

# App com Nome e Idade

X-Code

Prof. Agesandro Scarpioni

[agesandro@fiap.com.br](mailto:agesandro@fiap.com.br)

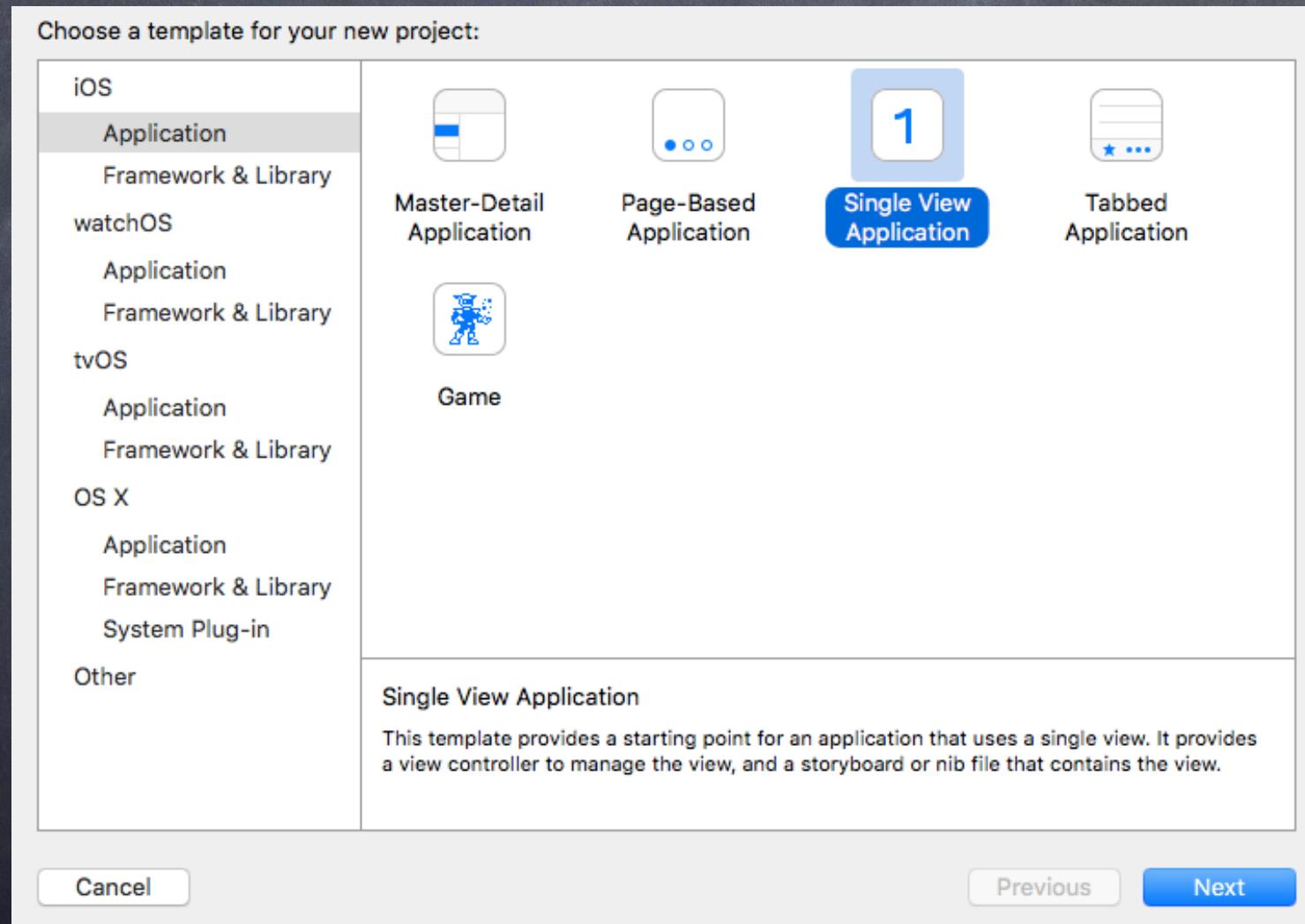
# Nome e Idade

💡 Vamos criar o aplicativo de uma forma diferente:

- Nesta aula vamos criar um projeto utilizando templates para otimizar o processo de desenvolvimento do App, tudo de forma automática.
- Em um segundo momento iremos incluir um IBAction e um IBOutlet de forma manual.
- A intenção é demonstrar as funcionalidades da ferramenta vista de dois ângulos diferentes.

