.

Outro ponto importante está na ausência de frameworks de desenvolvimento de aplicativos multiplataforma que disponibilizem uma API de acesso ao NFC. Estas características reforçam o percepção de que o GEPAT Mobile não pode ser um aplicativo multiplataforma.

#### VI. TRABALHOS RELACIONADOS

O trabalho realizado por Hartmann G. et al. (2011) tem o propósito de discutir as técnicas e abordagens existentes no desenvolvimento de aplicativos para aparelhos celulares, destacando abordagens de desenvolvimento multiplataforma. Para tanto são apresentadas diferentes técnicas e suas características.

Palmieri et al. (2012) também apresenta em seu estudo características de várias ferramentas de desenvolvimento de aplicativos multiplataforma para aparelhos celulares, destacando para cada framework pontos relevantes a serem observados na escolha da melhor ferramenta de desenvolvimento.

Smutny (2012) cita diferentes tipos de abordagens no desenvolvimento de aplicativos para aparelhos celulares, destacando também características de cada abordagem e apresentando ferramentas de desenvolvimento multiplataforma.

No estudo realizado por Charland e Leroux (2011) é realizada uma discussão a respeito das abordagens Nativa e Web no desenvolvimento de aplicativos para aparelhos celulares. Também é apresentada uma ferramenta de desenvolvimento multiplataforma.

É comum entre estes trabalhos, a percepção de novas tecnologias e avanços neste mercado. O debate a respeito de abordagens multiplataforma em confronto com as características de aplicativos Nativos é uma proposta existentes em todos os estudos apresentados.

Este estudo diferencia-se dos demais ao apresentar um estudo de caso utilizando o GEPAT Móvel, onde através da análise das funcionalidades do aplicativo foi possível definir o melhor paradigma para seu desenvolvimento.

#### VII. CONCLUSÃO E TRABALHOS FUTUROS

Há uma preocupação clara das empresas que desenvolvem aplicativos para aparelhos celulares com redução de custo, tempo e complexidade. As ferramentas de desenvolvimento multiplataforma tem o objetivo de apresentar soluções e atender a estas necessidades.

As ferramentas de desenvolvimento permitem a redução de custo, tempo e complexidade, entretanto, com limitações que

variam de acordo com a ferramenta multiplataforma existente.

É fundamental para um paradigma de desenvolvimento ideal utilizar linguagem de programação popular, simples e abrangente, que permita desenvolvimento e manutenção ágil, com acesso ilimitado aos recursos nativos do dispositivo móvel, podendo ser executado em todas as plataformas existentes, sem perda de desempenho, podendo ser empacotado e disponibilizado em lojas de aplicativos.

O cenário dos aplicativos para aparelhos celulares pode ser claramente associado à experiência recente com os desktops, onde softwares nativos foram superados pelos softwares Web, portanto, é sensato supor a evolução das abordagens com foco em desenvolvimento multiplataforma.

Aplicativos multiplataforma podem oferecer ganhos aos fabricantes de plataforma, uma vez que as empresas que desenvolvem aplicativos para aparelhos celulares são estimuladas a desenvolver novos aplicativos que possam ser desenvolvidos uma única vez e comercializados em várias lojas de aplicativos. Este fato sugere talvez participação futura das empresas proprietárias das plataformas para aparelhos celulares neste processo de evolução das ferramentas de desenvolvimento multiplataforma.

Atualmente, não há um paradigma de desenvolvimento ideal para todas as necessidades. O estudo de caso utilizando o GEPAT Móvel e os erros cometidos pela empresa Facebook apresentados neste estudo, demonstram a importância de conhecer todas as funcionalidades do aplicativo para definição do melhor paradigma de desenvolvimento.

Ao empreendedor que deseja atuar neste mercado, cabe conhecimento sobre todos os tipos de paradigmas existentes, conhecimento do mercado ao qual o aplicativo será inserido e das funcionalidades oferecidas por ele, bem como um conhecimento aprofundado do público alvo deste aplicativo, para que através de uma análise envolvendo todas estas questões, se faça a escolha do paradigma de desenvolvimento apropriado.

Em trabalhos futuros, pode-se apresentar um estudo de caso em aplicativos com outros tipos de funcionalidades, demonstrando a utilidade de outros tipos de paradigmas apresentados neste estudo.

