

DESENVOLVENDO PARA IOS E ANDROID CONHECENDO SEUS SDK'S

ALEX NUNES DA SILVA

Graduando do 8º período do curso de Sistemas de Informação
da Faculdade Joaquim Nabuco – Recife
alexnunes008@yahoo.com.br

RESUMO

Este artigo propõe apresentar os dois principais sistemas operacionais para dispositivos móveis na atualidade, o Android, sistema de código fonte aberto da Google e o IOS o sistema proprietário da Apple. Busca orientar os desenvolvedores, estudantes e curiosos de aplicativos para dispositivos móveis baseados nessas plataformas, fornecendo seus princípios básicos e apresentando seus sdk's e respectivos kits de ferramenta para desenvolvedores, demonstrando o desenvolvimento para IOS utilizando sua linguagem nativa, o Objective-c com o seu respectivo kit de desenvolvimento da Apple (*xcode*, *interface builder*, *cocoa touch* e o simulador), através da comparação com o *Android sdk*, kit de desenvolvimento que disponibiliza as ferramentas e *API's (Application Programming Interface)* necessárias para desenvolver aplicações para a plataforma Android, utilizando a linguagem popular Java, Adt um plugin do android para eclipse a principal IDE (*Integrated Development Environment*) para desenvolvimento em Java, tendo como objetivo mostrar como pode ser fácil programar para essas duas plataformas.

PALAVRAS-CHAVE: Xcode. IOS. Android

ABSTRACT

This article proposes to show two most important operational systems to mobile devices nowadays. Android, system with open source code from Google and the IOS, the owner system from Apple. These systems, based on SDK's and tool kits, try to guide developers, students and curious who want to get know better about applications for these devices. They have their own language, the Objective-c with your tool kit (xcode, interface builder, cocoa touch and the simulator) to developers, from Apple. By the comparison with Android sdk, its tool kit provides the tools and APIs needed to develop applications for a platform Android using the popular language, Java. ADT a Android's plugin for Eclipse (the main tool IDE to develop on Java) has as objective show how easy can it be programming to these two platforms.

WORD KEY: Xcode, IOS, Android

1. INTRODUÇÃO

Nos dias atuais existe um grande crescimento de vendas dos *Smartphones* e *tablets*, onde cada vez mais surgem novas plataformas e ferramentas de desenvolvimento. Neste artigo pretende-se explorar as duas principais plataformas de dispositivos móveis acima citadas, mostrando as principais vantagens dos Sistemas operacionais IOS e Android, demonstrando aos desenvolvedores, estudantes e curiosos o desenvolvimento básico (passo a passo) para os SO's.

Pretende-se ainda, demonstrar as facilidades que o IOS SDK da Apple (*Software Developer Kit*) possui: desde o ambiente para programação até o simulador, apresentando o que é preciso para desenvolver para esta plataforma que são realizados através da sua linguagem nativa - o *Objective-c* e um conjunto que tem como suas ferramentas o *xcode*, *interface builder*, *cocoa touch* e seu simulador que é disponibilizadas em um kit único de desenvolvimento da Apple.

2. Desenvolvimento e demonstração dos resultados

A tendência natural de muitos desenvolvedores é acompanhar as atualizações do mercado, criar produtos e soluções que tirem o máximo de proveito das vantagens oferecidas pela tecnologia do momento, atualmente é a de dispositivos móveis sendo mais preciso *tabletes* e *smartphones*, onde já existe uma grande guerra entre duas gigantes da informática a Apple com seu sistema operacional IOS e a google com seu Sistema operacional android.

O mercado de *smartphones* tem um líder. por muito. android tem 52% de todas as vendas, como mostra a tabela abaixo.”^[1]

Table 2
Worldwide Smartphone Sales to End Users by Operating System in 3Q11 (Thousands of Units)