# O App

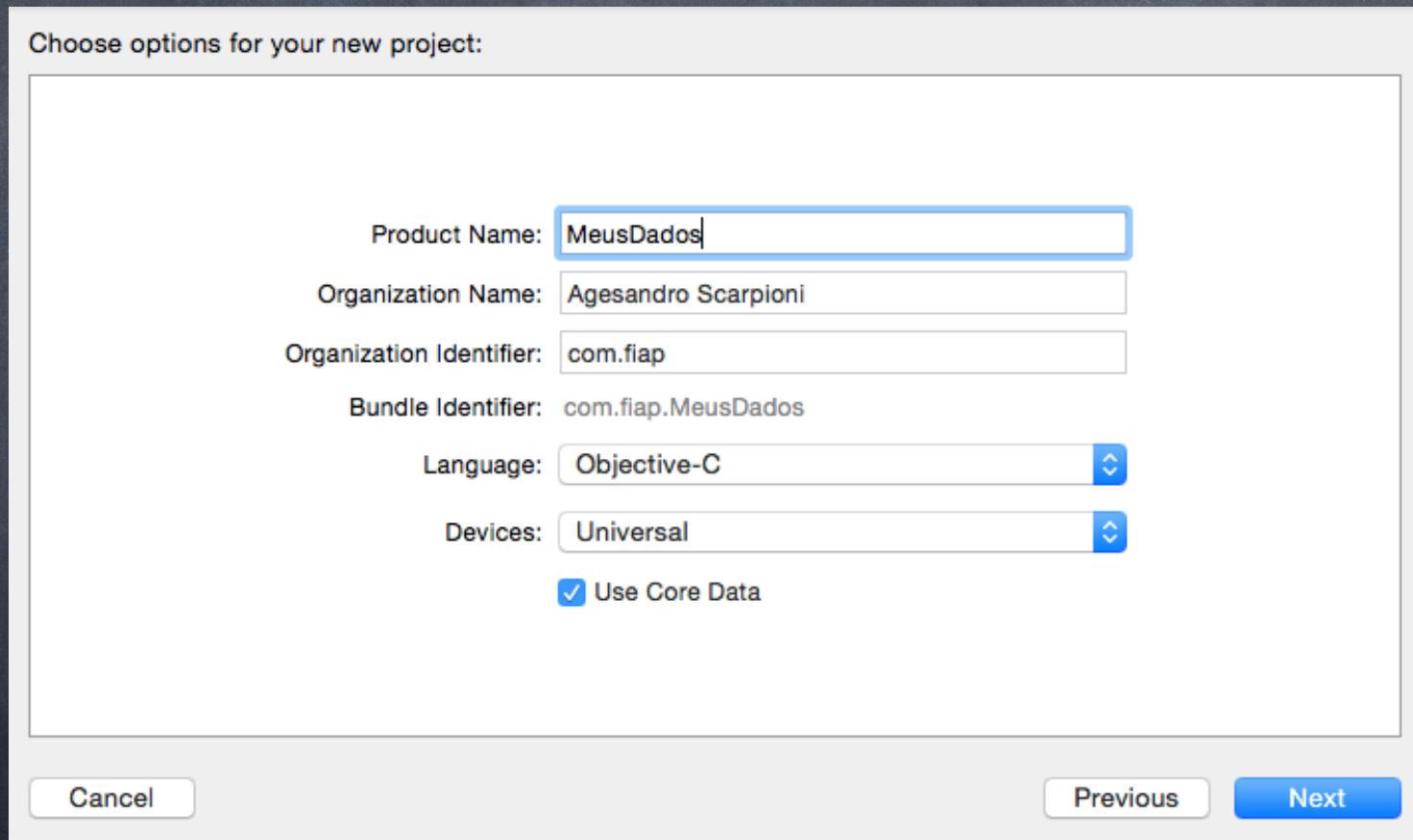
- Clique em File -> New Project -> Application -> Single View Application.



OBS: Este é o template que já cria uma viewController e uma classe com os arquivos .h e .m, inclusive o delegate já tem a referencia dessa classe.

