

**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS
FATEC PROFESSOR JESSEN VIDAL**

PAULO LUAN MARIANO SILVA

**DESENVOLVIMENTO DE APLICATIVOS INTEROPERÁVEIS PARA
DISPOSITIVOS MÓVEIS: AVALIAÇÃO DE TECNOLOGIAS E
IMPLEMENTAÇÃO**

**São José dos Campos
2012**

PAULO LUAN MARIANO SILVA

**DESENVOLVIMENTO DE APLICATIVOS
INTEROPERÁVEIS PARA DISPOSITIVOS MÓVEIS:
AVALIAÇÃO DE TECNOLOGIAS E IMPLEMENTAÇÃO**

Trabalho de Graduação apresentado à
Faculdade de Tecnologia São José dos
Campos, como parte dos requisitos
necessários para a obtenção do título de
Tecnólogo em Bancos de Dados.

**Orientadores: Ubirajara Moura de Freitas,
Giuliano Araújo Bertoti**

São José dos Campos
2012

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

SILVA, Paulo Luan Mariano
Desenvolvimento de aplicativos interoperáveis para dispositivos móveis: Avaliação de tecnologias e Implementação São José dos Campos, 2012 75f.

Trabalho de Graduação – Curso de Tecnologia em Bancos de
Dados, FATEC de São José dos Campos: Professor Jessen Vidal, 2012.
Orientadores: Ubirajara Moura de Freitas, Giuliano Araújo Bertoti.

1. METODOLOGIA E TÉCNICAS DA COMPUTAÇÃO. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Computação. II. Desenvolvimento de aplicativos interoperáveis para dispositivos móveis: Avaliação de tecnologias e Implementação.

REFERÊNCIA BIBLIOGRÁFICA –

SILVA, Paulo Luan Mariano. **Desenvolvimento de aplicativos interoperáveis para dispositivos móveis: Avaliação de tecnologias e Implementação.** 2012. 75f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

CESSÃO DE DIREITOS –

Paulo Luan Mariano Silva.
Desenvolvimento de Aplicativos Interoperáveis para Dispositivos Móveis: Avaliação de Tecnologias e Implementação.
Trabalho de Graduação 2012.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

Paulo Luan Mariano Silva
Faculdade de Tecnologia Jessen Vidal
CEP 12247-004 – São José dos Campos – São Paulo

PAULO LUAN MARIANO SILVA

**DESENVOLVIMENTO DE APLICATIVOS
INTEROPERÁVEIS PARA DISPOSITIVOS MÓVEIS:
AVALIAÇÃO DE TECNOLOGIAS E IMPLEMENTAÇÃO**

Trabalho de Graduação apresentado à
Faculdade de Tecnologia São José dos
Campos, como parte dos requisitos
necessários para a obtenção do título de
Tecnólogo em Bancos de Dados.

ADRIANA DA SILVA JACINTO, Me.

WILLIAM ANTÔNIO SIQUEIRA, Gr.

UBIRAJARA MOURA DE FREITAS, Me.

GIULIANO ARAÚJO BERTOTI, Me.

____/____/____

DATA DA APROVAÇÃO

À minha família, por sempre me incentivar a ser uma pessoa melhor.

AGRADECIMENTOS

Agradeço primeiramente a minha família, sem eles eu realmente não seria nada, o meu obrigado de todo o coração ao meu pai a minha mãe por serem as pessoas mais incríveis que conheço, por terem me educado e por terem me dado esta oportunidade de chegar onde cheguei. Agradeço também a todos os professores e funcionários da Fatec por acreditar sempre em meu potencial. E em especial, a instituição Funcate por me proporcionar a oportunidade de conhecer pessoas incríveis das quais sempre estavam dispostas a me ajudar e a ceder conhecimento.

“A Simplicidade é o último degrau da
sabedoria”

Khalil Gibran

RESUMO

Com a modernização da computação móvel e o crescimento do uso destes aparelhos surgiu uma nova tendência de mercado: os aplicativos móveis. Atualmente, existem diversos sistemas operacionais para dispositivos móveis, sendo que cada um dispõe de diferentes tecnologias para o desenvolvimento de aplicativos, dificultando o desenvolvimento de uma única aplicação para todas as plataformas. Algumas ferramentas permitem o desenvolvimento de apenas uma base de código que pode ser executada em diferentes plataformas, tendo um comportamento similar ao de uma aplicação nativa. Em meio a tantas tecnologias híbridas emergentes, existe a dificuldade de selecionar a melhor ferramenta baseada em suas necessidades. Esta escolha é essencial para o sucesso do projeto, pois a seleção inadequada traz prejuízos como atrasos na entrega, grande curva de aprendizado para o desenvolvedor, além de questões técnicas como projetos mal estruturados e baixo desempenho na execução da aplicação. Selecionar a melhor ferramenta aumenta a eficiência tanto na parte técnica como na parte administrativa do projeto, diminuindo por vezes o seu custo. O objetivo deste trabalho é selecionar e avaliar tecnologias emergentes em desenvolvimento móvel híbrido, demonstrando como estudo de caso o desenvolvimento de uma aplicação de visualização de gasodutos.

Palavras-Chave: Computação; Móvel; Comparativo; Celular; Tablet; Avaliação; Aplicativos; Dutos; Híbrido; Interoperável.

ABSTRACT

With the modernization of mobile computing and the growing usage of these devices has emerged a new market trend: mobile applications. Currently, there are several mobile operating systems and each has different technologies for application development. This difference hinders the development of the same application for all platforms because each has different programming languages involved. Some tools allow the development of a single source code that can be converted to different platforms, having the same behavior as a native application. Among many hybrid technologies emerging, it is difficult to select the best tool based on their needs. This choice is essential to project success because the improper selection harms as late delivery, big learning curve beyond technical issues, like bad projects structured and low performance in the execution of the application. Select the best tool increases efficiency in the technical and administrative areas of the project, sometimes decreasing its cost. The objective of this survey is to evaluate emerging technologies for developing mobile hybrid as a case study demonstrating the development of a visualization application pipeline with selected tools.

Keywords: Mobile Computing; Comparative; Phone; Tablet; Evaluation; Applications; Pipelines; Hybrid; Interoperable.

LISTA DE FIGURAS

FIGURA 1 - ARQUITETURA WEB HÍBRIDA	21
FIGURA 2: HIERARQUIA DA COMUNICAÇÃO EM UM APLICATIVO UTILIZANDO PHONEGAP.	23
FIGURA 3: PLATAFORMAS SUPORTADAS PELO PHONEGAP.	24
FIGURA 4: FUNCIONAMENTO DO PHONEGAP BUILD	24
FIGURA 5: MODELO DE ARQUITETURA DE INTERFACES GRÁFICAS NATIVAS NO TITANIUM.	25
FIGURA 6: À ESQUERDA APARÊNCIA NO EMULADOR DO IPHONE, À DIREITA APARÊNCIA EM EMULADOR ANDROID.	27
FIGURA 7: INFOGRÁFICO DO USO DA INTERNET NO BRASIL.	29
FIGURA 8: APLICAÇÃO UTILIZANDO OPENLAYERS COM JQUERYMOBILE.	33
FIGURA 9: APLICAÇÃO UTILIZANDO OPENLAYERS COM SENCHA TOUCH 2.	34
FIGURA 10: EXEMPLO DE MAPA PARA APARELHOS MÓVEIS DISPONÍVEL NO SITE OFICIAL DO FRAMEWORK.	36
FIGURA 11: EXEMPLO DE CRIAÇÃO DE MAPA COM UMA POPUP DE INFORMAÇÕES.	37
FIGURA 12: PLATAFORMAS SUPORTADAS PELO JQUERYMOBILE.	38
FIGURA 13: FERRAMENTA DE PROTOTIPAÇÃO DO JQUERY MOBILE.	39
FIGURA 14: FIGURA DA DOCUMENTAÇÃO DO SENCHA TOUCH 2	41
FIGURA 15: EXEMPLO DE WIDGETS NO JO	43
FIGURA 16: WEBAPP DE TESTE DOS WIDGETS DISPONÍVEIS NO JQTOUCH.	45
FIGURA 17: PRIMEIRA TELA, APLICAÇÃO DE EXEMPLO DA PLATAFORMA MOSYNC, SENDO EXECUTADA NAS PLATAFORMAS ANDROID E IOS, ALGUNS DADOS DO SISTEMA OPERACIONAL SENDO EXIBIDOS.	48
FIGURA 18: SEGUNDA TELA DA APLICAÇÃO, ALGUMAS MEDIÇÕES EXTRAÍDAS DE RECURSOS NATIVOS DA PLATAFORMA.	48
FIGURA 19: TERCEIRA TELA DA APLICAÇÃO, WEBVIEW QUE EXIBE UM MAPA NA TERCEIRA TELA DA APLICAÇÃO.	49
FIGURA 20: TESTES DE UMA APLICAÇÃO DE EXEMPLO, NO EMULADOR DA EMPRESA CODENAMEONE	50
FIGURA 21: TELA INICIAL SENDO EXECUTADA NOS EMULADORES.	58
FIGURA 22 - GASODUTOS FICTÍCIOS, LOCALIZADOS PRÓXIMOS DA FATEC SJ.	59
FIGURA 23: INFORMAÇÕES DO GASODUTO VERMELHO.	60
FIGURA 24: BALÃO DE INFORMAÇÕES DO GASODUTO AZUL.	60
FIGURA 25: DETALHES DO DUTO VERMELHO (COMGAS)	61
FIGURA 26: DETALHES DO DUTO AZUL (PETROBRÁS)	62

LISTA DE ABREVIATURAS E SIGLAS

API: Application Programming Interface.

CSS: Cascade Style Sheet

DOM: Document Object Model.

FUNCATE: Fundação de Ciência, Aplicações e Tecnologia Espaciais.

HTML: Hyper Text Markup Language

HTTP: Hyper Text Transfer Protocol.

JDK: Java Development Kit.

JSON: Javascript Object Notation

MVC: Model View Controller

OGC: Open Geospatial Consortium.

REST: Representational State Transfer.

SDK: Software Development Kit.

SIG: Sistema de Informação Geográfica.

URL: Universal Resource Location.

W3C: World Wide Web Consortium.

WMS: Web Map Service.

WWW: World Wide Web.

XML: eXtensible Markup Language.

LISTA DE TABELAS

Tabela 1: Tabela comparativa dos componentes dispostos em cada um dos frameworks. 55

Tabela 2: Frameworks e Softwares que serão utilizados para o desenvolvimento do trabalho e suas respectivas versões. 56

LISTA DE ANEXOS

Anexo A – Tabela de frameworks interoperáveis

78

SUMÁRIO

1.	Introdução.....	16
1.1	Motivação.....	16
1.2	Objetivo	17
1.2.1	Objetivo Geral.....	17
1.2.2	Objetivo Específico.....	17
1.3	Metodologia.....	17
1.4	Organização Do Trabalho.....	18
2	Revisão Bibliográfica.....	19
2.1	Desenvolvimento Híbrido Em Plataformas Móveis.....	19
2.1.1	Interfaces Gráficas Web Híbridas	21
2.1.2	Interfaces Gráficas Nativas Híbridas	25
2.2	Sistemas De Informação Geográfica.....	28
2.2.1	Sigs Móveis.....	28
2.2.2	Arquitetura Cliente Servidor.....	30
3	Avaliações Das Tecnologias	31
3.1	Bibliotecas De Visualização De Mapas	32
3.1.2	Openlayers	32
3.1.3	Leaflet	34
3.2	Frameworks De Interface Gráfica	37
3.2.1	Frameworks Web Mobile	37
3.2.2	Frameworks De Interfaces Gráficas Nativas	45
3.3	Tecnologias Seleccionadas.....	51
4	Estudo De Caso: Desenvolvimento Da Aplicação Dutosmobile	55
4.1	Dutos Mobile	55
4.1.1	Requisitos E Arquitetura.....	56
4.1.2	Funcionamento Da Aplicação Em Diferentes Plataformas	57
5	Considerações Finais.....	63
5.1	Contribuições.....	63
5.1.1	Publicação	64
5.2	Trabalhos Futuros	64

Referências	66
-------------------	----

1. INTRODUÇÃO

1.1 Motivação

O poder computacional de aparelhos móveis se desenvolve de maneira veloz pois cada dia a necessidade dos usuários em relação ao uso de aplicativos nestes aparelhos aumenta. Neste contexto, os sistemas operacionais se diferenciam nas diversas plataformas móveis existentes (HARTMANN, STEAD, DEGANI, 2011).

A *International Data Corporation* (IDC), corporação especializada em pesquisas com tecnologia da informação, analisa que as vendas de *SmartPhones* em 2011 foram de aproximadamente 8,9 milhões de aparelhos e estima-se que em 2012 as vendas saltarão para 15,4 milhões, um crescimento de aproximadamente 73% (IDC, 2011).

Atualmente, existem diversas plataformas móveis emergentes no mercado onde cada uma dispõe de diferentes sistemas operacionais, logo, cada uma delas possuem diferentes formas de lidar com serviços nativos de sua plataforma, tais como o GPS, acelerômetro, bússola, câmera, entre outros (GARDNER, GRIGSBY, 2011).

Visto a incompatibilidade entre os sistemas operacionais móveis, surge a necessidade de se obter interoperabilidade entre as aplicações nas diversas plataformas. Diz-se interoperabilidade ou aplicação híbrida, quando uma mesma base de código consegue de maneira efetiva, comportar-se de maneira semelhante em diversas plataformas (REID, 2011).

A interoperabilidade das aplicações depende principalmente da arquitetura de cada um dos sistemas móveis, esta dificuldade se baseia principalmente na linguagem utilizada e metodologias sobre qual o sistema foi desenvolvido (REID, 2011). Um dos problemas decorrentes desta dificuldade são os custos dos projetos que se elevam, visto que para cada plataforma seria necessário alocar uma equipe, ou alocar mais horas de trabalho para a criação de um mesmo aplicativo para todas as plataformas (STEWART, 2011).

Um das formas de se obter interoperabilidade da aplicação entre praticamente todos os sistemas operacionais móveis é o desenvolvimento de aplicativos web. Esta metodologia permite que a aplicação seja criada sobre uma arquitetura interoperável, porém os recursos

nativos do aparelho ficam cada vez mais restritos, visto que não há uma padronização na arquitetura dos sistemas operacionais móveis (PHONEGAP, 2011).

A dificuldade de acesso aos componentes nativos do aparelho móvel podem ser solucionadas utilizando *frameworks* que abstraem as camadas de serviços nativos para acesso interoperável. O uso destes *frameworks* permitem que uma mesma aplicação desenvolvida com apenas um código-fonte possa ser executada ou convertida para diferentes plataformas móveis.

1.2 Objetivos

Apresentam-se os objetivos que se deseja atingir no desenvolvimento deste trabalho.

1.2.1 Objetivo Geral

Selecionar e avaliar frameworks disponíveis no mercado para o desenvolvimento de aplicativos móveis interoperáveis e implementar como estudo de caso o programa DutosMobile, um aplicativo de visualização de dutos subterrâneos.

1.2.2 Objetivo Específico

Busca-se com este trabalho atingir os seguintes objetivos específicos:

- Pesquisar e avaliar tecnologias que auxiliem o desenvolvimento de aplicações móveis híbridas.
- Analisar frameworks e tecnologias para o módulo de mapas da aplicação.
- Desenvolver como estudo de caso a aplicação DutosMobile.

1.3 Metodologia

Para o desenvolvimento deste trabalho, serão avaliadas as tecnologias de desenvolvimento híbrido no módulo cliente, *frameworks* gráficos para desenvolvimento de

aplicativos de interfaces gráficas web mobile e interfaces gráficas nativas, pesquisa das tecnologias de geoprocessamento a ser utilizada na aplicação. Baseado que as ferramentas analisadas possuem diferentes funcionalidades, a análise qualitativa dos itens genéricos avaliados serão:

- Quantidade de componentes visuais disponíveis, bem como sua capacidade de customização;
- Emuladores e ferramentas de desenvolvimento (IDE, emuladores proprietários e ferramentas agregadas);
- Apoio da empresa em relação a qualidade da documentação e comunidade de desenvolvedores;

Por fim, para testar a qualidade das ferramentas, será desenvolvido um aplicativo utilizando as tecnologias selecionadas nas pesquisas.

1.4 Organização do trabalho

Este trabalho é composto pelos seguintes capítulos:

Capítulo 2: Introduz o leitor aos conceitos básicos sobre a arquitetura dos *frameworks* utilizados para o desenvolvimento de aplicações móveis híbridas, além de fornecer informações sobre os sistemas de informação geográfica (SIG).

Capítulo 3: Análise das tecnologias propostas e suas funcionalidades.

Capítulo 4: Estudo de caso com o desenvolvimento de uma aplicação móvel para visualização de dutos, utilizando as tecnologias analisadas no capítulo 3.

Capítulo 5: Considerações finais sobre o desenvolvimento do trabalho, conclusão e trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

O objetivo deste capítulo é introduzir o leitor aos conceitos básicos sobre as arquiteturas híbridas para o desenvolvimento de softwares para dispositivos móveis. Serão introduzidos também, conceitos de sistemas de informação geográfica que serão aplicados no desenvolvimento do aplicativo de estudo de caso.

A divisão deste capítulo consiste em: seção 2.1- apresenta o que é o desenvolvimento híbrido; 2.1.1- descreve a arquitetura híbrida web mobile; 2.1.1.1 - apresenta o *framework* Phonegap; 2.1.2 - descreve a arquitetura híbrida de interfaces gráficas nativas; 2.2 – descreve o que são sistemas de informação geográfica, quais são os seus objetivos e principais funcionalidades; a variação desta metodologia para ambientes móveis é abordada logo em seguida na seção 2.2.1.