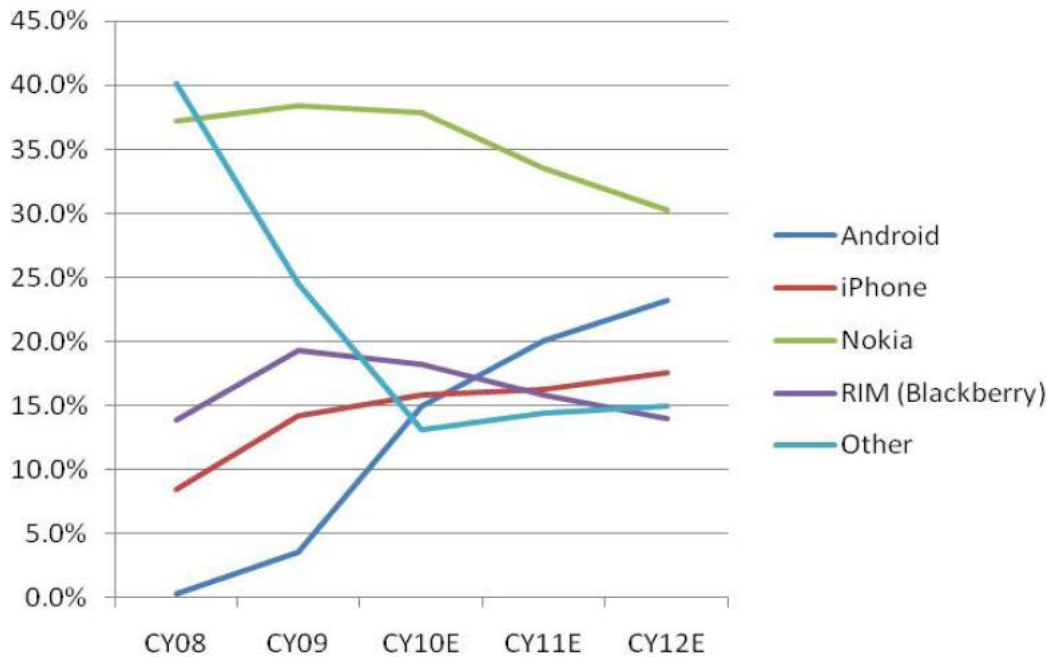
Operating System	3Q11 Units	3Q11 Market Share (%)	3Q10 Units	3Q10 Market Share (%)
Android	60,490.4	52.5	20,544.0	25.3
Symbian	19,500.1	16.9	29,480.1	36.3
iOS	17,295.3	15.0	13,484.4	16.6
Research In Motion	12,701.1	11.0	12,508.3	15.4
Bada	2,478.5	2.2	920.6	1.1
Microsoft	1,701.9	1.5	2,203.9	2.7
Others	1,018.1	0.9	1,991.3	2.5
Total	115,185.4	100	81,132.6	100

Source: Gartner (November 2011)

Figura 1 - Tabela dos sistemas móveis.

Segundo a Piper Jaffray empresa de pesquisas americana, o Android da Google terá mais da metade do mercado de *smartphones* e o iPhone IOS da Apple ficará entre 20-30% até 2012. Os outros sistemas operacionais para *smartphones* dividirão o restante do mercado. Os mais prejudicados serão o Blackberry OS da RIM e o Symbian da Nokia. Especulações dizem que a RIM migrará para o Android e a Nokia optará pelo Windows Phone 7 ou também pelo Android. Entretanto, a participação de mercado não significará lucro para a Google semelhantes aos da Apple. A previsão é que a Google fature US\$1,35 bilhão com o Android, ficando abaixo do lucro da Apple que só no primeiro trimestre do ano teve US\$1,5 bilhão de lucro. Veja a o gráfico abaixo. ^[2]

Piper Jaffray Smartphone Market Share Projections CY08-12



Source: Piper Jaffray

Figura 2 - Gráfico de faturamento das empresas de smartphones e tablets

Apple uma das líderes do mercado com números expressivos, alcançou uma receita de US\$28,3 bi neste último trimestre, provando que a empresa vai bem e pode alcançar mais de US\$ 100 bi no ano fiscal de 2011, embora represente um aumento de 66% em sua receita, firmando-se como a líder de mercado como fabricante individual de *smartphones* na questão de retorno financeiro, uma vez que a fatia do android é pulverizada em mais de 50 empresas.

O mercado de mobilidade tem um líder. a apple tem 52% do lucro, como mostra o gráfico abaixo. e a apple só vende 4% de todos os celulares.^[3]

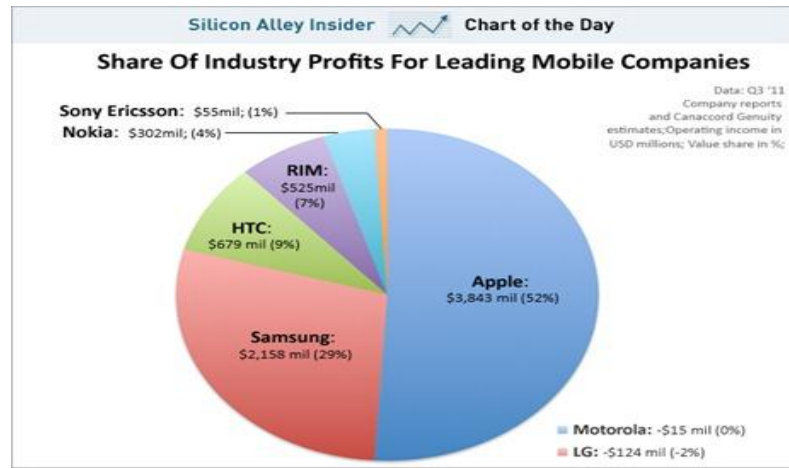


Figura 3 - Tabela dos lucros individuais dos smartphones

2.1. Primeiros Passo para Desenvolver para Dispositivos moveis

Depois de fazer um estudo de caso e escolher sua plataforma de desenvolvimento, o que é preciso antes de começar. Primeiro sua aplicação deve causar uma boa impressão, ela precisa ser intuitiva e bem objetiva. Faça como Steve Jobs.