#### REFERÊNCIAS

- BIN, Z., et al. (2011). Research and implementation of cross-platform development of mobile widget. Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on. Disponível em: <<http://ieeexplore.ieee.org/ielx5/5993482/6013532/06014238.pdf?tp=&arnumber=6014238&isnumber=6013532>>. Acesso em 17 set. 2013.
- CHARLAND A. and LEROUX B. (2011). Mobile Application Development: Web vs. Native. Communications of the ACM 54(5): 49-53. Disponível em: <<http://cacm.acm.org/e31.periodicos.capes.gov.br/magazines/2011/5/107700-mobile-application-development/fulltext>>. Acesso em 03 set. 2013.
- HARTMANN, G., et al. (2011). Cross-platform mobile development. Tribal, Lincoln House, The Paddocks, Tech. Rep. Disponível em: <<https://wss.apan.org/1539/JKO/mole/Shared%20Documents/Cross-Platform%20Mobile%20Development.pdf>>. Acesso em 16 out. 2013.
- MARTINS C. S, et al. (2013). Os desafios para a mobilização de aplicações baseadas em plataforma Web. X Encontro Anual de Computação - EnAComp 2013, Fundação CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações Campinas – SP – Brasil. Disponível em: <<https://projetos.extras.ufg.br/enacomp/anais/pdf/39.pdf>>. Acesso em 19 abr. 2013.
- NG MOON, H., et al. (2013). Cross-platform mobile applications for android and iOS. Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6548969>>. Acesso em 14 out. 2013.
- NGU PHUC, H. and DO VAN T. (2012). Selecting the right mobile app paradigms. Service-Oriented Computing and Applications (SOCA), 2012 5th IEEE International Conference on. Disponível em: <<http://ieeexplore.ieee.org/ielx5/6422836/6449417/06449450.pdf?tp=&arnumber=6449450&isnumber=6449417>>. Acesso em 18 set. 2013.
- PALMIERI, M., et al. (2012). Comparison of cross-platform mobile development tools. Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on. Disponível em: <<http://ieeexplore.ieee.org/ielx5/6362334/6376008/06376023.pdf?tp=&arnumber=6376023&isnumber=6376008>>. Acesso em 19 abr. 2013.
- PHONEGAP. Página oficial do framework. Disponível em: <<http://phonegap.com/>> Acesso em: 16 out. 2013.
- SAMBASIVAN, D., et al. (2011). Generic framework for mobile application development. Internet (AH-ICI), 2011 Second Asian Himalayas International Conference on. Disponível em: <<http://ieeexplore.ieee.org/ielx5/6104277/6113924/06113938.pdf?tp=&arnumber=6113938&isnumber=6113924>>. Acesso em 05 abr. 2013.
- SENCHA TOUCH. Página oficial do framework. Disponível em: <<http://www.sencha.com/products/touch>>. Acesso em 17 out. 2013.
- SMUTNY, P. (2012). Mobile development tools and cross-platform solutions. Carpathian Control Conference (ICCC), 2012 13th International. Disponível em: <<http://ieeexplore.ieee.org/ielx5/6222116/6228605/06228727.pdf?tp=&arnumber=6228727&isnumber=6228605>>. Acesso em 05 abr. 2013.
- TITANIUM MOBILE. Página oficial do framework. Disponível em: <http://www.appcelerator.com/titanium/>>. Acesso em 16 out. 2013.
- WHITE, J. (2013). Going native (or not): Five questions to ask mobile application developers. The Australasian



medical journal 6(1): 7. Disponível em: <<http://www.amj.net.au/index.php?journal=AMJ&page=article&op=viewFile&path%5B%5D=1576&path%5B%5D=1018>>. Acesso em 18 out. 2013.

YAQUB, M. U and SHAIKH, U. A. (2012). Near Field Communication. Disponível em: <[http://www1.kfupm.edu.sa/studentaffairs/ar/ssc4/4845\\_MohammedUmair\\_Yaqub.pdf](http://www1.kfupm.edu.sa/studentaffairs/ar/ssc4/4845_MohammedUmair_Yaqub.pdf)>. Acesso em 10 out. 2013.