2.1 Desenvolvimento híbrido em plataformas móveis

Os navegadores (*WEB Browsers*) em *smartphones* e tablets estão bastante evoluídos no quesito de compatibilidade com tecnologias WEB atuais como HTML5, CSS3 e javascript, que permitem a construção de aplicativos muito eficientes, visto o alto poder de manipulação e visualização dos dados. (HARREL, 2011).

O HTML é uma linguagem de marcação de hipertexto (do inglês *HyperText Markup Language*) e é utilizada para a criação de páginas da internet (WC3, 2009). O HTML foi criado originalmente por Tim Bernes-Lee em 1994 e utiliza *tags* (Etiquetas) para descrever a estrutura, aparência e sintaxe de um documento (BERNEES LEE, 1994). Fornece também, recursos para embutir outras linguagens como o javascript, que permite que documentos estáticos tornem-se dinâmicos.

O Javascript é uma linguagem de script utilizada por bilhões de páginas web para a adição de funcionalidades como validação de formulários, dinamismo e comunicação com servidores (W3SCHOOLS, 2012).

O benefício de implementar um sistema web é a interoperabilidade entre plataformas, visto que o navegador abstrai a camada de interpretação, fazendo com que a execução da aplicação seja independente da arquitetura e do sistema operacional utilizado (GARDNER, GRIGSBY 2011).

Para que a aplicação funcione de maneira eficiente em todas os navegadores dos dispositivos móveis, alguns *frameworks* fornecem recursos gráficos, que permitem que a aplicação seja moldada de forma flexível para que a experiência seja consistente em todos eles (JQUERYMOBILE, 2011). Segundo Fayad e Schmidt (1997), frameworks representam uma estrutura formada por blocos pré-fabricados de software que os programadores podem usar, estender ou adaptar para uma solução específica e linguagens de padrões.

Uma das deficiências da implementação de sistemas com tecnologia web, principalmente em ambientes móveis, é o acesso a recursos nativos do aparelho, como por exemplo, o GPS, acelerômetro, câmera entre outros recursos. Por outro lado, a implementação nativa de um mesmo software em cada uma das plataformas demandaria um custo maior, pois duplicaria o tempo de desenvolvimento. Os *frameworks* híbridos permitem a junção destes dois modelos, fazendo com que recursos nativos do aparelho possam ser acessados através da aplicação (PHONEGAP, 2011).

Frameworks como estes quebram paradigmas de programação, pois permitem a abstração de plataformas, não sendo necessário que se aprenda várias linguagens de programação, bem como todas as tecnologias envolvidas em cada um dos sistemas operacionais móveis, pois o foco é no aprendizado das linguagens de programação utilizadas no *framework*, pois toda a comunicação e chamadas de recursos no sistema operacional são realizadas por ele.

Um exemplo prático da importância que estes *frameworks* proporcionam para o mercado de desenvolvimento de aplicativos móveis é sistema operacional móvel iOS, para desenvolver aplicativos para a plataforma da empresa *Apple* é necessário que se tenha um computador específico, dificultando e impedindo por vezes o desenvolvimento de aplicativos para esta plataforma.

Basicamente, existem dois modelos principais de *frameworks* híbridos no mercado, os baseados em interfaces gráficas web (web UI) aliados ao PhoneGap, que permite o acesso a recursos nativos do aparelho, e a renderização dos componentes visuais é feita através de um navegador web embutido. E interfaces gráficas nativas (*Native UI*), que permitem que todos os componentes visuais sejam compilados para o respectivo componente da plataforma, gerando um código binário correspondente (WHINNERY, 2012).

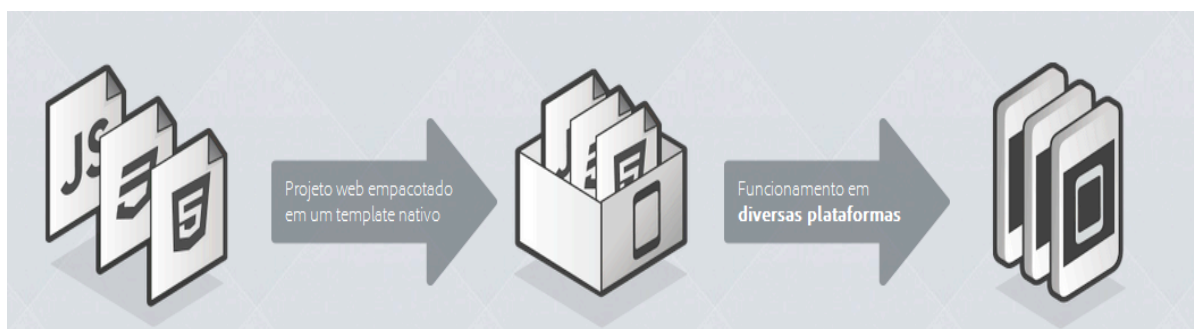
2.1.1 Interfaces gráficas web híbridas

A premissa da arquitetura baseada em web UI é a de permitir que uma aplicação criada com HTML, CSS e javascript possa acessar recursos nativos do celular, possibilitando a execução da aplicação para todas as plataformas nativas suportadas, sendo exibidas por meio do navegador WebKit.

WebKit é o motor gráfico de código-aberto utilizado pela maioria dos aparelhos móveis do mercado, ele permite que códigos HTML, CSS e javascript sejam interpretados e renderizados na tela do aparelho, utilizados geralmente por navegadores web (WEBKIT, 2012).

Uma das frases que definem este tipo de desenvolvimento é "*Write once, run everywhere*" ("Escreva uma vez, execute em todos"). A figura 1 demonstra o funcionamento de uma aplicação baseada em webUI.

FIGURA 1: Arquitetura web híbrida



FONTE: Adaptado de (PHONEGAP, 2012)

Um ponto importante a ser questionado neste modelo de desenvolvimento, é o desempenho de aplicativos web em relação a aplicativos nativos, pois aplicativos nativos conseguem concorrer a recursos de hardware com mais facilidade, visto que não há um software intermediário na execução, tais como um web browser (PALMIEIRI , SINGH, 2011).

Desenvolver um aplicativo utilizando o modelo web afeta o desempenho da aplicação já que não é gerado um código binário, e necessita de uma aplicação terceira para interpretar este conteúdo, a aplicação sempre roda através de um navegador e os recursos nativos (GPS, bússola entre outros) são acessados por meio do *framework* PhoneGap (WHINNERY, 2012).

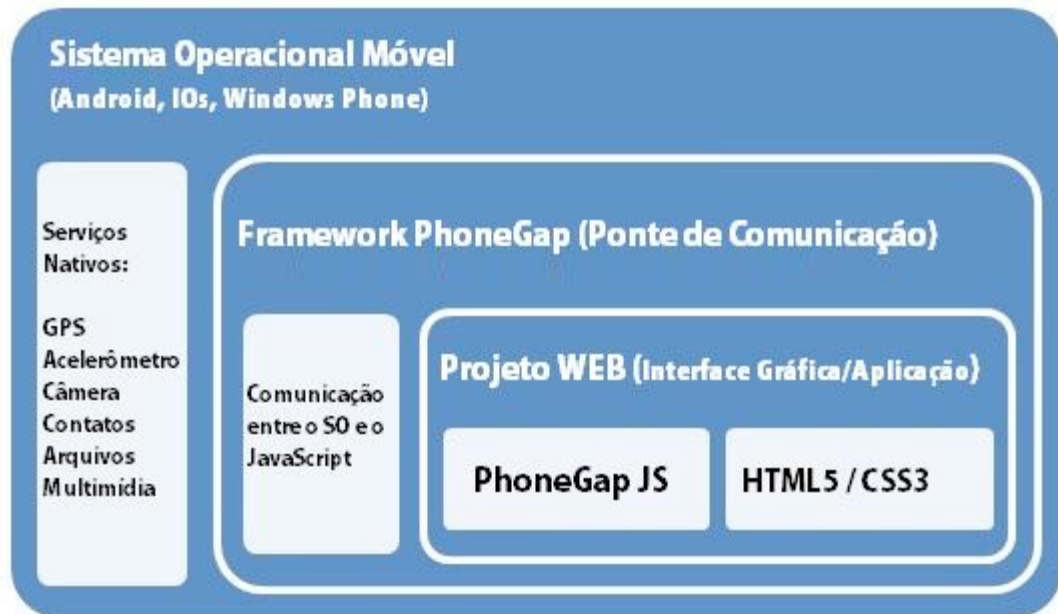
2.1.1.1 PhoneGap

PhoneGap é um *framework* Open-Source desenvolvido inicialmente pela empresa Nitobi, posteriormente adquirida pela empresa Adobe que continua com o desenvolvimento do *framework*. PhoneGap foi detentor do prêmio da feira *Web 2.0 Expo* em 2009, uma das principais feiras de tecnologia web do mundo, onde reúnem-se líderes empresariais, grandes pensadores e desenvolvedores inovadores que estão moldando o futuro da Web (WEB 2.0 EXPO, 2009).

A principal funcionalidade do PhoneGap é abstrair a camada de acesso aos componentes nativos do sistema operacional móvel, utilizando tecnologias web como HTML5 e JavaScript, é possível desenvolver um único sistema e implantá-lo em diferentes plataformas (ADOBE, 2011).

O funcionamento básico do PhoneGap é realizar a interface entre o sistema operacional nativo do aparelho móvel e o sistema implementado em HTML e javascript, logo, a comunicação entre os componentes envolvidos no sistema funciona como na figura 2.

FIGURA 2: Hierarquia da comunicação em um aplicativo utilizando PhoneGap.



FONTE: Adaptado de (FILANNINO, 2010)

As chamadas nativas do sistema operacional móvel são realizadas através do *framework* PhoneGap que se encarrega em realizar este pedido ao Sistema Operacional e retornar o resultado para a aplicação. O PhoneGap trata apenas da camada de serviços da aplicação, para o desenvolvimento da interface gráfica é necessário desenvolver um sistema baseado em tecnologias web como HTML, CSS e JavaScript, adequando os componentes visuais para plataformas móveis (WHINNERY, 2012).

Visto a inconsistência entre os diversos sistemas operacionais, como linguagens de programação e arquiteturas utilizadas, o PhoneGap possui uma mesma versão do *framework* disponibilizada para cada um dos sistemas operacionais suportados, logo, para fazer a execução do aplicativo na plataforma desejada é necessário que seja criado um projeto específico da plataforma utilizando o *framework* Phonegap correspondente a ela. A figura 3 apresenta as plataformas suportadas pelo PhoneGap.

FIGURA 3: Plataformas suportadas pelo PhoneGap (IOS, Android, Windows Phone, OS, WebOs, Symbian e Bada).

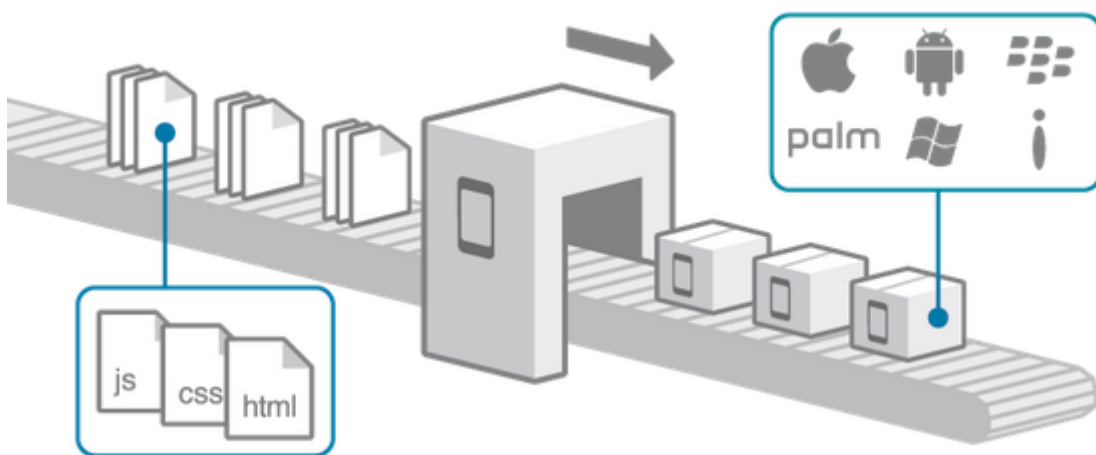


Fonte: (PhoneGap, 2012)

Uma solução interessante para realizar a compilação da aplicação sem ter a necessidade de ter o kit de desenvolvimento ou ferramentas específicas da plataforma, como por exemplo para IOS onde há a necessidade de se ter um mac, o *PhoneGapBuild* permite que isso seja possível através da “Compilação nas nuvens”.

O processo de compilação nas nuvens é simples e funcional, bastando apenas que o desenvolvedor faça o upload do projeto web para o servidor através de um arquivo .zip ou apontamento de repositórios como Git ou SVN, o servidor fará todo o trabalho de compilação e retornará a aplicação web "empacotada" em cada um dos *templates* de aplicações das plataformas disponíveis, gerando assim a aplicação pronta para a execução (Adobe, 2012).

FIGURA 4: Funcionamento do PhoneGap Build



FONTE: (PHONEGAPBUILD, 2012)

O *PhoneGapBuild* é gratuito apenas para projetos Open-Source. Nas compilações feitas para Android, IOS e BlackBerry é necessário que se adicione uma chave de certificação de autoria do aplicativo, desta forma ele estará pronto para ser distribuído nas respectivas lojas de distribuição de programas.

2.1.2 Interfaces gráficas nativas híbridas

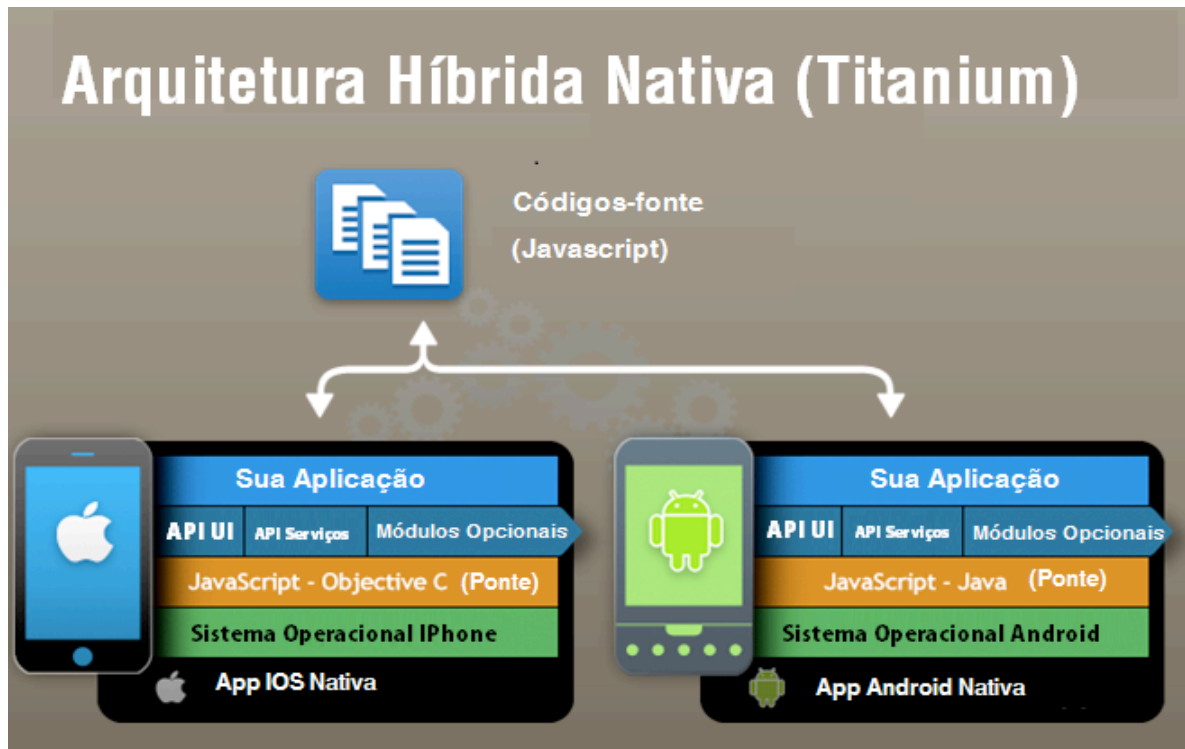
Desenvolver uma aplicação baseando-se no modelo de interfaces gráficas nativas permite que a aplicação tenha o melhor desempenho possível, pois os códigos-fonte visuais serão compilados para o código do respectivo sistema operacional de destino. Nesta arquitetura de programação, é necessário que se tenha o kit de desenvolvimento (SDK) de cada uma das plataformas desejadas, pois ele se faz necessário para a compilação do aplicativo (MOSYNC, 2011).

O lema da maioria destes *frameworks* é "*Write once, adapt everywhere*" ("escreva uma vez, adapte em todos"), ou seja, o código binário final será adaptado de acordo com a plataforma correspondente. (WHINNERY, 2012). O funcionamento destes *frameworks* são semelhantes, e a plataforma Titanium será utilizada para demonstrar este modelo arquitetural.

A aplicação é baseada em apenas um código-fonte, a linguagem de programação é definida pelo *framework* que no caso é o javascript, a interface gráfica da aplicação (*View*) será convertida para um código efetivamente nativo, já o código das demais funções serão executados através de um interpretador em tempo de execução, que fará a comunicação com a camada de serviços nativos (WHINNERY, 2012).