“É um elitista que considera quase todas as outras pessoas idiotas. Mas faz *gadgets* tão fáceis de usar que mesmo um idiota consegue dominá-los. [A cabeça de Steve Jobs, Leander Kahney, 2008]

Quando for desenvolver um aplicativo é importante pensar no aplicativo desde o início. Você deve questionar-se com a pergunta, o que o usuário esta tentando fazer?

Citamos abaixo alguns pontos que você deve avaliar antes de iniciar o desenvolvimento da sua aplicação.

Leve em consideração que smartphone e tablets na sua grande maioria tem recurso de memória limitado entre 256 *Megabyte* e 1 *Gygabyte* de memória e boa parte dela já esta em uso.

Sua aplicação dever possuir um desempenho satisfatório, usuários de dispositivos móveis em sua grande maioria não tem muito tempo querem um aplicativo no qual ele resolva seu problema e volta a fazer suas atividades. Dispositivos móveis consomem mais bateria que o celular comum. Enviar o texto ou acessar internet custa dinheiro, nem todos os usuários de dispositivos moveis possui

planos de internet, também a internet é tipicamente lenta com taxa de transferência entre 10 mb a 300 mb.

Telas dos dispositivos móveis possui tamanho e resolução diferente e pequenas, dificultando assim o desenvolvimento. Você precisa pensar muito em cada tela e manter-se focado na tarefa específicas que o usuário esta realizando.

Hardware específicos, nem todos os aparelhos possuem câmera, GPS, Acelerômetro etc. Além disso cada fabricante possui processadores com capacidades diferentes. Por isso é muito importante você verificar si o dispositivo para qual você pretende desenvolver comporta os recursos que você pretender utilizar.

2.2. Conhecendo o IOS SDK

Antes de iniciar é necessário saber que o *IOS SDK* não funciona em outro sistema operacional que não seja o *MAC*, ou seja, não é possível trabalhar com *IOS sdk* sem *Mac*, mas para solucionar esse problema existe ferramentas como *titanium*,mas não será abordada neste artigo.

IOS Iphone operating system sistema operacional proprietário da Apple para dispositivos móveis sendo mais preciso Iphone,Ipad e Ipod touch um dos sistemas mais moderno e seguro entre os sistemas operacionais para dispositivos móveis da atualidade disponibilizando, para o desenvolvedor seu *IOS Sdk*,para o empresário sua loja de aplicação App store e para o usuário garantindo a qualidade e segurança das aplicações homologando as antes de disponibilizar para os usuarios.

IOS é um sistema operacional móvel da Apple desenvolvido originalmente para o iPhone, também é usado em Ipad. Foi lançado na "Macworld Conference & Expo" em 9 de janeiro de 2007 em 17 de outubro de 2007, a Apple anunciou que a *SDK* nativo para os desenvolvedores iniciar o desenvolvimento para IOS.^[4]

A interface do usuário do IOS é baseado no conceito de manipulação direta, utilizando gestos em multi-toque. A interação com o sistema operacional inclui gestos como apenas tocar na tela, deslizar o dedo, e o movimento de "pinça"

utilizado para se ampliar ou reduzir a imagem. Acelerômetros internos são usados por alguns aplicativos para responder à agitação do aparelho (um resultado comum é o comando de desfazer) ou rodá-la em três dimensões (um resultado comum é a mudança do modo retrato para modo paisagem).

O IOS esta dividido em quatro camadas, conforme pode ser visualizado na imagem abaixo.



Figura 4 - Cocoa Touch

Os framework nesta camada diretamente suporte a aplicações baseadas em iOS. Eles incluem estruturas, tais como Game Kit, Map Kit e iAd. A camada *Cocoa Touch* e o *Core framework* de Serviços cada um tem um framework *Objective-C* que é especialmente importante para o desenvolvimento de aplicações para iOS.

Os *framework* e serviços nesta camada depende da camada Core Services e prestação de serviços gráficos e multimídia para a camada *Cocoa Touch*.

Eles incluem Core Graphics, Core text, OpenGL ES, Core Animation, *AVFoundation*, Audio Core, e reprodução de vídeo^[5]

Nesse camada são desenvolvido dos jogos e utilização de bibliotecas de som e áudio.