# O App

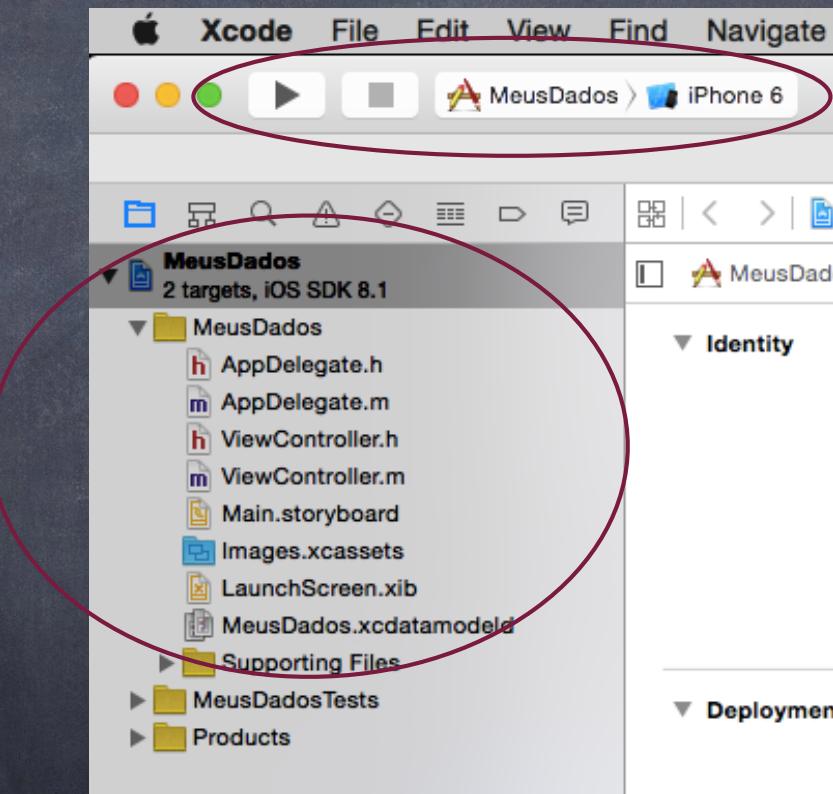
- Preencha com os dados abaixo, lembre-se que o Company Identifier é como se fosse o pacote no Java ou o namespace do VB, marque o 1º e 2º check box.



OBS: O device Universal serve para indicar que o app será tanto para iPhone quanto para iPad.

# Tudo em seu lugar

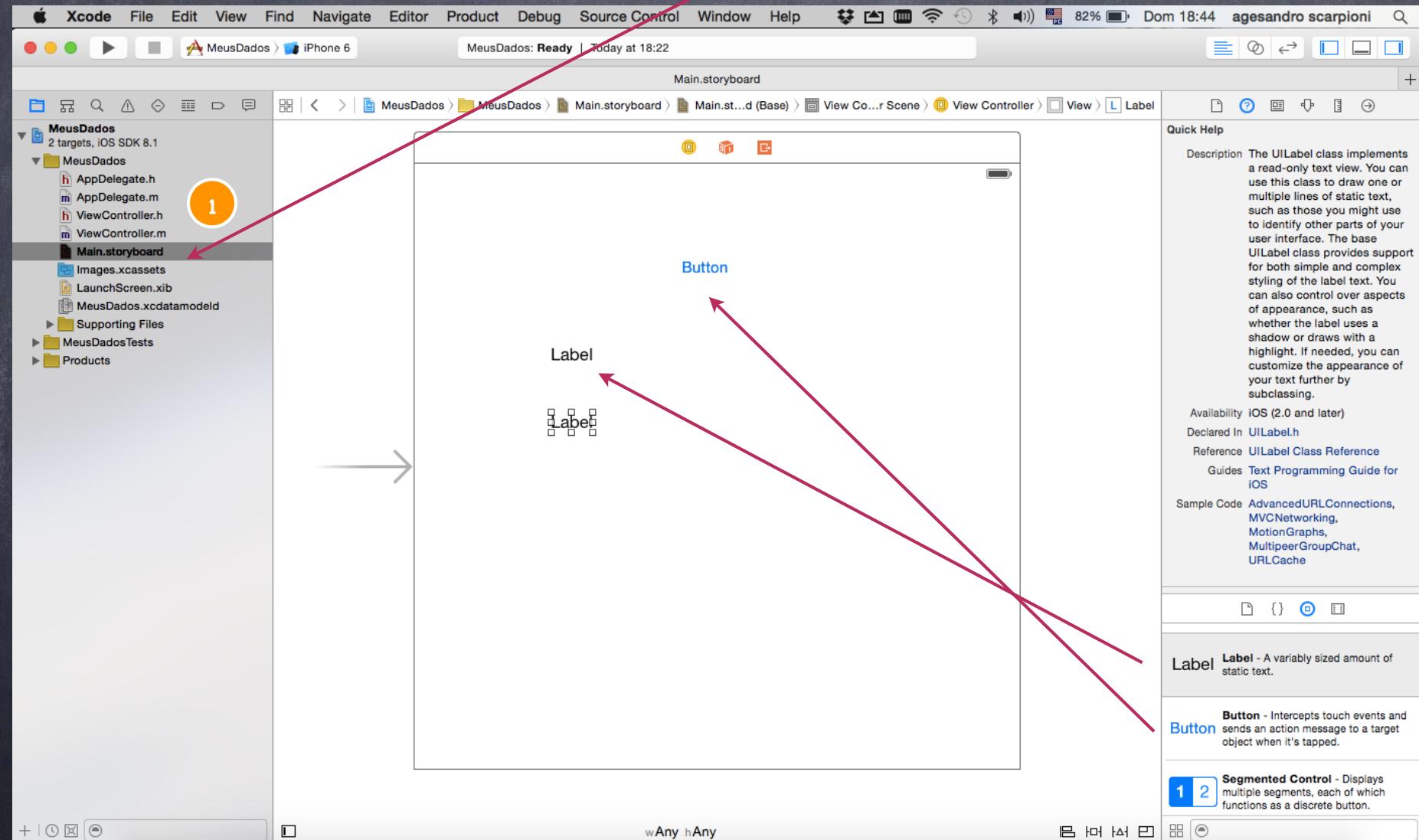
- Note que utilizando este template foi incluída uma classe chamada ViewController .h e .m, o delegate que já possui a classe referenciada e o Storyboard (iPhone e iPad).