A camada de serviços nativos disponibilizadas pelo aparelho é realizado através de uma interface entre o sistema operacional e a linguagem do *framework* (Ponte), essa comunicação é realizada através de chamadas em tempo de execução entre o *framework* e o SO (TITANIUM, 2012). A figura 5 demonstra esse modelo de arquitetura.

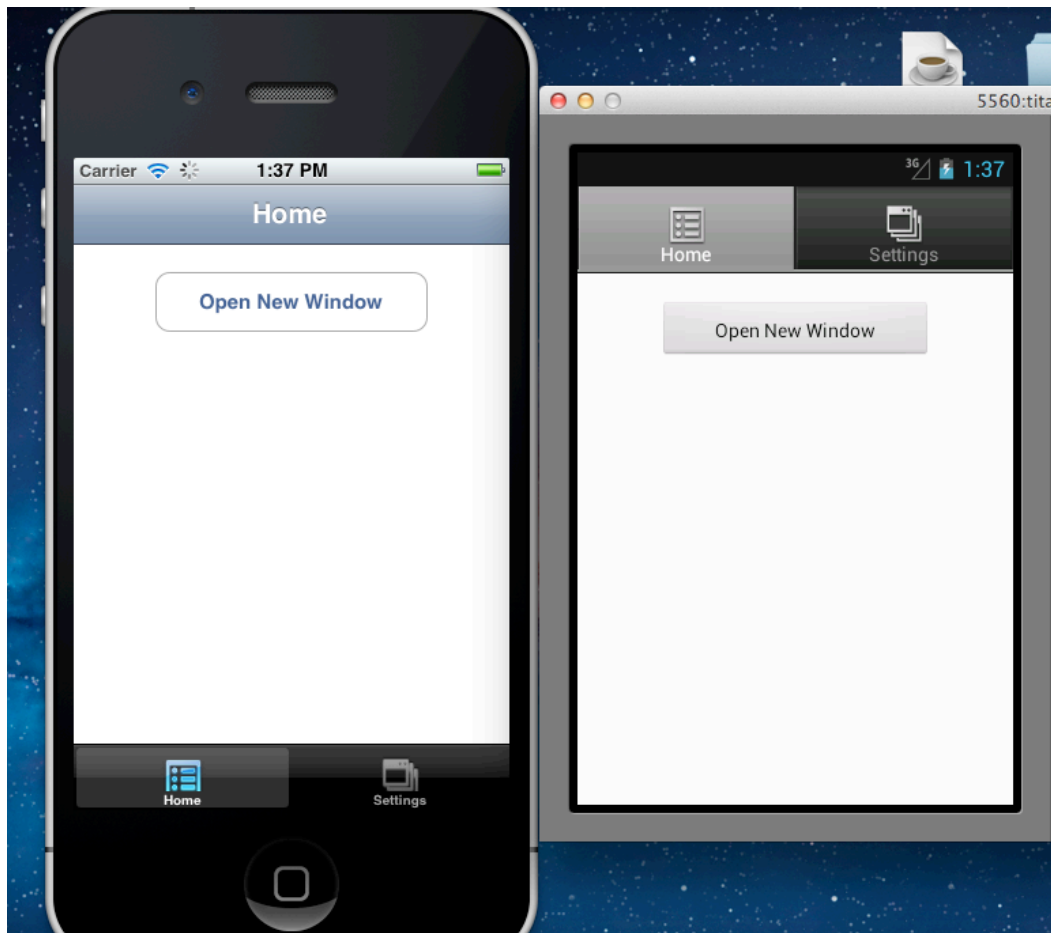
FIGURA 5: Modelo de arquitetura de interfaces gráficas nativas no Titanium.



Adaptado de (APPCCELERATOR, 2012)

Este modelo de desenvolvimento permite obter a flexibilidade da aplicação aliada ao desempenho da plataforma nativa, além disto, a aparência da aplicação será baseada na plataforma que está sendo executada, pois, o código visual será nativo. A figura 6 demonstra uma mesma aplicação sendo executada em diferentes plataformas com o mesmo código-fonte, utilizando o framework Titanium.

FIGURA 6: À Esquerda aparência no emulador do Iphone (ios 5.0.1), à direita aparência em emulador Android (4.0).



Percebe-se nesta figura que todos os componentes são os mesmos, porém com o visual do respectivo sistema operacional. A linguagem utilizada para a criação dos componentes é javascript e os componentes visuais são criados com a implementação do objeto à partir da API do Titanium.

Apesar da vasta diversidade de componentes visuais nativos, por vezes a aplicação necessita renderizar páginas HTML dentro da aplicação, seja ela remota ou local, o componente webview permite esta funcionalidade. A interatividade entre este componente e a plataforma nativa é completamente transparente, e a comunicação é efetiva, sendo que os dados podem ser transportados facilmente entre os dois contextos.

Neste trabalho será realizado a avaliação de alguns *frameworks* que utilizam as duas metodologias apresentas anteriormente e como estudo de caso uma aplicação móvel de informações geográficas será implementada.

2.2 Sistemas de Informação geográfica

Sistemas de informação geográfica, também conhecidos pela sigla SIG, são sistemas que permitem visualizar dados espaciais como mapas, imagens do solo, dados vetoriais entre outros. Um sistema de informação geográfica permite a visualização de imagens terrestres para um determinado fim, como por exemplo o controle de construções em uma cidade, também é possível visualizar dados vetoriais como dutos, cabos, canos ou objetos subterrâneos e terrestres, bem como realizar qualquer análise geográfica que se deseja obter alguma informação (CAMARA, 1996).

Segundo (BURROUGH, 1998) os sistemas de informação geográfica são essenciais para profissionais que utilizam dados geográficos para tomada de decisão, como profissionais que atuam nas áreas de planejamento urbano, engenharia civil, engenharia ambiental, geografia, ecologia entre diversas outras áreas que utilizam as informações geográficas para a extração de dados para o seu negócio.

O uso de SIGs é um importante meio de se obter informações para a resolução de problemas, cálculos estatísticos, etc. Uma característica cada vez mais interessante é o uso no meio educacional e entretenimento, visto que os SIGs estão cada vez mais acessíveis com o desenvolvimento deste tipo de aplicativos em tecnologias web, que dispensam a instalação do software em uma máquina local (SMITH, 2009).

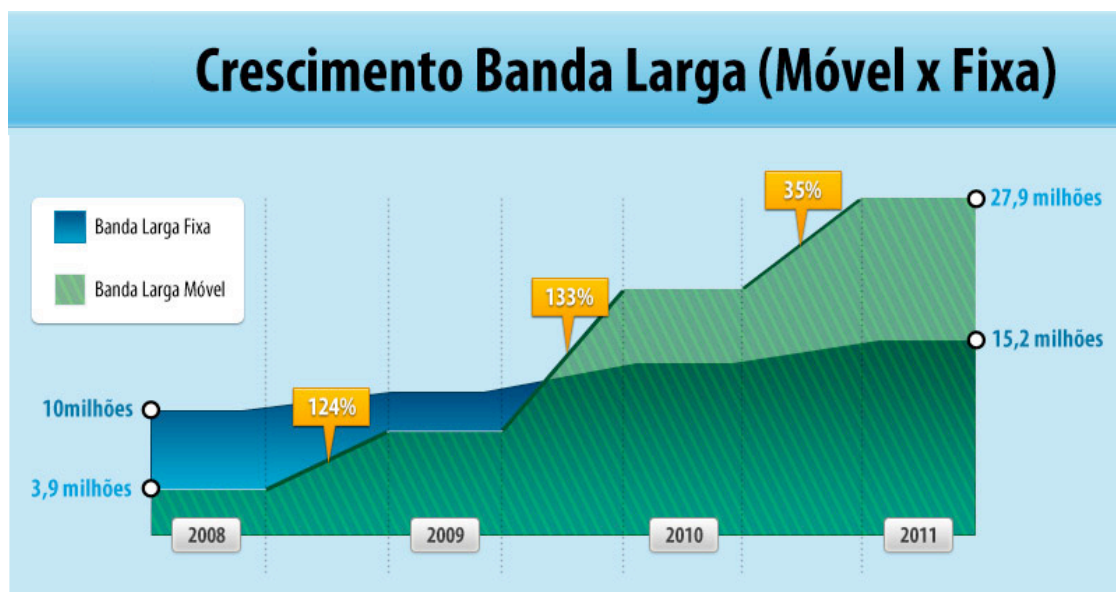
2.2.1 SIGs Móveis

Dispositivos móveis evoluem de maneira constante, a medida que o poder de processamento bem como o acesso a rede crescem, a interação de dispositivos como celulares e tablets com a internet facilita o acesso à informações remotas, o que impulsiona o uso destes

dispositivos. Estas facilidades proporcionam um cenário ideal para a implementação de SIGs em ambientes computacionais móveis.

Um estudo realizado pelo grupo Extra (2012), baseado em dados do governo e da Agência nacional de Telecomunicações, analisa que o número de usuários que possuem internet móvel é de 12.7 milhões maior do que o número de usuários de banda larga fixa. A figura 7 demonstra este cenário:

FIGURA 7: Infográfico do uso da internet no Brasil.



Fonte: Adaptado de (EXTRA, 2012)

Fangxiong (2004) denomina SIGs Móveis como a integração entre sistemas de informação geográfica no ambiente de computação móvel, neste contexto a computação móvel é um artifício bastante interessante quando se deseja analisar, manipular e usufruir de todas as funcionalidades de um SIG com mobilidade.

A computação móvel faz com que o usuário esteja inserido em diferentes contextos, algumas informações podem ser utilizadas pela aplicação para oferecer uma interatividade dinâmica, este paradigma da computação móvel é denominado de *Context-Aware* ou aplicações baseadas em contextos (CHEN, KOTS, 2000). A localização do usuário, hora do dia, pessoas e dispositivos próximos, entre outras informações relevantes que o sistema pode oferecer, são exemplos práticos da utilização deste paradigma (SCHILIT, ET AL, 1994). No

estudo de caso desenvolvido neste trabalho, a localização geográfica do usuário é utilizada para exibir os gasodutos mais próximos a ele.

2.2.2 Arquitetura Cliente Servidor

O modelo Cliente x Servidor em uma estrutura computacional é um sistema dividido entre clientes e servidores, sendo que o cliente faz requisições de determinados serviços ao servidor. O benefício de utilizar esta arquitetura no SIG móvel é que grande parte do processamento é realizado no servidor, que apenas recebe a requisição do cliente, processa o pedido e retorna o resultado. O cliente por sua vez será apenas um visualizador de dados, pois receberá as imagens ou dados processados pelo servidor e plotará o resultado na tela do aparelho móvel.

Outra característica interessante na utilização da arquitetura cliente x servidor é a interoperabilidade que os *WebServices* proporcionam para a aplicação. Zanelato (2010) define *Web Services* como serviços disponibilizados na web que possibilitam a comunicação entre sistemas distintos, a definição dos dados é feita através de alguma linguagem de marcação, geralmente o XML e o transporte a partir do protocolo HTTP, logo, os *Web Services* permitem que um cliente implementado sobre qualquer sistema operacional ou qualquer linguagem de programação possa utilizar o mesmo serviço.

Um dos problemas na utilização desta arquitetura é que o módulo cliente deve estar conectado de alguma maneira com o servidor, na maioria das vezes através da internet. As redes de comunicação móveis como 3G evoluem, mas o uso deste tipo de tecnologia deve ser planejado, visto que em algumas regiões ele pode não estar disponível.

3 AVALIAÇÕES DAS TECNOLOGIAS

Neste capítulo serão apresentados as características que levaram a escolha das ferramentas utilizadas para o desenvolvimento do aplicativo de caso de uso. Todos os testes foram realizados no aparelho Iphone3GS, baseado nos aplicativos disponibilizados como referência no site de cada um dos frameworks.

O critério escalar para medir o desempenho das aplicações foi baseado na experiência de uso da aplicação como usuário, e principalmente na agilidade da aplicação em relação às respostas aos toques na tela.

O principal objetivo desta avaliação é determinar qual tecnologia se encaixará nos requisitos propostos para a solução do sistema móvel de visualização de gasodutos. Todas as tecnologias serão avaliadas principalmente nos quesitos desempenho e recursos disponíveis. Ao final, as tecnologias serão selecionadas de acordo com os critérios expostos na avaliação individual de cada uma delas.

Este capítulo está organizado da seguinte forma:

- a) Seção 3.1 Avalia as tecnologias de visualização de mapas.
- b) Seção 3.2 Apresenta diversas tecnologias híbridas, expondo suas principais características bem como os seus pontos fortes e fracos.
- c) Seção 3.3 Finaliza a abordagem de todas as tecnologias, e seleciona o *framework* baseado nas avaliações dos subcapítulos anteriores.

3.1 Bibliotecas de Visualização de Mapas

Nos Seção à seguir, serão avaliados as tecnologias para a visualização de mapas, todas as soluções adotadas para o comparativo utilizam a linguagem de programação javascript e possui licença de código-aberto. As escolhas pelas bibliotecas mais adequadas, basearam-se no desempenho, quantidade de recursos disponíveis e implementação dos padrões OGC.

Open GeoSpatial Consortium (OGC) é um comitê aberto formado por mais de 438 empresas e universidades de todo o mundo. Este consórcio visa o estabelecimento de padrões que permitem interoperabilidade no setor geoespacial (OGC, 2011). Segundo Câmara (2005), as especificações promovidas pelo OGC trazem uma solução interessante com relação a intercâmbio de dados geográficos, em ambientes com tecnologias SIG heterogêneas.

3.1.2 OpenLayers

Open Layers é uma biblioteca javascript utilizada para facilitar a exibição de mapa, ele pode exibir componentes especiais cartográficos como marcadores carregados a partir de qualquer fonte (OpenLayers, 2006).

O Framework é completamente livre e de código-aberto, disponibilizado a partir da licença FreeBSD, seu objetivo é o de promover o uso da informação geográfica de todos os tipos e segue todos os padrões estabelecidos pelo OGC (OpenLayers, 2006).

Este é um projeto da *Open Source Geospatial Foundation* (OSGEO) fundação sem fins lucrativos, criada para apoiar o desenvolvimento colaborativo de software de código-aberto GeoEspacial (Open Source Geospatial Foundation, 2006).

O OpenLayers apresenta um nível de maturidade interessante em relação à visualização de mapas. Nos testes realizados apresentou desempenho razoável na maior parte do tempo, além de possuir integração e exemplos com aplicativos móveis em sua documentação oficial.

A rapidez com que a aplicação foi executada no aparelho móvel não foi tão satisfatória, elementos visuais tiveram comportamentos não esperados além de lentidões que também foram identificadas. As figuras 8 e 9 demonstram os testes realizados com as aplicações oficiais do OpenLayers, utilizando JQueryMobile e Sencha Touch respectivamente.

FIGURA 8: Aplicação utilizando OpenLayers com JQueryMobile.



FONTE: (OPENLAYERS, 2012)

FIGURA 9: Aplicação utilizando OpenLayers com Sencha Touch 2.



FONTE: (OPENLAYERS, 2012)

Os testes foram realizados no navegador Safari do aparelho móvel iPhone3GS e em ambas as aplicações foram detectados em alguns momentos falhas na renderização visual, principalmente em alguns componentes que ficaram fora do local original, dificultando por vezes, a utilização do software..

3.1.3 Leaflet

Leaflet é um moderno *framework* para visualização de mapas na web, com a premissa de uma “folha de árvore” em alusão à sua leveza (apenas 27KB em sua versão minimizada), a biblioteca Leaflet é escrita em javascript, seu principal objetivo é o de fornecer recursos de

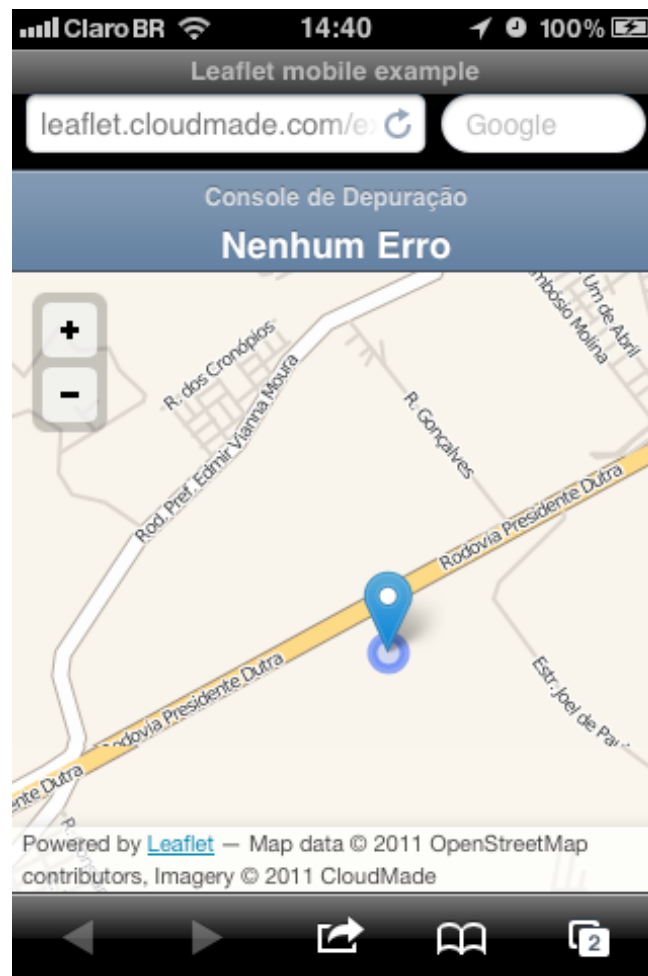
visualização de mapas de maneira rápida e rica em detalhes para o usuário final, principalmente em ambientes móveis (LEAFLET, 2012).