Os *frameworks* nesta camada oferecem serviços essenciais, tais como manipulação de string, recolha gestão de rede, utilitários de URL, gerenciamento de contatos e preferências. Eles também oferecem serviços baseados em recursos de hardware de um dispositivo, como o GPS, bússola, acelerômetro e giroscópio. Exemplos de *frameworks* nesta camada são *Core Location*, *Motion Core*, e configuração do sistema.

Este *frameworks* inclui tanto Foundation e Core Foundation, frameworks que fornecem abstrações para tipos de dados comuns, como string e coleções. O Core camada *Frameworks* também contém dados Core, uma estrutura para gerenciamento objeto gráfico e persistência de objetos.

Engloba o ambiente do *kernel, drivers* e interface básicas do sistema operacional fornece um conjunto de *interfaces*, através de bibliotecas de sistema, para acesso a recursos do sistema operacional. As interfaces são baseadas em C e fornecer suporte para: - *Threading* (POSIX *threads*), *Memory allocation* - *Networking* (BSD *sockets*), Acessado em ^[6]

Objective-c uma linguagem orientada a objeto que utiliza o padrão de projeto MVC é possui um moderno gerenciador de memória.

Objective-C é uma linguagem projetada para permitir a programação orientada a objetos. É definida como um pequeno, mas poderoso conjunto de extensões ao padrão da linguagem *ANSI C*. Seus aditamentos a C são essencialmente baseados em *Smalltalk* que é uma das primeiras linguagens de programação orientada a objeto. A linguagem *Objective-C* foi concebida para permitir a programação orientada a objeto de forma simples e direta.^[7]

A linguagem Objective-c por muitos considerado uma grande dificuldade no desenvolvimento para IOS, por ter uma sintaxe diferente das linguagens convencionais, outra dificuldade e que *objective-c* não possui *Garbage collection* obrigando o programador a gerencia os objetos da memória.

Uma diferença de *objective-C* e linguagem como Java e C# é que as classe em *objective -c* é implementada através de 2 arquivos: o arquivo .h o de declaração onde si define os métodos e atributos da classe assim como toda sua estrutura. Para melhor compreensão veja a figura abaixo.


```

@interface Card : CocosNode {
    int value;
}

-(id) initWithValue: (int) value;
-(void) dealloc;

@property (readwrite) int value;

@end

```

Figura 5 - Classe H

O arquivo. m. é onde fica a implementação do métodos a serem desenvolvidos como pode ser visualizado na figura abaixo.

```

#import "Card.h"

@implementation Card

-(id) initWithValue: (int) value{
    // .....
}

-(void) dealloc{
    // .....
}

@synthesize value;

@end

```

Figura 6 - Classe M.

MVC padrão de projeto adota pela Apple para melhorar o desenvolvimento de aplicações, uma das principais vantagens do *mvc* é a possibilidade do desenvolvedor utilizar um *controller* em varias *view's* fazendo assim uma economia de memória.

A arquitetura MVC Modelo visão e controle fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multicamadas.[8]

Mvc é um padrão de projeto onde a camada *view* não pode se comunicar diretamente com a *model* sem antes se comunicar com a de controle em objective-c funciona da seguinte forma quando você está na *view* *click* no botão ele aciona o *IBAction* que é responsável por fazer uma ação e quando retorna a informação será exibido em um objeto *IBOutlet*. Veja a figura abaixo.

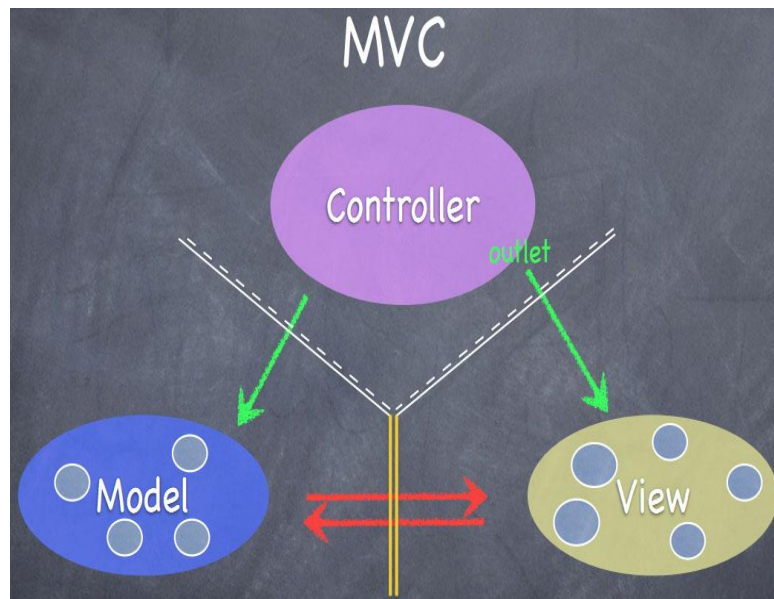


Figura 7 - Model View Controller

Xcode é a principal ide de desenvolvimento para IOS, existem outras alternativas, mas nenhuma possui tantos recursos e compatibilidade com o IOS quanto a *xcode*.

