
APLICATIVOS PARA IPHONE – UMA ABORDAGEM PRÁTICA

IPHONE APPLICATIONS – A PRACTICAL APPROACH

Ângelo Cabral Silva¹

RESUMO: Este artigo tem como objetivo explicar de modo geral para aqueles que desejam começar a desenvolver aplicativos para iPhone, informando quais as ferramentas necessárias para seu desenvolvimento.

PALAVRAS-CHAVE: Aplicativos. iPhone. Mobile

ABSTRACT: *This article aims to explain in general those who wish to start developing applications for iPhone, which states the necessary tools for development.*

KEYWORDS: *Applications. iPhone. Mobile*

¹ Aluno de graduação de Ciência da Computação da Universidade UNIVERITAS/UNG.

1. Introdução

Para desenvolver um aplicativo para o iPhone precisamos de um computador que rode o sistema operacional IOS (*Internet operating system* - Sistema operativo de internet) da Apple, além de ferramentas para desenvolvimento, que podem ser acessadas no próprio site de Apple, sendo necessário o cadastro de desenvolvedor de aplicativos Apple mediante cobrança de taxa direcionada a cada categoria de desenvolvimento.

A ferramenta disponível para download é o IOS *Software Development Kit* ou SDK, que é composto basicamente pelo ambiente de desenvolvimento Xcode e pela sua API (*Application Programming Interface* - Interface de programação aplicativos) de componentes. ” (Marzullo 2012). Para a criação do aplicativo IOS, deve-se aprender a linguagem de programação orientada a *Objective-C*.

2. Objective-C

A linguagem *Objective-C* foi criada para o desenvolvimento de rotinas programacionais que se utilizam om Programação Orientada a Objetos. Para se criar uma classe, são gerados dois arquivos com extensão **.h** (cabeçalho da classe) e extensão **.m** (Implementação). O arquivo **.h** inicia com **@interface** e finalizamos com **@end** que são os marcadores do cabeçalho. No arquivo **.m** definimos **@implementation** e **@end** para sua implementação. (Marzullo, 2012). Segue exemplos abaixo;

```
@interface Pessoa: Object
{
    NSString *nome;
    NSString *logradouro;
}
...
@end
```

Figura 1: Imagem retirada do livro iPhone na Prática – Aprenda passo a passo a desenvolver soluções para IOS. Pág. 42.

```
@implementation Pessoa
@synthesize nome;
@synthesize logradouro;
@end
```

Figura 2: Imagem retirada do livro iPhone na Prática – Aprenda passo a passo a desenvolver soluções para IOS. Pág. 44.

Em *Objective-C* existe a necessidade de alocar e liberar a memória quando necessário durante a execução do programa, lembrando que não se deve liberar da memória os objetos que são referenciados na aplicação. Deve-se levar em conta o gerenciamento da memória por estar utilizando aparelhos com hardwares limitados. (Marzullo, 2012).

Para mais informações O Tutorial “*The Objective-C Programming Language*” está disponibilizado para download no site da Apple (2012) <<https://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>>.

3. SDK

O SDK é composto pelo Xcode, um ambiente integrado de desenvolvimento de software (*Integrated Development Environmet* - IDE), um ambiente poderoso de desenvolvimento da Apple integrado para criar ótimos aplicativos para Mac, iPhone e iPad (Apple 2012) que possui ferramentas para criar e apurar erros do código fonte, compilando aplicativos e ajuste do desempenho dos aplicativos criados (Mark; LaMarche 2009). Integra ainda, a *Interface Builder*, ferramenta visual para desenhar a interface do aplicativo, o IOS Simulator como o nome diz simula o aplicativo e o *Cocoa Touch*, que é uma biblioteca (API) orientada a objetos.

3.1 Xcode



Figura 3: Tela inicial do Xcode. (<https://itunes.apple.com/us/app/xcode/id497799835?ls=1&mt=12>)

A interface agrega edição de código, interface design com Interface Builder, testes e depuração, tudo dentro de uma única janela. O embutido da Apple LLVM compilador sublinha erros de codificação enquanto digita e ainda é inteligente o suficiente para corrigir os problemas automaticamente. (Apple 2012).

3.2 Interface Builder

Interface Builder, ou IB, cria interfaces para os aplicativos e manipula arquivos **.XIB**. Um arquivo **.nib** é basicamente a versão compilada (binário) de um arquivo XIB que em si é um arquivo XML que é gerenciado pelo IB (Nahavandipoor, 2012).

3.3 IOS Simulador



Figura 4: Execução do simulador IOS sem linhas de códigos inseridas. (<http://macmagazine.com.br/2010/09/12/como-criar-seu-primeiro-projeto-para-iosiphone/>)

O tutorial disponibilizado pela Apple (2012) afirma que com o IOS Simulador pode-se fazer simulações com muitas ações que são realizadas no dispositivo real, como Bloqueio, gestos, rotações para a esquerda e direita, aviso de memória, entre outros. Porém, o IOS simulador não acessa os dispositivos de acelerômetro ou câmera.

3.4 Cocoa Touch

O Cocoa Touch é um framework com várias coletâneas de objetos para desenvolvimento (Marzullo 2012).