O desenvolvimento da API foi projetado para ser modular, extensível e fácil de entender, a biblioteca foi construída para fornecer eficiência na execução de aplicativos web em desktops e principalmente em aparelhos móveis, utilizando recursos como HTML5 e CSS3 para a renderização dos componentes visuais, além de Implementar os padrões OGC WMS e GeoJSON (LEAFLET, 2012).

Leaflet também implementa o padrão OGC WMS de visualização de mapas, apresenta menos recursos que o OpenLayers, porém, segundo os desenvolvedores, o Leaflet não tenta fazer tudo para todos, o seu objetivo é fazer com que os requisitos básicos funcionem da maneira mais eficiente o possível, além de ser flexível para a escrita de novos plugins (LEAFLET, 2012).

Este framework suporta nativamente aplicativos web *mobile*, no site oficial constam apenas um aplicativo para testes, que foi executado com excelente desempenho. A figura 9 demonstra o uso de um aplicativo de mapas para aparelhos móveis disponível no site oficial.

FIGURA 10: Exemplo de mapa para aparelhos móveis disponível no site oficial do *framework*.

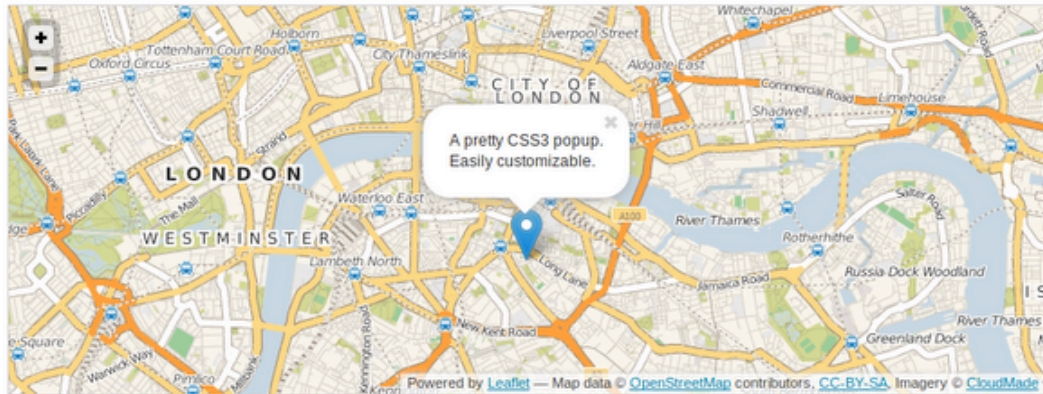


FONTE: (LEAFLET, 2012)

A lista de componentes disponíveis no *framework* está descrita no site oficial, a documentação da API é interessante pois demonstra o exemplo e, a seguir o seu respectivo código, representado na figura a seguir.

FIGURA 11: Exemplo de criação de mapa com uma PopUp de informações.

Basic usage example



Here we create a map with tiles of our choice, add a marker and bind a popup with some text to it:

```
// initialize the map on the "map" div
var map = new L.Map('map');

// create a CloudMade tile layer (or use other provider of your choice)
var cloudmade = new L.TileLayer('http://{s}.tile.cloudmade.com/[your-API-key]/997/256/{z}/{x}/{y}.png', {
  attribution: 'Map data &copy; [-]',
  maxZoom: 18
});

// add the CloudMade layer to the map set the view to a given center and zoom
map.addLayer(cloudmade).setView(new L.LatLng(51.505, -0.09), 13);

// create a marker in the given location and add it to the map
var marker = new L.Marker(new L.LatLng(51.5, -0.09));
map.addLayer(marker);

// attach a given HTML content to the marker and immediately open it
marker.bindPopup("A pretty CSS3 popup.<br />Easily customizable.").openPopup();
```

FONTE: (LEAFLET, 2012)

Leaflet apresentou funcionalidades interessantes, fatores como tamanho compacto e suporte para ambientes móveis são um atrativo para o uso desta ferramenta.

3.2 Frameworks de Interface Gráfica

3.2.1 Frameworks Web Móvel

As bibliotecas de Interface gráfica avaliadas foram: JQueryMobile, SenchaTouch, Jo e JQTouch, os testes basearam-se nas demonstrações oficiais disponíveis no site de cada uma das empresas.

3.2.1.1 JQuery Mobile

JQuery é um *framework* desenvolvida em javascript que permite que uma série de efeitos visuais e animações sejam incrementadas em páginas web (JQUERY, 2009). baseado no conceito “*Write less, do more*” ou seja “Escreva menos, faça mais”. O JQuery permite que uma série de recursos sejam implementados na aplicação, sem demandar um esforço demasiado do programador, é de código-aberto e sua licença de uso é GPL (*General Public License*).

No *framework* móvel do JQuery uma característica bastante interessante é que ele permite que um mesmo aplicativo web possa ser compatível com todas as plataformas móveis populares do mercado, isso permite que não seja necessário que a aplicação tenha um tratamento específico na interface gráfica para cada uma das plataformas. (JQUERY MOBILE, 2010).

FIGURA 12: Plataformas suportadas pelo JQueryMobile.



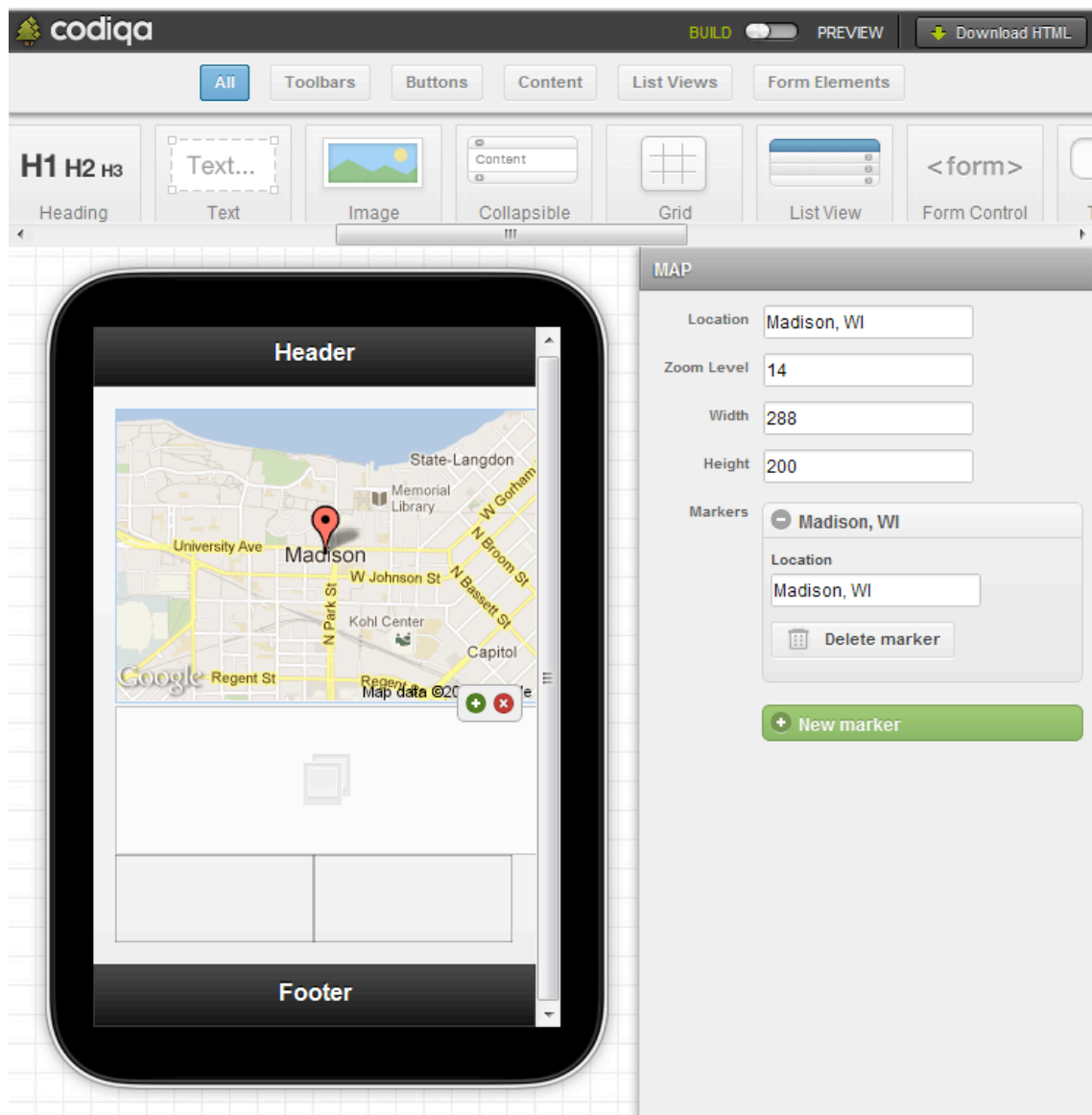
FONTE: (JQUERYMOBILE, 2012)

JQuery Mobile apresenta um leque de recursos muito interessantes, a documentação do sistema é bem elaborada e demonstra todas as funcionalidades implementadas no *framework*. Um fator negativo é o versionamento das documentações, em diferentes versões dos lançamentos, a versão anterior continha determinada funcionalidade que o novo versionamento não possuía, e vice-versa.

O *framework* utiliza HTML5 e CSS3 para incorporar elementos visuais e eventos aos seus widgets, sua principal característica é o uso das marcações data-attributes que são elementos sintáticos do HTML5 que permitem a adição de dados, eventos e efeitos baseados somente no tipo de marcação realizada (REID, 2011).

Uma ferramenta de prototipação presente no site oficial do JQuery Mobile da empresa codiga, permite a rápida criação de aplicativos utilizando todos os widgets presentes no JQueryMobile (CODIGA, 2012). O que mais chama a atenção foi a qualidade do código gerado que não utiliza componentes desnecessários, viabilizando o uso da ferramenta não só como prototipação mas como ferramenta auxiliar na produção do software.

FIGURA 13: Ferramenta de prototipação do JQuery Mobile.



FONTE: (JQUERYMOBILE, 2012)

Além da documentação oficial o *framework* demonstra por meio de aplicativos reais o uso da ferramenta, disponível em <http://www.jqmgallery.com/>. Todos os exemplos demonstram de

forma prática as diferentes formas de se utilizar os widgets (JQUERYMOBILE, 2012), além de existir a possibilidade da inspeção do elemento para visualizar os seus comportamentos e códigos-fonte, um auxílio interessante à documentação oficial.

Um dos pontos negativos detectados na utilização do JQuery Mobile foi a renderização dos componentes visuais, que não apresentou um desempenho adequado, além de que bugs com efeitos de transições de tela também foram identificados, deficiência esta que possivelmente será corrigida nas próximas versões.

Por fim, o JQueryMobile apresentou-se um *framework* interessante, mas para aplicativos para a produção ele não se saiu como uma alternativa viável, pois seu desempenho na renderização visual não foi adequado. A arquitetura JQueryUI e JQuery são um complemento essencial na utilização do *framework*, pois possui funções de apresentação e comunicação imprescindíveis para o desenvolvimento das aplicações.

3.2.1.2 Sencha Touch 2

Semelhante ao JQueryMobile, o SenchaTouch 2 é um *framework* que permite a construção de aplicativos web móveis híbridos, utilizando HTML5, CSS3 e javascript. Sua arquitetura permite que qualquer aplicação criada com este *framework* funcione de maneira eficaz em todas as plataformas móveis, como celulares e tablets que possuem os sistemas operacionais Android, Ios, BlackBerry, WindowsPhone entre outros.

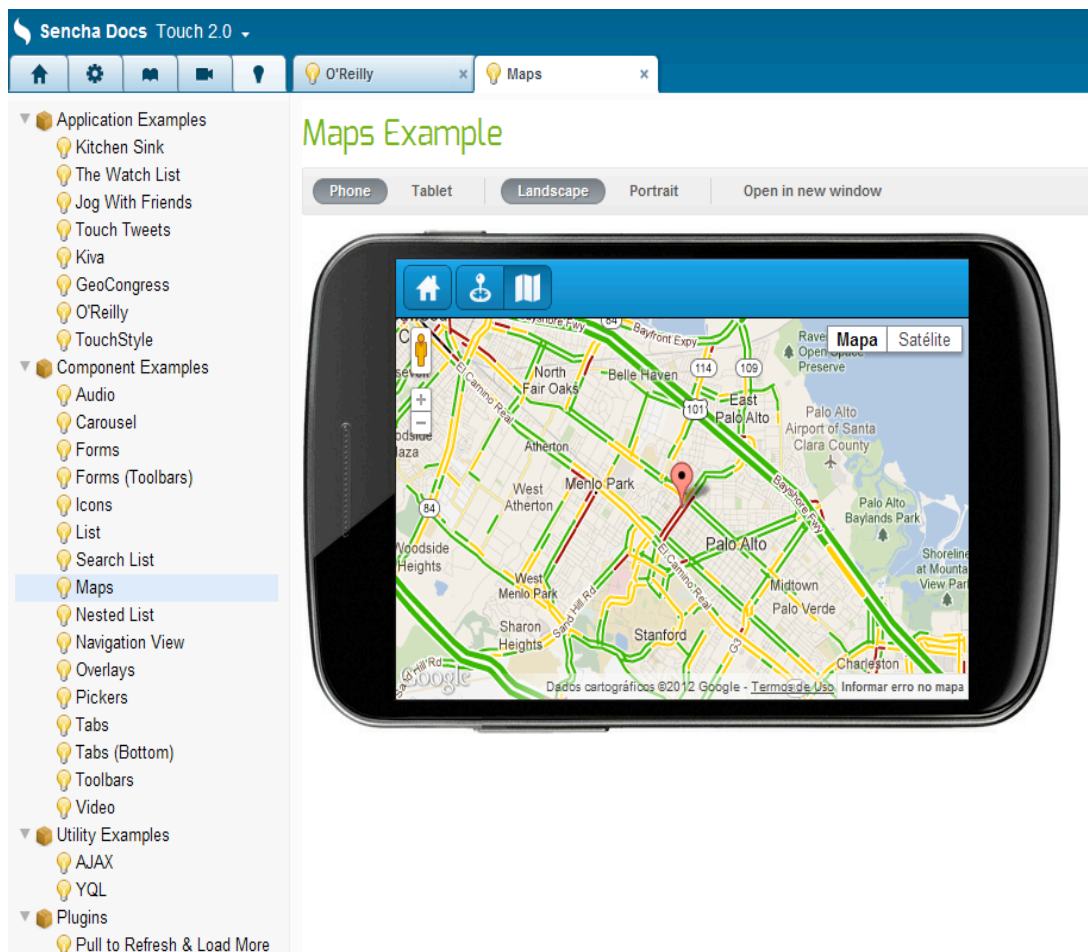
A API Sencha Touch possui uma sólida arquitetura MVC, é fácil de aprender, é rápida e possui uma documentação rica (Sencha, 2012). Seu uso é livre e o código-fonte do *framework* pode ser modificado pois é aberto. A principal fonte de recursos da empresa é a sua licença de manutenção e suporte.

Uma das preocupações no uso deste *framework* é a questão da licença de uso dos produtos Sencha, alguns produtos da empresa possuem licença comercial. Uma preocupação decorrente disto é que a decisão da licença de uso do SenchaTouch é exclusivo da empresa que mantém o *framework*.

Um dos recursos que apresentam um maior destaque no uso desta ferramenta são as widgets de visualização, com aproximadamente 50 componentes pré-construídos (SENCHA, 2012). A experiência do usuário ao utilizar um sistema com SenchaTouch é rica e muito bem detalhada, porém, nos testes de desempenho realizados, foram detectadas lentidões no carregamento das páginas, mesmo com metodologias de redução e ofuscamento do código-fonte.

Outro componente que chama a atenção é sua documentação, que é muito completa, apresentando os seus recursos em quatro partes, como representado na figura a seguir:

FIGURA 14: Figura da Documentação do Sencha Touch 2



FONTE: (SENCHA, 2012).

- a) Documentação da API: demonstra todos os eventos, componentes, classes e utilidades presentes em seu framework;

- b) Guias: Tutoriais que demonstram passo-a-passo sobre como uma aplicação deve ser construída. Dividida por módulos e devidamente pensada para que desenvolvedores iniciem pelo nível mais básico e evolua conforme a leitura;
- c) Vídeos: auxiliam o programador com vídeo-aulas e palestras referentes ao desenvolvimento na plataforma;
- d) Exemplos: diversos exemplos de aplicações reais que são demonstradas em um aparelho móvel virtual. Um dos pontos negativos desta demonstração é que o javascript é ofuscado, técnica que melhora o desempenho da aplicação e dificulta a leitura do código para humanos, logo, não é possível inspecionar os elementos da aplicação em tempo de execução.

Utilizando a arquitetura MVC o *framework* apresenta uma curva de aprendizado grande, pois o programador deve ter um conhecimento mais aprofundado na linguagem javascript para entender como utilizar cada um dos componentes.

Licenciado sob Código-Aberto, o Sencha Touch é uma ótima opção para desenvolvimento, porém, sua vocação é empresarial e dependendo do seu uso existe cobrança para o desenvolvimento das aplicações. A dificuldade de uso e grande curva de aprendizado podem dificultar o andamento de um projeto com uma equipe que não tenha um amplo conhecimento em javascript.