Xcode é um ambiente de desenvolvimento integrado (*IDE*), que fornece todas as ferramentas que você precisa para criar e gerenciar seus projetos de IOS e arquivos de origem, montar sua interface de usuário, construir o seu código em um arquivo executável e executar e depurar o seu código seja em Simulator ou em um dispositivo. *Xcode* incorpora uma série de recursos para tornar mais fácil o desenvolvimento de aplicações^[9]

Xcode oferece uma lista de *templates* para você criar suas aplicações, ele cria toda a estrutura do seu projeto diretório subdiretório, possui suporte interno para troca de versão, editor interno para arquivos de recursos como *plist*.

Mesmo você utilizando ide's de terceiro você ainda precisa do *xcode* para publicar sua aplicação uma vez que IOS tem um fonte de distribuição única App

Store, onde você precisa configura a build para enviar sua aplicação para validação e posteriormente publicação. Veja a figura abaixo.

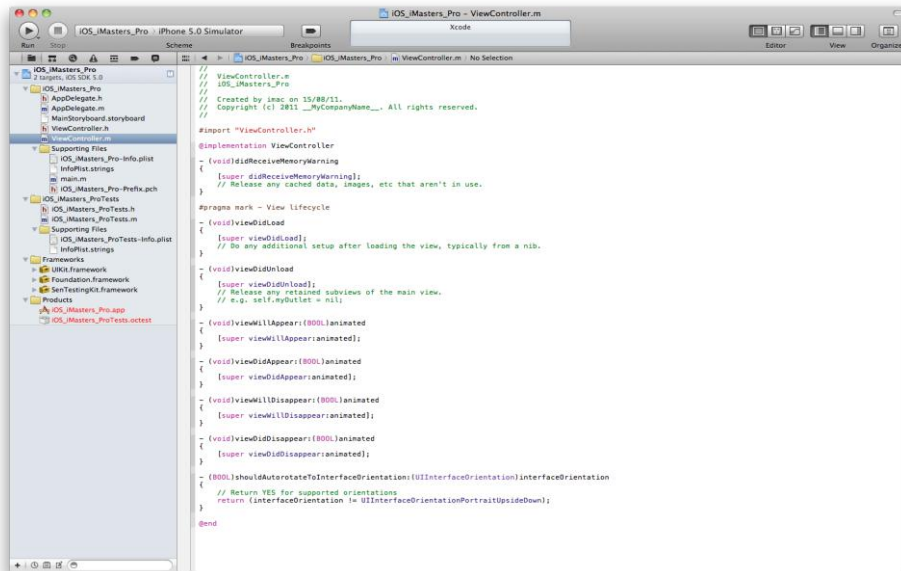


Figura 8 - Model View Controller

a) Interface Builder

Interface Builder é a ferramenta usada para montar a interface visual de sua aplicação, ou seja, o designer. Usando o *Interface Builder*, para criar o designer da sua aplicação você precisa apenas ir arrastando e soltando componentes pré-configurados (caixa de texto, *label*, *maps*) dentro da sua view. Os arquivos criados no *interface builder* tem extensão XIB. Como podem serem visualizado na figura abaixo.



Figura 9Interface Builder

“Use um *IBOutlet* quando você precisa de uma referencia com o interface builder para que seus componentes gráficos possa ser conectado com o *interface builder* ele precisa ser declarado como *IBOutlet*”(PILONE,2011,P.25)

“*IBAction* quando você vai fazer alguma interação com *viewController* chamar o método que vai realizar alguma ação,fazer a soma de dois números por exemplo”. (PILONE,2011,P.25)

b) Instruments

O ambiente *Instruments* permite analisar o desempenho de seus aplicativos IOS enquanto executa no simulador ou em um dispositivo. *Instrumentos* de coleta de dados do seu aplicativo em execução e apresenta os dados em uma exibição gráfica chamada visão *timeline*. Você pode coletar dados sobre o uso de seu aplicativo de memória, atividade do disco, a atividade da rede e desempenho gráfico. A visão *timeline* pode exibir todos os tipos de informações lado a lado, permitindo-lhe relacionar o comportamento global da sua aplicação, não apenas o comportamento em uma área específica. ^[10]

Com o *instrumente* você pode realizar teste de desempenho da sua aplicação ,verificar vazamento de memória,consumo de banda da rede antes de enviar para Apple para validação.

c) **Debug**

Um dos passos naturais no desenvolvimento de software é alguma coisa dar errado no seu código e o aplicativo não funcionar. Para isso que existe o *debug*. Xcode oferece vários recursos de depuração integrados, que oferecem uma visão geral do código de sua aplicação e detalhes quando você precisar deles.