Segundo Apple (2012) é desenvolvido em Objective-C, para interfaces, baseadas com toque e otimização. Algumas das estruturas mais utilizadas são o: Core Animation, Core Audio e Core Data, além de outras.

3.5 SQLite

O SQLite é uma biblioteca implementada em um Banco de Dados, que não necessita de um servidor para gerencia-lo, ocupa pouco espaço na memória, mas não suporta trabalhar com armazenamento de: imagens, vídeos e áudio, e nem trabalhar com controle concorrente. (Marzullo 2012).

O SQLite é desenvolvido em Objective-C, onde todo código foi escrito do zero não estando contaminado por nenhum outro código licenciado. Pertence a um domínio público, porém mesmo que o SQLite seja de domínio público não significa que não possa obter uma licença, sendo por motivos a utiliza-lo em uma jurisdição que não reconheça o domínio público ou armazenamento de um documento como prova legal ou mesmo, por ser obrigado pelo departamento jurídico a obter a licença. (Site Oficial da SQLite 2012).

4. iCloud

iCloud é uma infraestrutura de nuvem da Apple, que é dado o nome de nuvem a um tipo de serviço que armazena informações em um local centralizado, onde o usuário não tem acesso físico a memória de disco e o objetivo de uso do iCloud do ponto de vista de um programador, é dar a capacidade do usu-

ário de ter seus dados transferidos de uma máquina para outra. (Nahavandipoor 2012)

5. Web Services – Serviços Web

Um serviço web é uma aplicação que roda em um servidor web. Um aplicativo do iPhone pode pedir a um serviço web para executar o método que o serviço web complements. Normalmente, os métodos de um serviço web coleta dados do aplicativo iPhone e armazena-o em um banco de dados ou a informação de retorno dos bancos de dados para o aplicativo para iPhone (ou ambos). Os dados transferidos entre um serviço web e uma aplicação cliente normalmente é formatado em formato XML ou JSON. (Conway; Hillegass 2010).

Considerações Finais

Para que seja possível desenvolver aplicativos para iPhone há a necessidade da ferramenta SDK que é o ambiente de desenvolvimento do Xcode composta com o Interface Builder, Cocoa Touch e IOS Simulator, sendo que esta ferramenta funciona (roda) em um sistema operacional IOS. A integração dessas ferramentas facilitou muito o desenvolvimento de aplicativos, onde a estrutura da linguagem Objective-C se assemelha muito a Java, sendo imprescindível adquirir conhecimento prévio da linguagem para a ferramenta SDK que a utiliza para desenvolvimento das aplicações.

O Xcode não é difícil de utilizar então o maior desafio será a linguagem, principalmente no que diz respeito a armazenamento de dados, onde o artigo comenta sobre o SQLite, porém podemos utilizar de outros tipos de Banco de dados, mas cada um diferenciara na estrutura do código a ser desenvolvido.

REFERÊNCIAS

- Apple. [S.l.]: [s.n.], 2012. **Cocoa-Touch**. Disponível em: <<https://developer.apple.com/xcode/interface-builder/>>. Acesso em: 24 maio 2017.
- <<https://developer.apple.com/technologies/ios/cocoa-touch.html>>. Acesso em 23/05/2017
- Apple. [S.l.]: [s.n.], 2012. **IOS simulator**. Disponível em: <<https://developer.apple.com/technologies/tools/>>. Acesso em: 24 maio 2012.
- Apple. [S.l.]: [s.n.], 2012. **The objective-C programming language**. Disponível em: <<https://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>>. Acesso em: 24 maio 2017.
- Apple. [S.l.]: [s.n.], 2012. **Tools workflow guide for IOS**. Disponível em: <http://developer.apple.com/library/ios/#documentation/Xcode/Conceptual/ios_development_workflow/25-Using_iOS_Simulator/ios_simulator_application.html#/apple_ref/doc/uid/TP40007959-CH9-SW1>. Acesso em: 30 maio 2017.
- Apple. [S.l.]: [s.n.], 2012. **Xcode**. Disponível em: <<https://developer.apple.com/xcode/index.php>>. Acesso em: 16 maio 2012.
- Apple. [S.l.]: [s.n.], 2012. **Which developer program is for you?**. Disponível em: <<https://developer.apple.com/programs/which-program/>>. Acesso em: 16 maio 2017.
- CONWAY, Joe; HILLEGASS, Aaron. **iPhone programming the big nerd ranch guide**. Estados Unidos da América: Big nerd ranch, 2010.
- MARK, Dave; LAMARCHE, Jeff. **Dominando o desenvolvimento no iPhone: explorando o SDK do iPhone**. Rio de Janeiro: Alta Books, 2009.
- MARZULLO, Fabio. **iPhone na prática: aprenda passo a passo a desenvolver soluções para IOS**. São Paulo: Novatec, 2012.
- NAHAVANDIPOOR, Vanda. **IOS 5 programming CookBook**. Sebastopol: O'Reilly, 2012.
- SQLite**: site oficial. [S.l.]: [s.n.], 2017. Disponível em: <<http://www.sqlite.org>>. Acesso: em 15 abr. 2017.