3.2.1.3 Jo

Jo é um *framework* para o desenvolvimento de aplicações móveis web, foi desenvolvido principalmente para o desenvolvimento de aplicações e não sites (JO, 2012). Utilizando CSS3 este *framework* permite que as modificações visuais sejam feitas de maneira simples e livre.

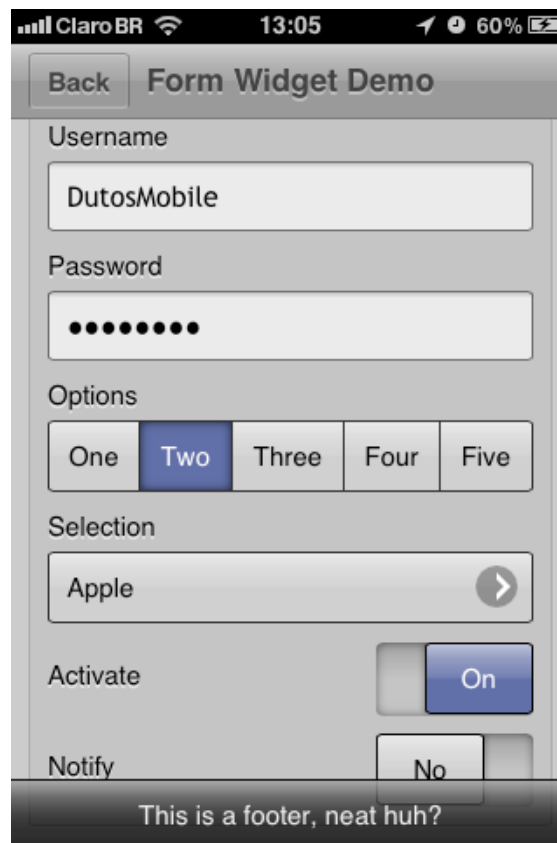
Segundo Balmer (2012) criador da ferramenta, o Jo possui as seguintes características:

- Utiliza o poder do CSS3 para lidar ao máximo com interfaces ricas e animações.
- É bem estruturado para funcionar como aplicação web mobile com o PhoneGap.
- Não utiliza muitos recursos, além de ser interoperável entre navegadores web.

- Não requer conhecimento de DOM (*Document Object Model*), que é uma especificação do W3C para que documentos web possam ser dinamicamente alterados (W3C, 2012),
- Utiliza identificadores próprios para seus atributos, logo, a estrutura do *framework* é mais organizada.

A quantidade de componentes presentes no *framework* não é tão extensa em relação aos demais *frameworks* analisados, Jo está em fase de amadurecimento e necessita da inclusão de melhorias nos temas nativos, para que a visualização dos aplicativos seja mais agradável.

FIGURA 15: Exemplo de Widgets no JO



FONTE: (JO, 2012)

No site oficial do framework existem apenas quatro exemplos a respeito do uso dos *widgets* disponíveis. A documentação é bem estruturada e possui formato de livro, descrevendo o uso da API em capítulos.

A biblioteca é pequena e nos testes realizados o desempenho foi relativamente satisfatório, porém, as demais bibliotecas apresentaram uma quantidade de recursos mais interessantes e variadas.

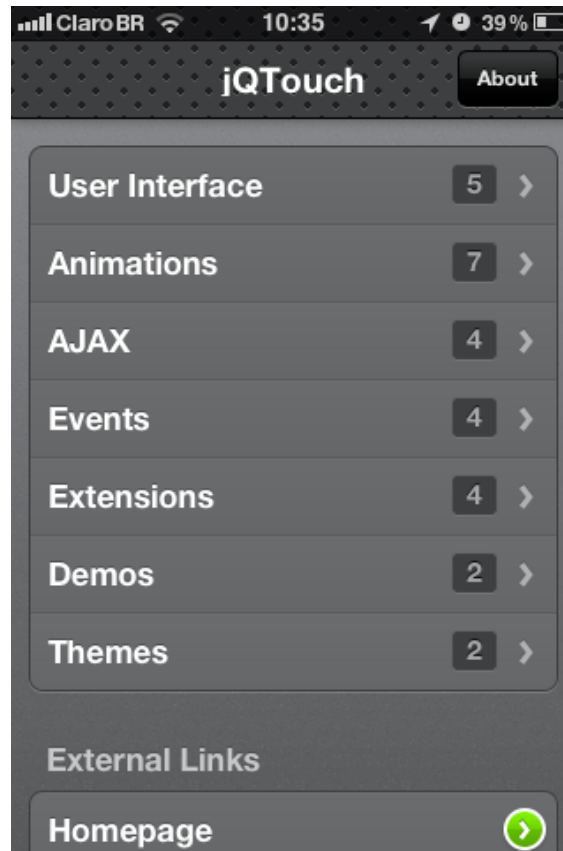
3.2.1.4 JQTouch

JQTouch é uma implementação da biblioteca ZeptoJS com a adição de funcionalidades do JQuery (JQTOUCH, 2012). ZeptoJS é uma biblioteca minimalista javascript para navegadores modernos, seu principal objetivo é o de prover funcionalidades e efeitos gráficos para navegação em aparelhos móveis (ZEPTO, 2012).

JQTouch alia a qualidade das funcionalidades implementadas no JQuery com o desempenho razoável da biblioteca ZeptoJS que é extremamente leve, apresentando em sua menor versão apenas 9KB (ZEPTO, 2012), característica interessante, visto que o tamanho dos arquivos no ambiente móvel é de vital importância, dado a quantidade de recursos serem limitadas.

Apesar deste *framework* ter um desempenho razoável, em alguns momentos a aplicação deixou de responder aos toques na tela, além de demonstrar certa lentidão em algumas transições, a Figura 16 demonstra o teste realizado em forma de *WebApp* no aparelho iPhone 3GS.

FIGURA 16: *WebApp* de teste dos widgets disponíveis no JQTouch.



FONTE: (OPENLAYERS, 2012)

Zepto mostrou-se a biblioteca mais leve em relação às demais bibliotecas HTML5 pesquisadas, dentre os *frameworks web mobile* esta é uma das bibliotecas mais viáveis para se desenvolver aplicações.

3.2.2 Frameworks de Interfaces gráficas nativas

3.2.2.1 Titanium

Titanium Mobile é uma plataforma híbrida nativa que permite que a aplicação seja compilada para as plataformas Android e IOS, além de um esforço inicial para BlackBerry. Utiliza a linguagem de programação javascript, mas, apesar desta premissa os aplicativos não são executados por meio de um navegador, em vez disto a arquitetura da linguagem de programação é utilizada para interpretar os códigos desenvolvidos (EGENIAL, 2012).

O funcionamento básico de uma aplicação Titanium é uma base de código javascript que é interpretada em tempo de execução por um interpretador que é embutido na aplicação no momento de sua compilação. Já o código-fonte visual é compilado para a respectiva plataforma, mantendo o comportamento e aparência da plataforma de destino (WHINNERY, 2012), como explícito no capítulo de arquiteturas híbridas de interfaces gráficas nativas.

Em uma aplicação Titanium grande parte do código-fonte é utilizado por ambas as plataformas, mas como cada sistema operacional possui alguns comportamentos e funcionalidades específicas, estes códigos podem ser tratados de maneira separada, de forma a permitir a exploração de determinada funcionalidade sem comprometer o restante da aplicação (APPCCELERATOR, 2011).

O Titanium é open-source e o seu código-fonte está disponível em <https://github.com/appcelerator>, além da ferramenta em si, diversos códigos de aplicações de templates podem ser utilizadas como forma de iniciação ao aprendizado.

Um fator que diferencia o Titanium das demais plataformas é a facilidade de customização dos componentes visuais, que é refletida na qualidade final da aplicação desenvolvida. O portfólio <http://www.builtwithtitanium.com/> "Built with Titanium" (construído com Titanium) permite analisar uma quantidade razoável de aplicativos que foram desenvolvidos utilizando a plataforma.

Uma loja de componentes, chamada Marketplace, permite a comercialização de componentes para a plataforma, seja ela desktop (componentes para a IDE) ou para *mobile*, como componentes visuais, bibliotecas, entre outros.

No site da empresa Appcelerator (<http://www.appcelerator.com/developers>), uma vasta documentação se faz presente, contendo a análise detalhada de todos os componentes seguidos de seu exemplo prático. Além da documentação comum, existe um treinamento gratuito para a certificação de desenvolvedor Titanium, que mostra a fundo a arquitetura da aplicação e seus derivados.

A empresa Appcelerator é parceira da international data corporation (IDC) e anualmente faz pesquisas sobre o mercado de desenvolvimento móvel, mostrando através de indicadores, os

dados relevantes sobre os rumos que o mercado móvel está sendo direcionado. Todas as pesquisas são abertas e encontram-se disponível no site <http://www.appcelerator.com/thinkmobile/surveys>.

3.2.2.2 MoSync

O *framework* Mosync permite a conversão do código-fonte para nove plataformas, essa premissa especificada pelo fabricante do software pode não ser tão relevante, visto que a quantidade de componentes visuais das plataformas é mais escassa em relação aos demais frameworks, além da deficiência de componentes que funcionam em apenas uma plataforma.

O desenvolvimento de aplicativos pode ser feito nas linguagens C++/C e JavaScript sendo que ambas as plataformas visam gerar a mesma aplicação. Porém para obter uma aplicação melhor trabalhada é necessário que se tenha uma curva de aprendizado em C++, pois a API disponível para a linguagem javascript é mais escassa em relação à API de C++. O desenvolvimento das aplicações podem ser baseados tanto em interfaces gráficas nativas quanto em webviews.

MoSync oferece uma IDE baseada em eclipse e os testes da aplicação podem ser feitos através do emulador nativo de cada uma das plataformas móveis de destino, ou com o uso da ferramenta Reloaded, que permite que as aplicações sejam testadas diretamente no aparelho, por uma injeção de código em tempo de execução. A aplicação Reloaded deve ser instalada no aparelho real, o SDK do MoSync injeta o bytecode de sua aplicação dentro da aplicação Reloaded que a executa em qualquer um dos aparelhos. Disponível para as plataformas Android, IOS e Windows Phone.

Poucas aplicações de interfaces gráficas nativas estão disponíveis como exemplo no portfólio da ferramenta. As figuras 17, 18 e 19 demonstram uma aplicação que obtém informações do dispositivo do qual ela está sendo executada, além de construir a interface gráfica com componentes visuais nativos.

FIGURA 17: Primeira tela, aplicação de exemplo da plataforma MoSync, sendo executada nas plataformas Android e IOS, alguns dados do sistema operacional sendo exibidos.

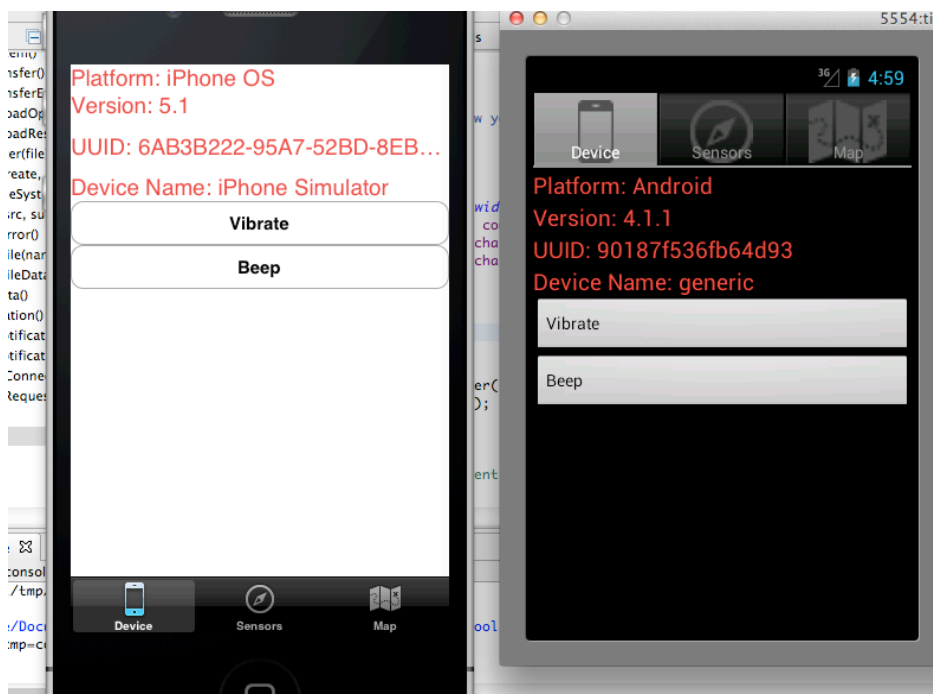


FIGURA 18: Segunda tela da aplicação, algumas medições extraídas de recursos nativos da plataforma.

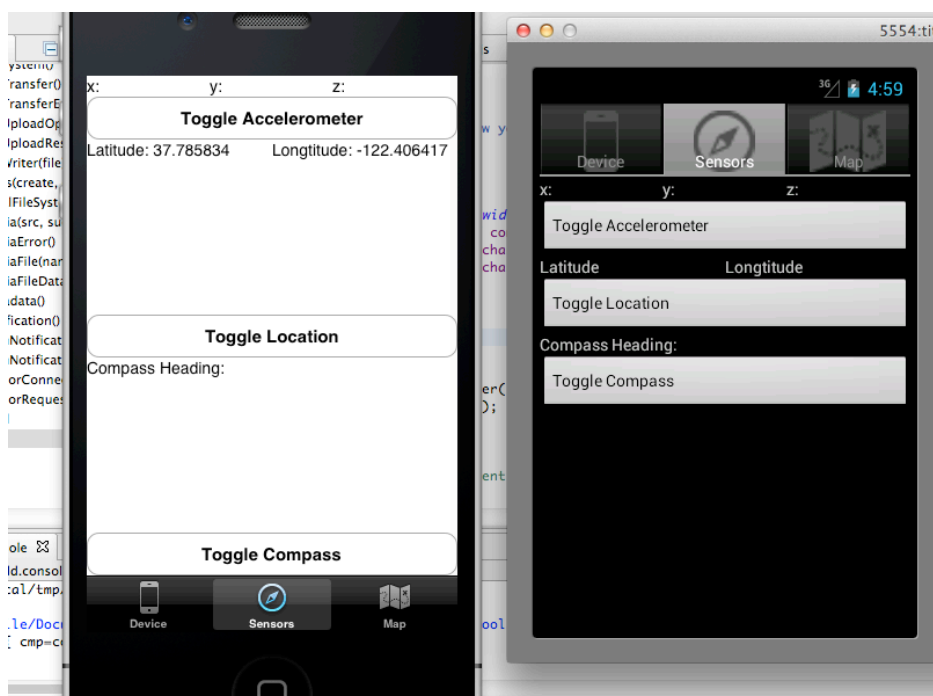
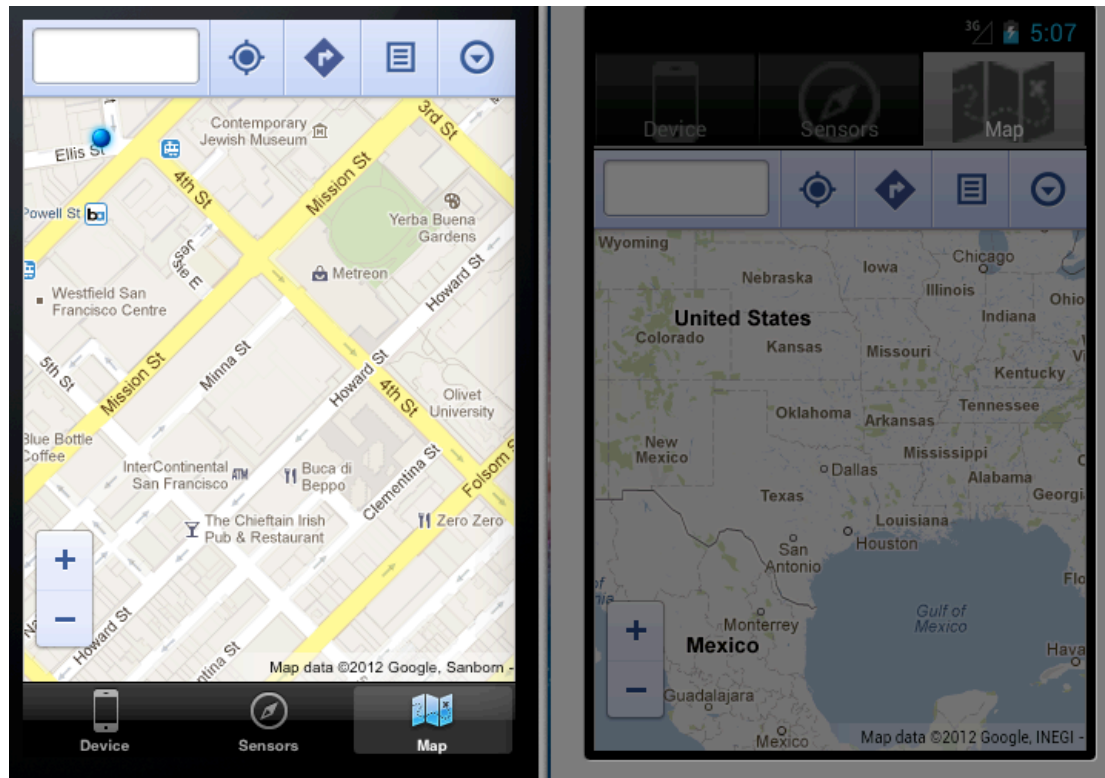


FIGURA 19: Terceira tela da aplicação, webview que exibe um mapa na terceira tela da aplicação.