- **Debug via console**

Há um console interno com informações de *debug* e log. Ele lhe dá avisos ao compilador para identificar problemas.

Debug no console é feito através da função de `NSLog` colocados dentro do seu código no *xcode*.

Os *IOS frameworks*, como *UIKit*, envia entradas de log para o console para indicar, entre outras coisas, quando ocorre um evento inesperado. Você pode emitir mensagens de console em seus aplicativos iOS, também. Uma forma de emitir mensagens de console é usar a função `NSLog`. Logs do console pode ajudar a analisar a lógica do aplicativo e rastrear *bugs*. Ao executar o aplicativo em um simulador.^[11]

```

WhatATool - Debugger Console
10.6 | Debug | x86_64
Overview Breakpoints Build and Run Tests Source Run Clear Log

2018-04-08 13:42:04.359 WhatATool[15485:apf] Section 1: Strings as file system paths
2018-04-08 13:42:04.361 WhatATool[15485:apf] ~~~~~
2018-04-08 13:42:04.362 WhatATool[15485:apf] Home path is /Users/george
2018-04-08 13:42:04.362 WhatATool[15485:apf] Now let's split out the path components...
2018-04-08 13:42:04.362 WhatATool[15485:apf] /
2018-04-08 13:42:04.363 WhatATool[15485:apf] Users
2018-04-08 13:42:04.363 WhatATool[15485:apf] george
2018-04-08 13:42:04.363 WhatATool[15485:apf] ~~~~~
2018-04-08 13:42:04.364 WhatATool[15485:apf] Section 2: Finding out a bit about our own
process
2018-04-08 13:42:04.364 WhatATool[15485:apf] ~~~~~
2018-04-08 13:42:04.364 WhatATool[15485:apf] Process Name: 'WhatATool' Process ID: '15485'
2018-04-08 13:42:04.365 WhatATool[15485:apf] ~~~~~
2018-04-08 13:42:04.365 WhatATool[15485:apf] Section 3: A little bookmark dictionary
2018-04-08 13:42:04.365 WhatATool[15485:apf] ~~~~~
2018-04-08 13:42:04.365 WhatATool[15485:apf] Key: 'Stanford on iTunes' URL: 'http://
itunes.stanford.edu'
2018-04-08 13:42:04.366 WhatATool[15485:apf] Key: 'Stanford University' URL: 'http://
www.stanford.edu'
2018-04-08 13:42:04.366 WhatATool[15485:apf] Key: 'Stanford Mall' URL: 'http://
stanfordshop.com'
2018-04-08 13:42:04.367 WhatATool[15485:apf] ~~~~~
2018-04-08 13:42:04.367 WhatATool[15485:apf] Section 4: Selectors, Classes and
introspection
2018-04-08 13:42:04.368 WhatATool[15485:apf] ~~~~~
Debugging of 'WhatATool' ended normally.
Succeeded

```

Figura 10 - Xcode Debug console

- **Debug via Breakpoint**

O Desenvolvedor vai colocar uma pequena seta azul ao lado da linha que ele deseja que sua aplicação pare quando sua aplicação chegar nesta linha de código ela vai exibir no console e também em uma visão de pilha da sua aplicação em uma janela para você inspecionar a variável.