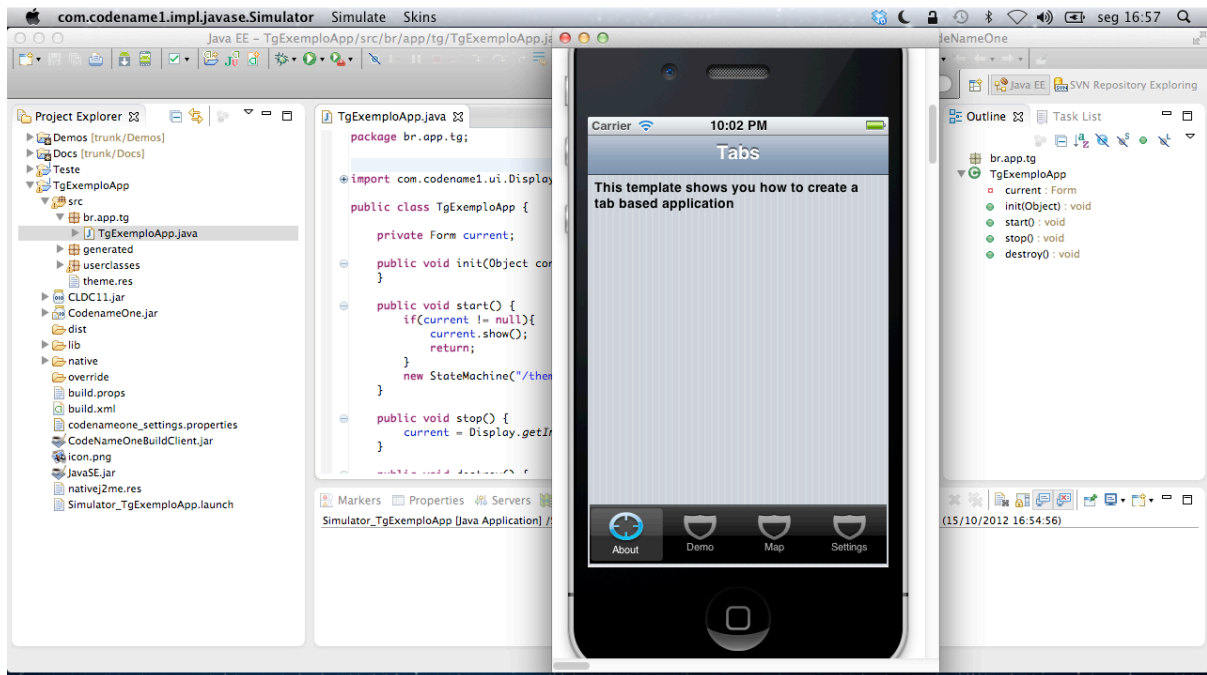


O Projeto é de código aberto e todos os códigos-fonte, tanto do SDK como exemplos de aplicações estão em <https://github.com/MoSync>. No site oficial da ferramenta (<http://www.mosync.com/>) está a documentação da API que não é tão bem estruturada, pois existem poucos exemplos práticos e códigos prontos para interfaces gráficas nativas, além de que poucas aplicações reais foram encontradas para desenvolver baseado em exemplos.

3.2.2.3 CodeNameOne

O *framework* Codenameone é diferente das demais ferramentas analisadas, ele não necessita do SDK de nenhuma das plataformas móveis, pois, o desenvolvimento bem como os testes são efetuados sobre o emulador do próprio *framework*. Esta funcionalidade é positiva pois não necessita do SDK de plataformas fechadas como a Apple, a qual necessita de um computador específico para efetuar os testes. Em contrapartida os comportamentos da aplicação eventualmente serão como os da própria plataforma.

**FIGURA 20: TESTES de uma aplicação de exemplo, no emulador da empresa
CodeNameOne**



O deploy da aplicação é realizado no servidor da empresa, sendo que a licença para tal, só pode ser adquirida com a ajuda direta do desenvolvedor ao projeto, respeitando alguns critérios como ser commitar no projeto oficial, divulgação de conteúdos da plataforma em blogs, entre outros critérios disponibilizados na documentação oficial.

Alguns projetos estão disponíveis no portfolio de aplicativos (<http://www.codenameone.com/gallery.html>), que foram publicados nas lojas oficiais de cada um dos sistemas operacionais móveis. O acabamento visual das aplicações disponíveis nesta galeria não são tão bem definidos como aplicativos desenvolvidos com o SDK nativo, esta inferência baseia-se na inspiração de interfaces gráficas disponíveis em sites como <http://www.lovelyui.com/> e <http://inspired-ui.com/> que disponibilizam padrões de interfaces gráficas de aplicativos que já foram publicados e possuem um acabamento gráfico bem definido. Esta afirmação pode ser gerada a partir de duas variáveis, a primeira é a de que os componentes visuais da plataforma são efetivamente inferiores, a segunda é a de que os aplicativos foram desenvolvidos sem um designer especializado na aparência das aplicações.

Um construtor de interfaces gráficas também está disponível na plataforma, porém, os testes realizados demonstram que este construtor não funciona de maneira efetiva, o código

gerado é de difícil manutenção e compreensão, fazendo com que o desenvolvimento das aplicações torne-se um processo lento e complicado.

A documentação da aplicação disponibilizada em <http://www.codenameone.com/developer-guide.html> também não é um fator preponderante para a escolha do *framework*, pois ainda está em construção, alguns componentes ainda não foram documentados, dificultando o desenvolvimento das aplicação. Todos os códigos-fonte do *framework* e aplicações de testes presentes nos tutoriais oficiais estão em <http://code.google.com/p/codenameone>.

3.3 Tecnologias Selecionadas

Todas as ferramentas avaliadas possuem pontos positivos em diversos quesitos, porém, por existir esta diferença a escolha pelo melhor *framework* baseou-se na questão do desempenho da aplicação e facilidade na customização de componentes visuais nativos.

O bom desempenho das aplicações é essencial para o seu sucesso, visto que a interação do usuário com o sistema, principalmente nos ambientes móveis, é de primordial importância. A experiência como usuário na utilização de aplicativos que utilizam a metodologia web não foi eficiente, visto que a resposta da aplicação foi consideravelmente mais lenta em relação à aplicativos desenvolvidos com *frameworks* de interface gráfica nativa. Considerando estes aspectos, plataformas que fazem com que o binário da aplicação gere a interface gráfica nativa são as opções mais interessantes para desenvolvimento de aplicações híbridas.

Conclui-se com o comparativo dos frameworks de interfaces gráficas, que apesar da tecnologia HTML5 ser interessante, quando realizado o desenvolvimento de aplicativos reais para produção a rapidez com que a aplicação foi executada não foi eficiente.

Presume-se que os fatores que influenciam diretamente nesta lentidão são os eventos, como toques, que são tratados pelo próprio ambiente web, que não trata tão eficientemente a aplicação como uma aplicação nativa, em que todos os recursos são alocados diretamente do

hardware e toda a questão de eventos é tratada pelo próprio sistema operacional, que efetua um melhor gerenciamento de recursos para a aplicação (CAELUM, 2012).

O desempenho das aplicações *web mobile* pode ser melhorado pelo uso de metodologias como *web workers*, que são simuladores de Threads para o ambiente web, mas melhorar o desempenho da aplicação demanda um trabalho desnecessário, visto que além do desenvolvimento da aplicação é necessário retrabalhá-la para que o seu funcionamento seja aceitável. A análise aprofundada do melhoramento do desempenho e causas da lentidão da aplicação, podem ser retratadas em um trabalho futuro (MOZILLA, 2012).

Segundo a Caelum (2012), uma das mais respeitadas instituições de ensino voltada ao desenvolvimento de software no Brasil, relata que a discussão em relação aos dois tipos de aplicações são de extrema importância, pois, em lojas mais exigentes um aplicativo criado em *web mobile* pode não ser aceito, por motivos como lentidão e experiência de uso que não possui o visual da plataforma, como por exemplo, uma aplicação com visual de IOS sendo executada no Android.

Outro caso interessante que ocorreu em paralelo ao desenvolvimento deste trabalho foi a declaração de Mark Zuckerberg, CEO do Facebook, dizendo que foi um erro confiar na tecnologia HTML5 para dispositivos móveis, pois a lentidão da aplicação em ambientes web móveis fizeram com que a empresa tivesse que refazer o seu software para plataformas nativas. Mick Johnson, gerente de produtos do Facebook, salientou ainda, que a reconstrução da aplicação era iminente, e que a aplicação em HTML5 não era o que os usuários esperavam, sendo que a nova aplicação é duas vezes mais rápida em todos os aspectos. Segundo ele, o software refeito para IOS não possui diferença em sua aparência, mas deve satisfazer as centenas de milhões de usuários, que imploram por uma experiência que não seja extremamente lenta.

Dentre as ferramentas híbridas com interfaces gráficas nativas, todos os *frameworks* possuem um desempenho parecido, já que o resultado final das aplicações serão praticamente os mesmos. O *framework* Titanium possui os melhores componentes visuais, sendo que o bom acabamento aliado ao excelente desempenho, demonstra ser a ferramenta que melhor condiz com os objetivos deste estudo.

A quantidade de plataformas suportadas não é um fator tão relevante, pois as funcionalidades variam de plataforma para plataforma, sendo que a quantidade de recursos são limitadas entre elas. Por exemplo, componentes presentes em IOS podem não estar disponíveis para o WP7, gerando uma necessidade de tratar as particularidades de maneira separada.

A análise dos *frameworks* foram embasadas em quesitos qualitativos, demonstrados através das funcionalidades dispostas em cada um deles, dada a diferença entre elas, a disposição de todos os componentes presentes em cada uma das plataformas analisadas será resumida na tabela demonstrada a seguir.

Tabela 1: tabela comparativa dos componentes dispostos em cada um dos frameworks.

	Documentação	Emulador	Curva de Aprendizado	Plataformas Suportadas	Linguagens de Programação
Titanium	Bem acabada e demonstra a maioria dos componentes através de exemplos	do SDK de cada uma das plataformas	JavaScript intermediário	Android IOS BlackBerry	Javascript
CodeNameOne	Em andamento, partes da documentação está incompleta.	Emulador da própria empresa	Java Avançado.	Android, IOS, WP7, BlackBerry, J2ME	Java
MoSync	Não demonstra os componentes através de exemplos, um programador inexperiente poderá se confundir em meio à API.	Emuladores do SDK e ferramenta Reloaded que permite testes no aparelho real	C++ Avançado.	Android, IOS, WP7, BlackBerry, J2ME	Javascript, C, C++

No contexto de aplicação de leitura de mapas, a escolha foi pelo *framework* Leaflet, ambos apresentaram ótimas funcionalidades, um diferencial decisivo em relação à biblioteca Leaflet, é sua arquitetura modular, possibilitando a redução do código utilizado, quesito importantíssimo no contexto de aplicações móveis. Outro ponto importante é o foco e suporte que a empresa dá ao desenvolvimento deste tipo de aplicação.

Baseado nos conceitos apresentados, a tabela 2 demonstra todas as ferramentas selecionadas:

Tabela 2: *Frameworks* e Softwares que serão utilizados para o desenvolvimento do trabalho e suas respectivas versões.

Módulo	Software Selecionado	Versão
Interface Gráfica	Titanium	2.1.2
Mapas	Leaflet	0.3.1

As ferramentas selecionadas neste capítulo serviram como base para tomada de decisão em relação a qual software utilizar para desenvolver aplicativos móveis. No capítulo a seguir será desenvolvido o estudo de caso, a aplicação DutosMobile.

4 ESTUDO DE CASO: DESENVOLVIMENTO DA APLICAÇÃO DUTOSMOBILE

O desenvolvimento das aplicações ocorre em parceria com a Fundação de Ciências Aplicações e Tecnologias Espaciais, sendo que todos os requisitos das aplicações demonstradas neste capítulo são de necessidades reais da empresa.

Além do aplicativo proposto neste trabalho, outros aplicativos foram desenvolvidos para a empresa, neste desenvolvimento ocorreram diversas reuniões para decisão das melhores tecnologias a serem utilizadas, sendo uma das principais motivações para este trabalho.

A pesquisa permitiu com que diversos *frameworks* fossem analisados, além de expôr fatores como a mudança e reestruturação da aplicação para manter o objetivo dos requisitos, visto que por alguns momentos o comportamento do HTML5 não supriu as necessidades das aplicações propostas no desenvolvimento dos aplicativos da empresa.

As decisões inferidas no desenvolvimento das aplicações da Funcate, permitiram o direcionamento do desenvolvimento da aplicação DutosMobile, utilizando os *frameworks* selecionados e analisados nos capítulos anteriores. As seções à seguir descrevem a arquitetura e desenvolvimento da aplicação DutosMobile.

Na seção 4.1 serão descritos a motivação do desenvolvimento da aplicação, a seção 4.1.1 descreve os requisitos e arquitetura do sistema, seguidos da seção 4.1.2 que mostra a aplicação funcionando nas plataformas Android e IOS.

4.1 Dutos Mobile

Dutos são estruturas geralmente metálicas e ocas nas quais transportam-se produtos de uma região para outra. O transporte de produtos por dutos faz-se necessário para grandes distâncias. Sua principal finalidade é a de proporcionar segurança, rapidez e economia em seu transporte. Geralmente os componentes transportados são gases naturais, petróleo e serviços de cabeamento como internet e energia (CEPA, 2012).

Este meio de transporte habitualmente se faz por canos subterrâneos, terrestres ou submarinos. No caso dos dutos subterrâneos a periculosidade e o cuidado com os locais por onde passa são essenciais, pois, grandes acidentes podem ser ocasionados por perfurações não supervisionadas, ocasionando por vezes, mortes ou custos inestimáveis para o serviço prestado (CERQUEIRA, FRANCISCO, 2012).

Nas grandes cidades o uso de dutos subterrâneos é substancialmente necessário, visto que no caso de cabeamentos externos, a exposição a condições climáticas interferem na qualidade dos serviços prestados, já que os cabos são danificados. Dutos subterrâneos fazem com que os cabos fiquem protegidos além de proporcionar a despoluição visual.

Dado a importância dos dutos subterrâneos é necessário que se utilize sistemas de geoprocessamento para manter o controle sobre os locais por onde passam estes componentes. A proposta da aplicação DutosMobile é de permitir a consulta dos dutos e contatos com seus respectivos responsáveis técnicos, proporcionando mobilidade ao usuário final, pois as consultas poderão ser realizadas em campo.

4.1.1 Requisitos e Arquitetura

Serão descritos neste documento, os requisitos necessários para o desenvolvimento da aplicação DutosMobile, este sistema deverá fornecer interfaces visuais para os usuários, que permitam guiá-lo a executar qualquer intervenção nos gasodutos, mostrando as informações básicas sobre o local, além de fornecer o contato com os respectivos responsáveis.

A saída visual será um mapa no qual os objetos geográficos aparecem destacados pela cor específica ao grupo do qual é integrante, ou seja, dependendo da empresa da qual ele esteja associado.

Sua descrição será mostrada a partir de um balão, que será acionado à partir do toque do usuário no duto selecionado, que conterà macro informações sobre o local, tais como:

- Nome da Empresa;
- Nome do Responsável técnico;

- Botão para efetuar a ligação aos técnicos;
- Botão para redirecionar o usuário para a tela de detalhamento das informações do Duto;

Além da camada de visualização dos dutos, o mapa deverá possuir uma camada do OpenStreetMaps mostrando de maneira clara o local no qual o duto localiza-se.

O Programa foi implementado na plataforma Titanium, sendo que todos os componentes visuais, com exceção da tela dos mapas, serão nativos, dado que a resposta da aplicação será mais efetiva, como demonstrado no capítulo 2.

De acordo com a pesquisa, inexitem ferramentas nativas no titanium que auxiliem o desenvolvimento de aplicativos de geoprocessamento, desta forma, o módulo de mapas será implementado com o framework Leaflet, através de um webview.