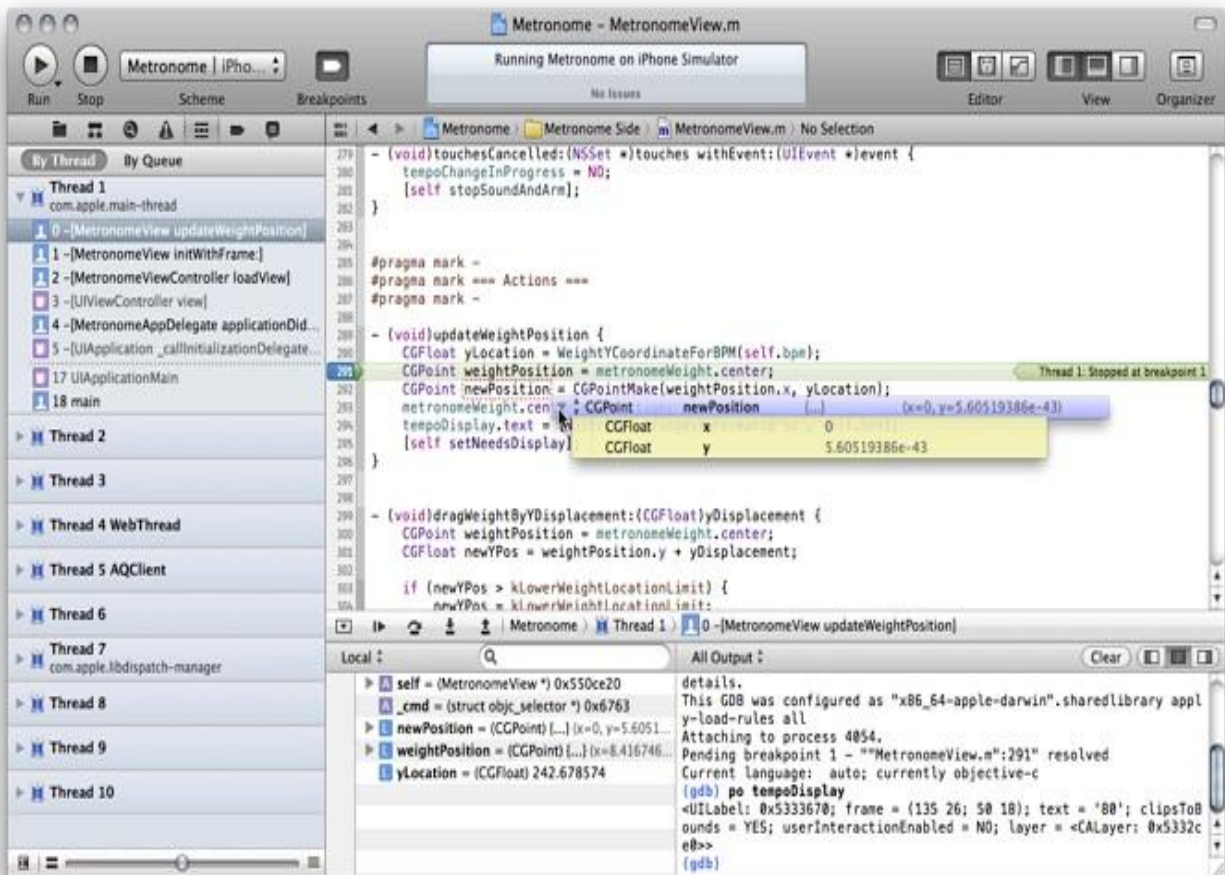


Figura 11 - Debug xcode Via Breakpoint

d) Simulador

Serve para testar seu aplicativo no Mac sem a necessidade de um iPhone ou iPad, você pode usar o iOS Simulator, que engloba a maioria das funções de um aparelho, mas para alguns recursos como GPS, *OpenGL* e câmera o simulador não reproduz perfeitamente, por isso é recomendado que você adquira um iPhone ou iPad para testar suas aplicações.

3. ANDROID

3.1. Historia

Android primeira sistema operacional de código fonte aberto para dispositivo atualmente líder do mercado em numero de aparelhos que utiliza Android como seu sistema operacional.

A plataforma Android foi concebida inicialmente pelo Google. A plataforma está sendo e será mantida pelo Open Handset Alliance, que é um grupo formado por mais de 30 empresas (de tecnologias de dispositivos móveis, provedoras de serviços móveis, fabricantes, etc) as quais se uniram para inovar e acelerar o desenvolvimento de aplicações, serviços, trazendo aos consumidores uma experiência mais rica em termos de recursos, menos dispendiosa em termos financeiros para o mercado móvel. Pode-se dizer que a plataforma Android é a primeira plataforma móvel completa, aberta e livre.

Esta plataforma foi desenvolvida utilizando o sistema operacional Linux. Sendo assim, todas as características intrínsecas deste sistema foram incorporadas, bem como sistema de arquivos, o kernel, os servidores de terminais (X server), etc^[12]

Steve Jobs em sua biografia afirmava que o android criado por Eric Schidt CEO da Google e Ex membro do conselho da Apple seria uma copia mal feita do IOS, após o lançamento do primeiro aparelho com android, Steve Jobs enviou uma mensagem agressiva a Schmidt na qual acusava a Google de um grande roubo.

4. CONCLUSÃO

Ao longo do desenrolar deste projeto pode-se perceber a grande procura das pessoas por dispositivos móveis smartphones e tablets. Com isso cada dia mais

surgiu novas oportunidades no mercado precisando nós estarmos preparados para ocupá-las .