4.1.2 Funcionamento da Aplicação em diferentes plataformas

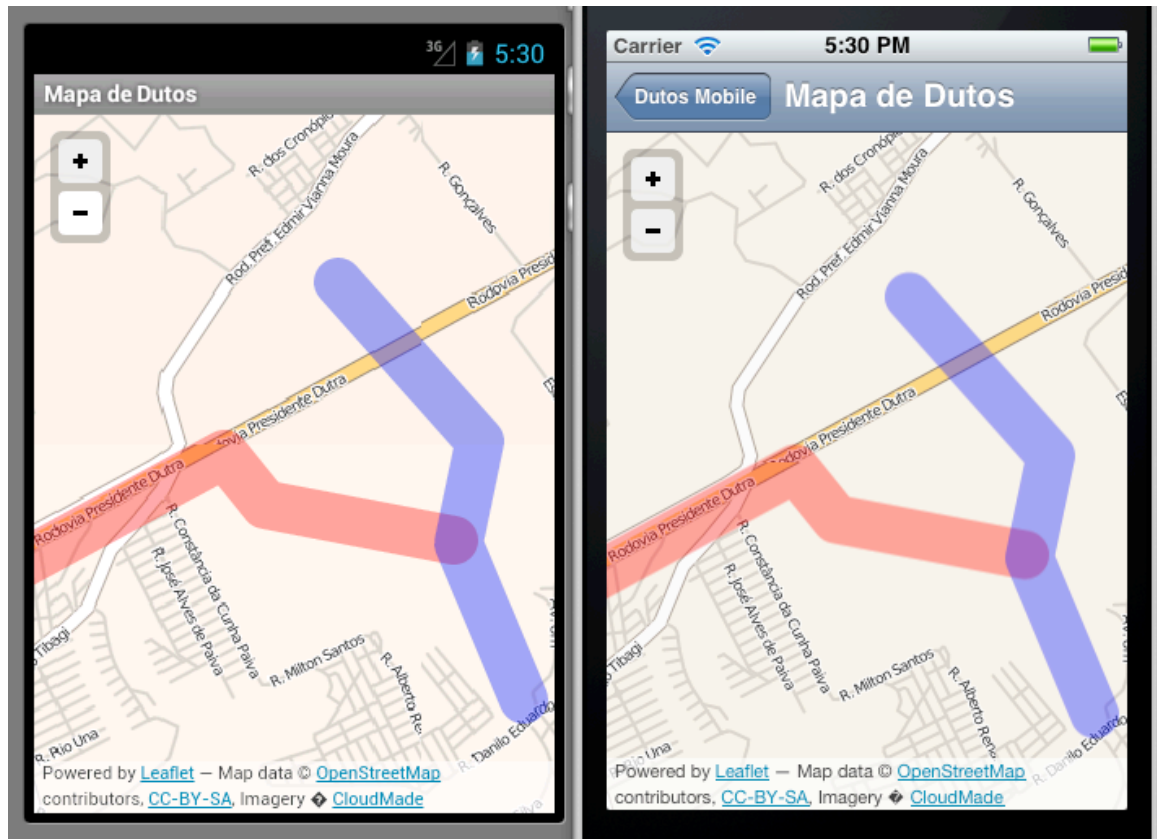
As figuras a seguir demonstram as telas do sistema, executadas em duas plataformas distintas, sendo elas Android e IOS, sendo que em todas as figuras dos emuladores, à esquerda é o software sendo executado no Android e a direita no IOS.

Na primeira tela exibesse o logo da empresa Funcate, seguido do logo da Faculdade de Tecnologia de São José dos Campos (FATEC) e em seguida um botão para redirecionar o usuário para a tela de visualização do mapa de Gasodutos.

FIGURA 21: TELA inicial sendo executada nos emuladores.

A segunda tela exibe dois mapas ao usuário, sendo que um sobrepõe-se ao outro. A primeira camada é o mapa de gasodutos representado por dois Gasodutos de diferentes empresas / cores. A camada que se antepõe ao mapa de gasodutos é do OpenStreetMap para melhorar a identificação do local onde o gasoduto encontra-se.

FIGURA 22 - Gasodutos fictícios, localizados próximos da Fatec SJC.



O toque do usuário no gasoduto fará com que seja exibido um balão com informações referentes ao duto, como o nome da empresa e nome do responsável, seguidos da tecla de ligação que permite que uma chamada telefônica seja efetuada ao responsável técnico, além de um link para ver o detalhamento deste local, como é exibido nas figuras 23 e 24:

FIGURA 23: Informações do gasoduto vermelho.

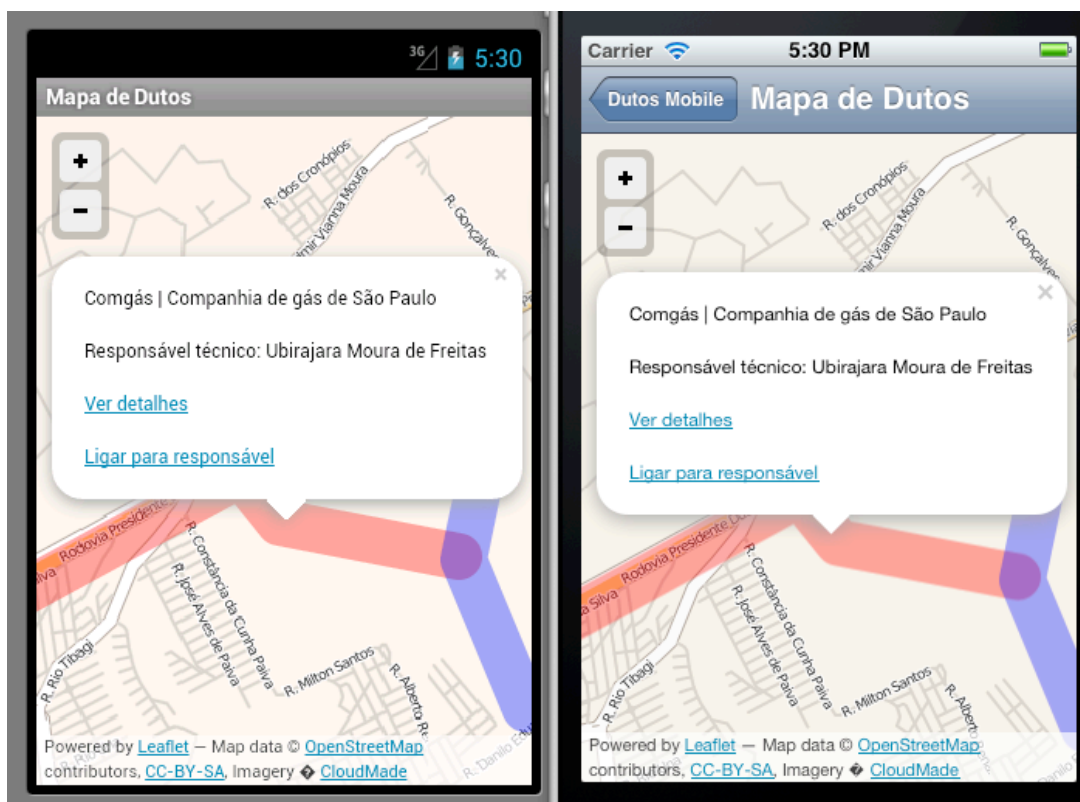
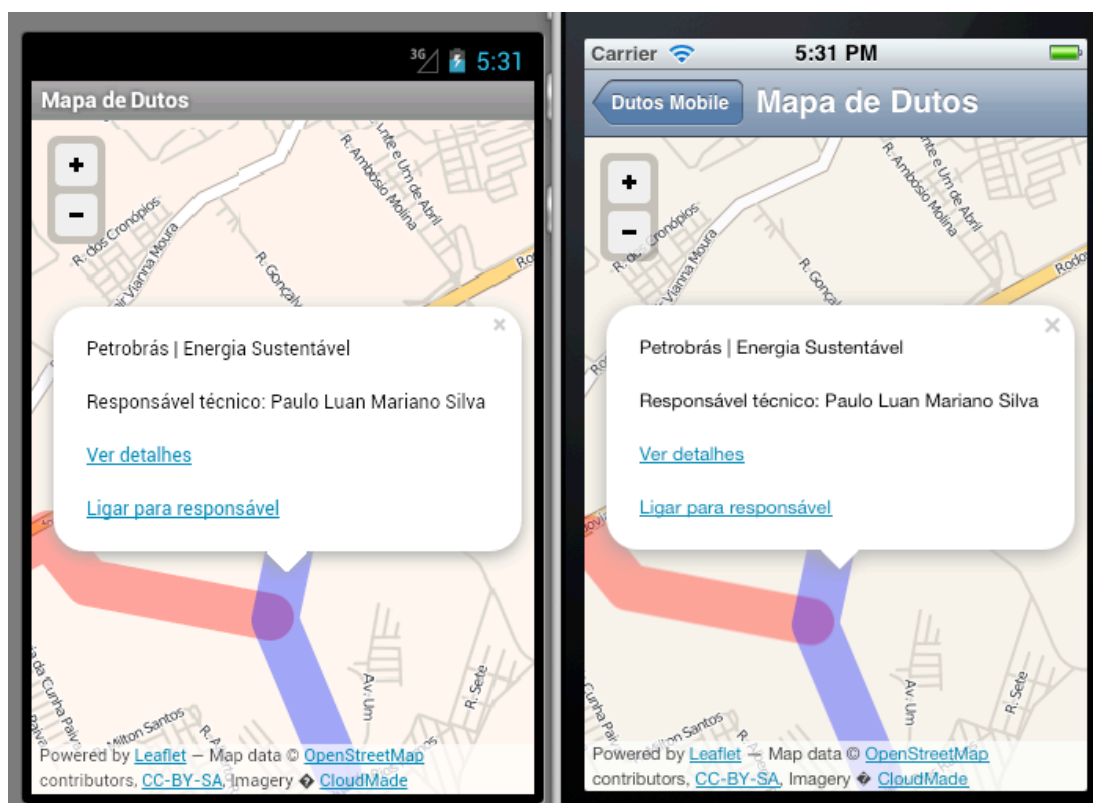
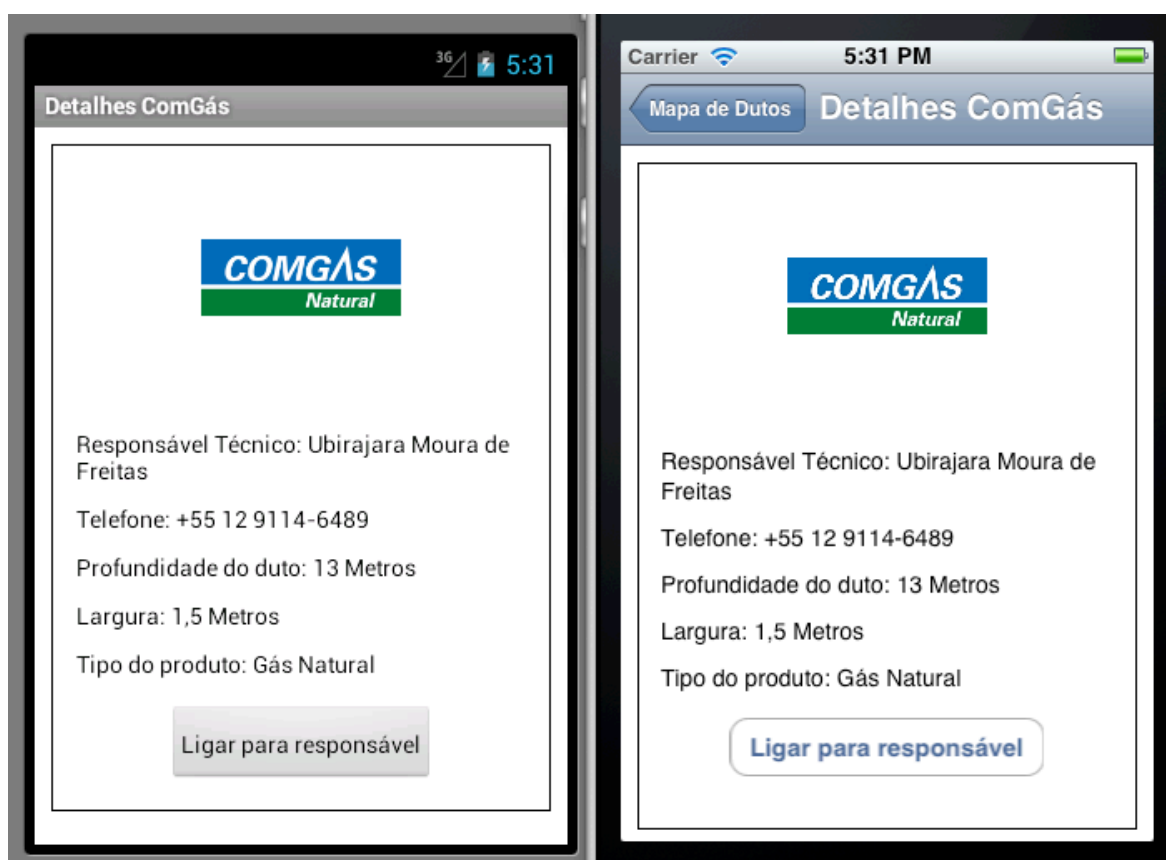


FIGURA 24: balão de informações do gasoduto azul.

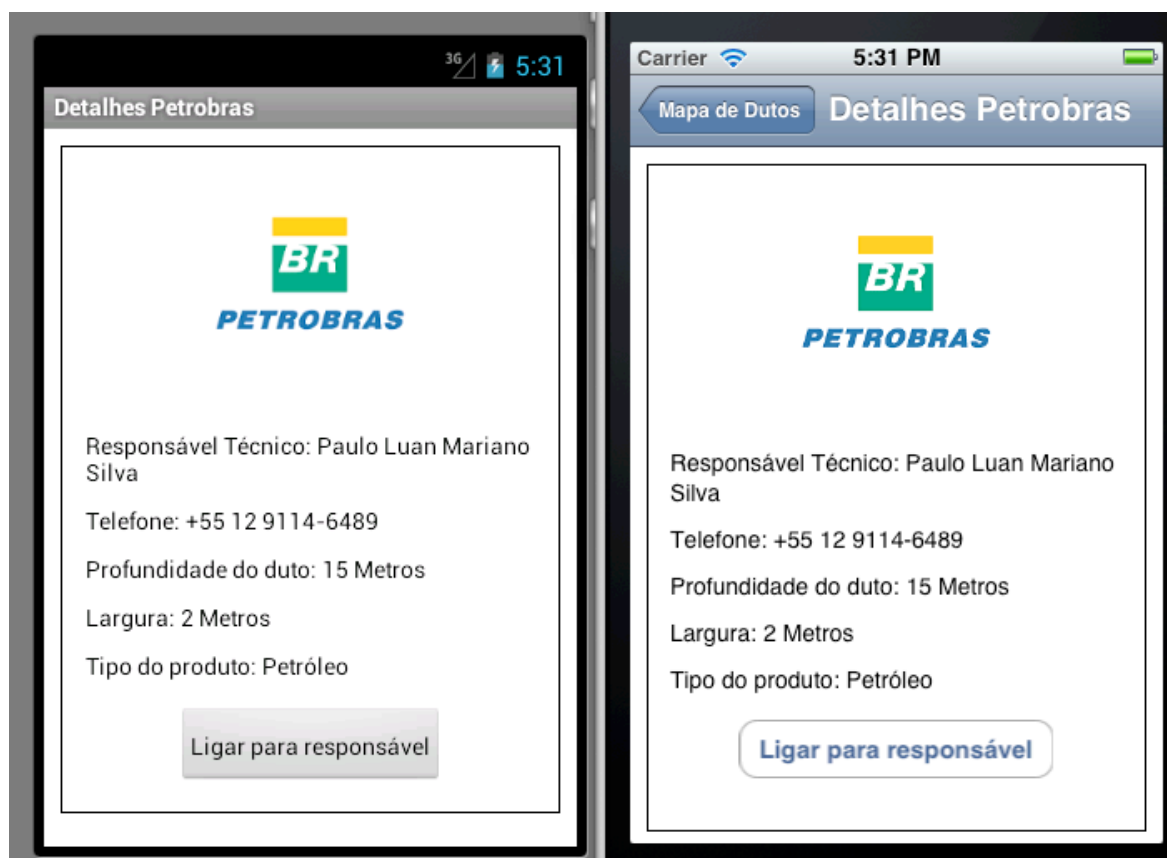


Por fim, a tela de detalhes exibe dados como profundidade, grossura e tipo de componente que é transporado pelo duto, permitindo que qualquer intervenção no local seja segura, além de mostrar novamente a opção de realização da ligação ao responsável técnico, para que detalhes ou dúvidas específicas sejam sanadas.

FIGURA 25: Detalhes do Duto vermelho (ComGas)



**FIGURA 26: Detalhes do duto azul
(Petrobrás)**



5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um comparativo entre tecnologias móveis híbridas emergentes no mercado de desenvolvimento de software mobile, além do desenvolvimento de um protótipo de aplicação com a plataforma Titanium, comprovando sua eficiência.

O desenvolvimento da aplicação DutosMobile permitiu a análise do comportamento de projetos web dentro de um projeto de aplicação nativa, sendo que toda a questão de eventos é tratada pelo sistema operacional. Já o módulo de geoprocessamento foi desenvolvido com ambiente e bibliotecas web, sendo exibidos através de WebViews.

Um dos pontos a se levar em consideração na adoção de qualquer um dos modelos de desenvolvimento híbrido, é que o desenvolvedor fica a mercê das empresas que mantêm o *framework*. As diretrizes do desenvolvimento dos sistemas operacionais móveis podem ser modificadas, e estas restrições podem impactar diretamente sobre o funcionamento das aplicações desenvolvidas em ferramentas de terceiros.

5.1 Contribuições

As contribuições deste trabalho são:

- Apresentação de alternativas ao desenvolvimento de aplicativos nativos com as ferramentas disponibilizadas pelos próprios fabricantes dos sistemas operacionais;
- Comparativo das principais tecnologias híbridas disponível no mercado de desenvolvimento de aplicativos móveis;
- Implementação de um aplicativo com as ferramentas selecionadas no comparativo dos frameworks;

A partir destas contribuições pode-se concluir que:

- É possível desenvolver de maneira híbrida, fazendo com que o aplicativo funcione de maneira efetiva em multi-plataformas;

- O nível de amadurecimento da ferramenta Titanium viabiliza o desenvolvimento de aplicativos para produção.
- HTML5 em ambientes móveis ainda não está em um nível de maturidade suficiente para aplicativos para produção, mas a tecnologia está em constante evolução e futuramente a tendência é que esta seja uma das promissoras ferramentas de desenvolvimento de aplicativos móveis.

A seguir são apresentadas experiências obtidas ao longo do desenvolvimento deste trabalho:

- Aprendizado da linguagem de programação Javascript;
- Abertura do conhecimento para novos paradigmas de programação;
- Crescimento como desenvolvedor / programador;
- Crescimento de habilidades como analista de sistemas;

5.1.1 Publicação

O seguinte artigo foi publicado em periódico como resultado desse trabalho: SILVA, P. L. M.; FREITAS, U. M.; BERTOTI, G. A. Avaliação de Tecnologias Híbridas para o desenvolvimento de aplicativos móveis. Boletim Técnico da Faculdade de Tecnologia de São Paulo. 14º Simpósio de Iniciação Científica e Tecnológica, ISSN: 1518-9082, 2012.

5.2 Trabalhos futuros

As contribuições alcançadas com este trabalho não encerram as pesquisas relacionadas ao comparativo de ferramentas híbridas, mas abrem oportunidades para alguns trabalhos futuros. O desenvolvimento deste trabalho, possibilitou a análise de algumas ferramentas de desenvolvimento, mas além destes *frameworks*, existem diversos outros com funcionalidades semelhantes, porém, não foram abordados neste trabalho. A análise aprofundada de tais *frameworks* possibilitariam um embate entre os *frameworks* analisados, e pode ser realizada em um trabalho futuro. A tabela completa das demais ferramentas está presente no Anexo A.

Alguns fatores que levaram a não análise de alguns *frameworks*, foram que muitos deles possuem licença comercial, fator este que por vezes impede o uso e análise aprofundada de

sua arquitetura. Além disto, muitos deles foram descobertos ao fim da pesquisa, causando um ciclo eterno de análise, o que impediria a finalização do estudo e da aplicação.

Extensões desta pesquisa podem ser realizadas tais como:

- Estudo de melhorias do desempenho de aplicações web mobile;
- Estudo arquitetural dos *frameworks* apresentados;
- Análise de outros *frameworks* híbridos;
- Análise dos fatores que influenciam no desempenho de uma aplicação HTML5 mobile.
- Adição de novas funcionalidades do DutosMobile;
- Armazenamento de mapas offline na app;
- Visualização dos Dutos a partir de um servidor, gerando as imagens dinamicamente.
- Permitir o usuário editar a localização dos Dutos;

REFERÊNCIAS

(APPCCELERATOR, IDC, 2011) Apple shines, google slows, and microsoft leapfrogs RIM in battle for mobile developer mindshare. <http://www.appcelerator.com.s3.amazonaws.com/blog/www/Appcelerator-IDC-Q2-2011-Mobile_Developer-Report.pdf>

(APPCCELERATOR, IDC, 2011) Google+ and apple iCloud bring mobile platform battle to the cloud. <http://www.appcelerator.com.s3.amazonaws.com/blog/www/Appcelerator-IDC-Q3-2011-Mobile_Developer-Report.pdf>

(APPCCELERATOR, IDC, 2011) Tablet onslaught reprioritizes development efforts as businesses accelerate their mobile app strategies. <http://www.appcelerator.com.s3.amazonaws.com/blog/www/Appcelerator-IDC-Q1-2011-Mobile_Developer-Report.pdf>

(APPCCELERATOR, IDC, 2012) Surveys finds apple leading the charge in the enterprise; Developers see android a consumer opportunity and express interest in windows 8, cloud continues momentum. <<http://www.appcelerator.com.s3.amazonaws.com/blog/www/Appcelerator-Report-Q2-2012.pdf>>

(APPCCELERATOR, IDC, 2012) The Google and Facebook battle moves to mobile while HTML5 gains speed <http://www.appcelerator.com.s3.amazonaws.com/blog/www/Appcelerator-IDC-Q1-2012-Mobile_Developer-Report.pdf>

(BALMER, DAVE, 2012) Jo, Framework Javascript para HTML5. Disponível em <<http://joapp.com/docs/#About>> Acessado em 12/04/2012.

(BURROUGH, 1998) BURROUGH, P.A.; McDONNELL R.A. Principles of Geographical Information Systems. 1 ed. Universidade de Oxford, 1998. ISBN 0-19-823366-3. Disponível em < http://www.rc.unesp.br/igce/geologia/GAA01048/papers/Burrough_McDonnell-Two.pdf> Acessado em 02/04/2012.

(CAELUM, 2012) 2012 é o ano do mercado mobile no Brasil. Sérgio Lopes, 2012. Disponível em <<http://blog.caelum.com.br/2012-e-o-ano-do-mercado-mobile-no-brasil/>>

(CAELUM, 2012) Aplicações mobile, web ou nativas? André Silva, 2012 Disponível em <<http://blog.caelum.com.br/aplicacoes-mobile-web-ou-nativa/>>

(CAELUM, 2012) Não use JQuery Mobile no seu site web mobile, conheça o ZeptoJS. Sérgio Lopes, 2012 <<http://blog.caelum.com.br/nao-use-jquery-no-seu-site-mobile-conheca-o-zepto-js/>>

(CÂMARA, 1996) CÂMARA, G.; CASANOVA, M. A.; HEMERLY, A. S.; MAGALHÃES, G. C.; MEDEIROS, C. M. B. Anatomia de Sistemas de Informação Geográfica. Unicamp – Universidade de Campinas, Instituto de Computação 1996.

(CÂMARA, 2005) CÂMARA, G.; CASANOVA, M.; DAVIS, C.; VINHAS L.; QUEIROZ G. R. Banco de Dados Geográficos. 2005. Disponível em: <<http://www.dpi.inpe.br/livros/bdados>>. Acesso em: 10/03/2011.

(CEPA, 2012) Tranposte por dutos, Centro de Ensino e Pesquisa Aplicada, Disponível em <<http://www.cepa.if.usp.br/energia/energia1999/Grupo4A/dutos.htm>> Acessado em 08/09/2012.

(CERQUEIRA, FRANCISCO, 2012) Transporte por dutos. Wagner De Cerqueira e Francisco, Disponível em <<http://www.mundoeducacao.com.br/geografia/transporte-por-dutos.htm>> Acessado em 03/04/2012

(CHEN, KOTS, 2000) Guanling Chen and David Kotz Department of Computer Science Dartmouth College A Survey of Context-Aware Mobile Computing Research, 2000. Disponível em <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.3131&rep=rep1&type=pdf>>. Acessado em 25/03/2012

(CODENAMEONE, 2012) The Best Cross Platform Java Tool For Mobile App Developers, 2012, disponível em <www.codenameone.com> acessado em 10/07/2012

(CODIGA, 2012) Rapid Prototyping with JQuery Mobile, 2012. Disponível em <<http://www.codiga.com/>> Acessado em 07/07/2012.

(EGENIAL, 2012) Afinal o que é Titanium Mobile e qual é diferença entre ele, jQuery Mobile e Sencha Touch? EGenial, 2012. Disponível em <<http://blog.egenial.com.br/2012/02/12/>> Acessado em 10/10/2012.

(EGENIAL, 2012) Titanium Mobile: Por que investir no mercado de aplicativos para dispositivos móveis? Disponível em <<http://blog.egenial.com.br/2011/09/04/curso-de-titanium-mobile-por-que-investir-no-mercado-de-aplicativos-para-dispositivos-moveis/>> Acessado em 10/10/2012.

(EXTRA, 2012) Infográfico: Uso do celular no Brasil. 2012. Acessado em 11/05/2012. Disponível em <<http://imagens.extra.com.br/html/infografico/celular-no-brasil.html>>

(FACEBOOK, 2012) Under the hood: Rebuilding Facebook for iOS, Jonathan Dann. 2012. Disponível em <<https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920>>

(FANGXIONG, 2004) Fangxiong, W.; Fuling, B.; Yingzi, H. A Distributed architecture for WAP based mobile GIS. INTERNATIONAL CONFERENCE ON GEOINFORMATICS – GEOSPATIAL INFORMATION RESEARCH: BRIDING THE PACIFIC AND ATLANTIC, Universidade de Wuhan, 2004. Disponível em <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.548&rep=rep1&type=pdf>> Acessado em: 09/06/2012.

(FAYAD, SCHMIDT, 1997) FAYAD, M. E., SCHMIDT, D. C. Object-oriented Application frameworks. Communications of the ACM, Vol. 40, 10 p., 1997.

(FORREST, 2009) PhoneGap, The Mobile Platform democratizer, O'Reilly Media, Brady Forrest, 2009. Disponível em <<http://radar.oreilly.com/2009/04/phonegap-mobile-platform-democratizer.html>> Acessado em 05/07/2012.

(GARDNER, GRIGSBY 2011). Head First Mobile Web. O'Reilly Media. Dezembro 2011 ISBN-10: 1449302661.

(GEOSERVER, 2012) GEOSERVER Disponível em <<http://geoserver.org/display/GEOS/Welcome>>. Acessado em 09/04/2012.

(HARREL, 2011) William Harrel, HTML, CSS e Javascript mobile development for dummies, 2011 ISBN 978-1-118-02622-9, Editora Wiley.

(HARTMANN, STEAD, DEGANI, 2011) Cross-platform mobile development, Gustavo Hartmann, Geoff Stead, Asi DeGani 2011, Mobile learning enviroment of Naval Research Global (ONRG). Disponível em: <http://www.it.hiof.no/mobapp/misc/WP4_crossplatform_mobile_development_March2011-1.pdf> Acessado em 10/10/2012

(IDC, 2012) International Data Corporation, 2012. Disponível em: <<http://www.idc.com/>> Acessado em 13/04/2012,

(JO, 2012) JO Framework Web, 2012. Disponível em <<http://joapp.com/>> Acessado em (12/04/2012).

(JQTOUCH) A Zepto/jQuery plugin for mobile web mobile development. Acessado em 15/06/2012. Disponível em <<http://jqtouch.com/>>

(JQUERY, 2009) JQUERY. 2009. Disponível em: < <http://jquery.com/>>. Acessado em: 08/04/2012

(JQUERYMOBILE, 2010) JQUERYMOBILE. 2010. Disponível em: < <http://jquerymobile.com/>>. Acessado em: 08/04/2012

(KLEIN, 2012) How to decide: Mobile websites vs. mobile apps, Dave Klein 2012. Disponível em <<http://www.adobe.com/inspire/2012/02/mobile-websites-vs-mobile-apps.html>> acessado em 07/09/2012.

(LEAFLET, 2012) Leaflet, a modern lightweight open-source javascript library for interactive maps, 2012. Disponível em <<http://leaflet.cloudmade.com/>> Acessado em 07/06/2012

(MAPSERVER, 2012) Disponível em <<http://mapserver.org/>> Acessado em 12/04/2012.

(MICHELE FILANNINO, 2010) PhoneGap recebe il nulla osta di Apple per App Store, Michele Filannino, 2010. Disponível em <<http://www.melablog.it/post/11421/phonegap-riceve-il-nulla-osta-di-apple-per-app-store>> Acessado em 23/05/2012.

(MOSYNC, 2012) MoSync cross-platform mobile application development, 2012. Disponível em <<http://www.mosync.com/>> Acessado em 12/09/2012.

(MOZILLA, 2012) Using Web Workers, 2012. Disponível em <https://developer.mozilla.org/en-US/docs/DOM/Using_web_workers> Acessado em 17/07/2012.

(NITOBI, 2012) Nitobi – Web and Mobile Consulting <<http://www.nitobi.com/>> Acessado em 01/04/2012

(NODEJS, 2012) NodeJS server-side javascript, 2012. Disponível em <<http://nodejs.org/>> Acessado em 07/07/2012.

(OGC, 2011) Open GeoSpatial Consortium, 2011. Disponível em: <<http://www.opengeospatial.org/>>. Acessado em: 09/04/2012.

(OPEN SOURCE GEOSPATIAL FOUNDATION, 2006) THE OPEN SOURCE GEOSPATIAL FOUNDATION, 2006. Disponível em <<http://www.osgeo.org/>>. Acessado em 09/04/2012.

(OPENLAYERS, 2006) OPENLAYERS, 2006. Disponível em: <<http://openlayers.org/>>. Acessado em 09/04/2012.

(PALMIEIRI, SINGH, 2011) Comparison of Cross-Platform Mobile Development Tools, Manuel Palmieri, Inderjeet Singh. Malardalen University Innovation, Development and Technology, 2011. Disponível em: <http://www.idt.mdh.se/kurser/ct3340/ht11/MINICONFERENCE/FinalPapers/ircse11_submission_16.pdf> Acessado em 24/05/2012

(PHONEGAP, 2011) PhoneGap – The only open-source mobile framework that supports 7 Platforms. Disponível em <<http://phonegap.com/>> Acessado em 02/04/2012.

(REID, 2011) JONATHAN REID; JQuery Mobile: Building Cross-Platform Mobile Applications. O'Reilly Media, 2011. ISBN: 978-1-449-30668-7.

(SCHILIT, ET AL, 1994) Bill Schilit, Norman Adams, Roy Want. CONTEXT AWARE COMPUTING APPLICATIONS, 1994 Disponível em <<http://graphics.cs.columbia.edu/courses/mobwear/resources/schilit-mcsa94.pdf>> Acessado em 05/04/2012

(SENCHA TOUCH, 2012) SENCHA TOUCH criação de aplicativos com HTML5. Disponível em <<http://www.sencha.com/products/touch>> Acessado em 12/04/2012.

(SILVA, ET AL, 2008) WAISTER SILVA MARTINS, BRUNO RABELLO MONTEIRO, JUGURTA LISBOA FILHO, MAURO NACIF ROCHA, Problemas e Soluções no Desenvolvimento de um SIG Móvel para Aplicações Urbana. Informática Pública ano 10 (1): Páginas 69-80, 2008. Disponível em: <http://www.ip.pbh.gov.br/ANO10_N1_PDF/problemas_solucoes_desenvolvimento_sig.pdf>, Acessado em 03/04/2011

(SMITH, 2009) SMITH, T. M.; V. LAKSHMANAN, 2006: Utilizing Google Earth as a GIS platform for weather applications. Preprints, 22d Int. Conf. on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, Atlanta, GA,

Amer. Meteor. Soc. Disponível em <<https://ams.confex.com/ams/pdfpapers/104847.pdf>>
Acessado em: 09/06/2012.

(STEWART, 2011) PhoneGap Basics: What it is and what it can do for mobile developers. O'Reilly Media, Bruce Stewart, 2011. Disponível em <<http://radar.oreilly.com/2011/10/phonegap-mobile-development.html>> Acessado em 05/07/2012.

(TITANIUM AWARDS, 2012) Titanium Awards, 2012 Disponível em <<http://www.appcelerator.com/company/awards/>> Acessado em 07/09/2012.

(TITANIUM, 2011) Titanium Mobile SDK, 2012. Disponível em <<http://www.appcelerator.com/products/titanium-mobile-application-development/>> Acessado em 14/04/2012.

(W3C, 2012) W3C,Standards, Disponível em: <<http://www.w3.org>>, Acessado em 06/04/2012

(W3SCHOOLS, 2012) Disponível em <<http://www.w3schools.com/js/default.asp>> Acessado em 10/04/2012.

(WANT, ET AL, 1992) Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The Active Badge location system. ACM Transactions on Information Systems, 1992. Disponível em <<http://luci.ics.uci.edu/predeployment/websiteContent/weAreLuci/biographies/faculty/djp3/LocalCopy/p91-want.pdf>> Acessado em 04/04/2012.

(WEB 2.0 EXPO, 2009) Web 2.0 Expo Disponível em <<http://www.web2expo.com>> Acessado em 12/04/2012

(WEBKIT, 2012) The WebKit Open-Source Project, 2012. Disponível em <<http://www.webkit.org/>> Acessado em 05/09/2012

(WHINNERY, 2012) Kevin Whinnery: Comparing Titanium and PhoneGap, 2012. Disponível em <<http://kevinwhinnery.com/post/22764624253/comparing-titanium-and-phonegap>> Acessado em 05/09/2012.

(ZANELATO, 2010). ZANELATO, A. WEB SERVICES: UMA NOVA ARQUITETURA DA INTERNET, ETIC – Encontro de Iniciação Científica, v.6, n.6, 2010. ISSN 21-76-8498. Disponível em: <<http://intertemas.unitoledo.br/revista/index.php/ETIC/article/view/2541/2065>>. Acesso em: 09/04/2012.

Anexo A:

Tabela 3 : Tabela de frameworks híbridos.

Ferramenta	Site	Licença
Rhodes	http://rhomobile.com/products/rhodes/	Código-Aberto
Phonegap	http://www.phonegap.com	Código-Aberto
FeedHenry	http://developer.feedhenry.com/	Comercial
Appcelerator	http://www.appcelerator.com/	Código-Aberto
Grapple	http://www.grapplemobile.com/	Comercial
Kivy	http://kivy.org/	Código-Aberto
MotherApp	http://www.motherapp.com/	Comercial
Corona	http://www.anscamobile.com/corona/	Comercial
Sencha Touch	http://www.sencha.com/products/touch/	OS/Comercial
MoSync	http://www.mosync.com/	Código-Aberto
Resco	http://www.resco.net/	Comercial
CouchOne	http://www.couchone.com/products	Comercial
MobileIron	http://mobileiron.com/	Comercial
WidgetPad	http://widgetpad.com	Código-Aberto
AML	http://www.amlcode.com	Código-Aberto
Jo	http://joapp.com	Código-Aberto
xui	http://xuijs.com	Código-Aberto
JQuery	http://jquerymobile.com	Código-Aberto
JQTouch	http://jqtouch.com	Código-Aberto

QT	http://qt.nokia.com/products/qt-formobile-platforms/	Código-Aberto
QuickConnectFamily	http://www.quickconnectfamily.org/	Código-Aberto
Bedrock	http://www.metismo.com	Comercial
WebApp.net	http://webapp-net.com/	Código-Aberto

FONTE: Adaptado de (HARTMANN, STEAD, DEGANI, 2011)