Entre os dispositivos móveis smartphones e tablets existe duas grandes empresas que se destacam Apple com seu sistema operacional IOS líder como fabricante individual e a Google que é a principal empresa associada da Open Handset Alliance com seu sistema operacional Android líder com o sistema operacional mais usado.

Comparando o Android sdk e IOS sdk, no quesito sdk o IOS esta um pouco na frente do android mesmo que para desenvolver para IOS você precise fazer um certo investimento,o resultado obtido é notório exemplos disso são segurança as aplicações são homologadas antes de irem para a loja garantindo para o usuário a qualidade do produto gerando mais credibilidade para as aplicações, designer mais robusto e intuitivo usufruindo dos recursos de fácil utilização da interface builder, facilidade de migração uma aplicação feita para iPhone facilmente migrada para Ipad ,uma loja mais rentável uma vez que 60% das aplicações baixadas na android market são free.Ambos os Sdk's são bons mais é importante que se conheça bem as características de ambos,antes de ingressar na carreira de desenvolvedor de aplicativos móveis.

REFERÊNCIAS

[1] **Meira, Silvo TERRA. Mobilidade: fragmentação, diversidade e execução. Entre "americanos**

Disponível em: <<http://smeira.blog.terra.com.br/2011/11/16/mobilidade-fragmentao-diversidade-e-execuo-entre-americanos/>>Acessado em 30 de novembro de 2011

[2] **EFAGUNDES .Apple e Google dominarão o Mercado de Smartphones**

Disponível em:<<http://efagundes.com/wp-blog/?tag=android> >Acessado em 01 de novembro de 2011.

[3] **Meira, Silvo TERRA. Mobilidade: fragmentação, diversidade e execução. Entre "americanos**

Disponível em: <<http://smeira.blog.terra.com.br/2011/11/16/mobilidade-fragmentao-diversidade-e-execuo-entre-americanos/>>Acessado em 30 de novembro de 2011

[4] **APPLE DEVELOPER.IOS.**Disponível em :

< <http://developer.apple.com/devcenter/ios/index.action> >Acessado em 25 de setembro de 2011.

[5] **APPLE DEVELOPER. Cocoa Touch.** Disponível em:

<<http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CocoaFundamentals/CocoaFundamentals.pdf>>Acessado em 28 de setembro 2011.

[6] **STANFORD. CS 193P iPhone Application Development**

Disponível em :< <http://www.stanford.edu/class/cs193p/cgi-bin/drupal/downloads-2010-fall>>Acessado em 01 de setembro de 2011.

[7] **APPLE DEVELOPER. Objective-c.** Disponível em:

<<http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>>.Acessado em 15 de novembro de 2011.

[8] **MACORATTI.NET.MVC - Model-View-Controller - Introdução**

Disponível em:<http://www.macoratti.net/08/06/asp_mvc1.htm>Acessado em 10 de dezembro de 2011.

[9] **APPLE DEVELOPER. Xcode.** Disponível em:

<http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSDeveloperTools/iPhoneOSDeveloperTools.html#//apple_ref/doc/uid/TP400-SW1>. Acessado em 13 de novembro de 2011.

[10] **Apple Developer. Instruments.** Disponível em:

<http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSDeveloperTools/iPhoneOSDeveloperTools.html#//apple_ref/doc/uid/TP40007898-CH7-SW1>Acessado em 12 de novembro de 2011.

[11] **APPLE DEVELOPER.Debug**

Disponível em:

<http://developer.apple.com/library/ios/#documentation/Xcode/Conceptual/ios_development_workflow/130-Debugging_Applications/debugging_applications.html>Acessado em 07 de setembro de 2011.

[12] **WEBMEDIA. ARTIGO-WEBMOBILE-18-ANDROID**

Disponível em: <<http://www.devmedia.com.br/post-9350-Artigo-webmobile-18-android--um-novo-paradigma-de-desenvolvimento-movel.html>>Acessado em 29 de novembro 2011.

TECMUNDO. Steve Jobs estava determinado a destruir o Android.

Disponível em: <<http://www.tecmundo.com.br/steve-jobs/14584-steve-jobs-estava-determinado-a-destruir-o-android.htm>>Acessado em 29 de novembro de 2011.

DIEGOSANTOS. Interface Builder. Disponível em: <<http://diegosantos.wordpress.com/category/mobilidade/>>Acessao em 14 de novembro de 2011.

Pilone,Dan;Pilon,Tracey.Use a Cabeça Desenvolvendo para iPhone.Rio de Janeiro:Alta Book,2011.

Isaacson,Walter.Steve Jobs:Berilo Vargas.Denise Bottmann,Pedro Maia Soares-São Paulo:Companhia das letras 2011.

ANDROID DEVELOPER.Android

Disponível em :<<http://developer.android.com/index.html> >Acessado em 10 de dezembro de 2011.