*Obs:* Se você executar o programa agora, irá aparecer a tela de iPad ou iPhone no simulador sem termos que programar ao menos uma linha.

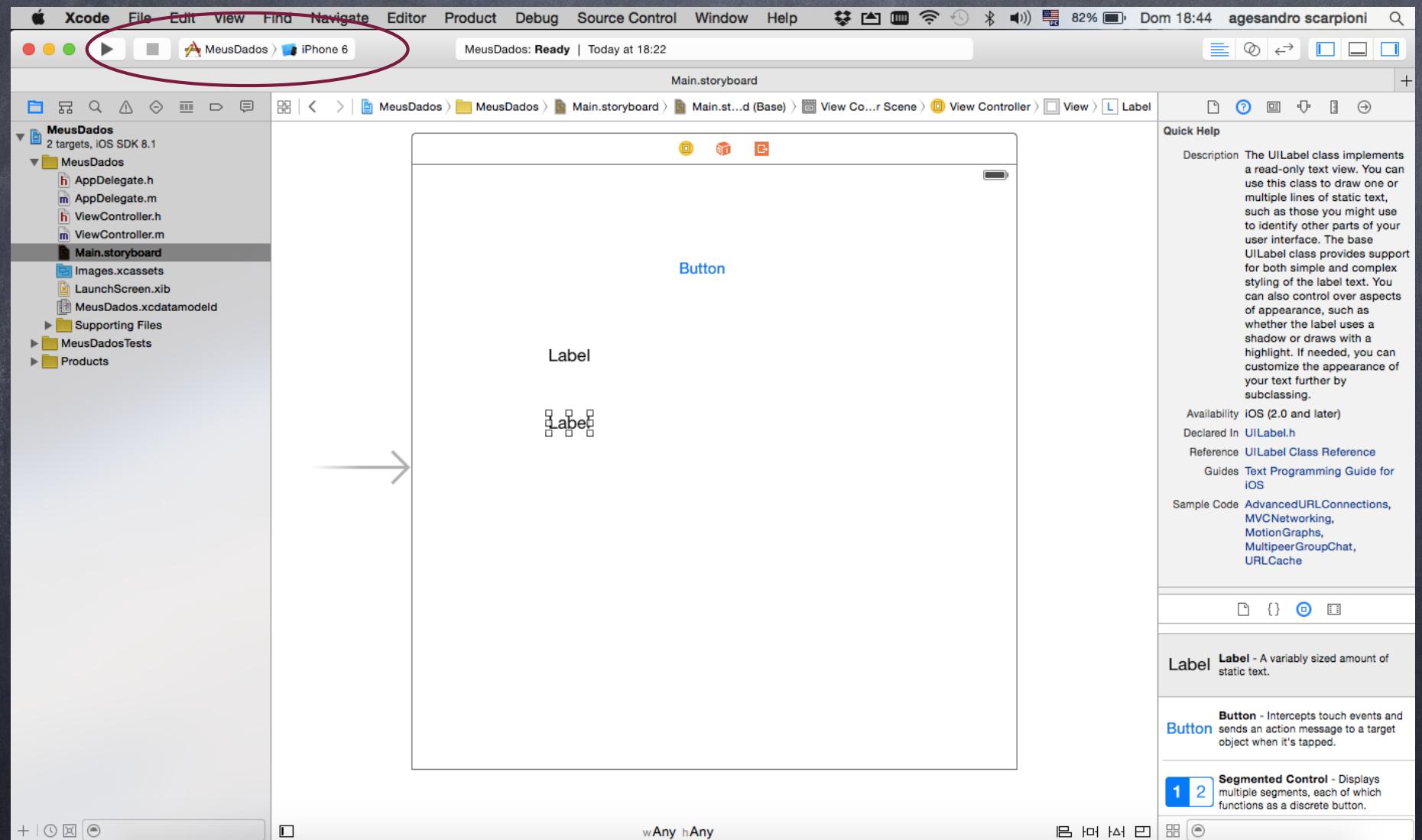
# Adicionando objetos

- Inclua dois labels e um botão no Storyboard, altere o texto do botão para Exibir.



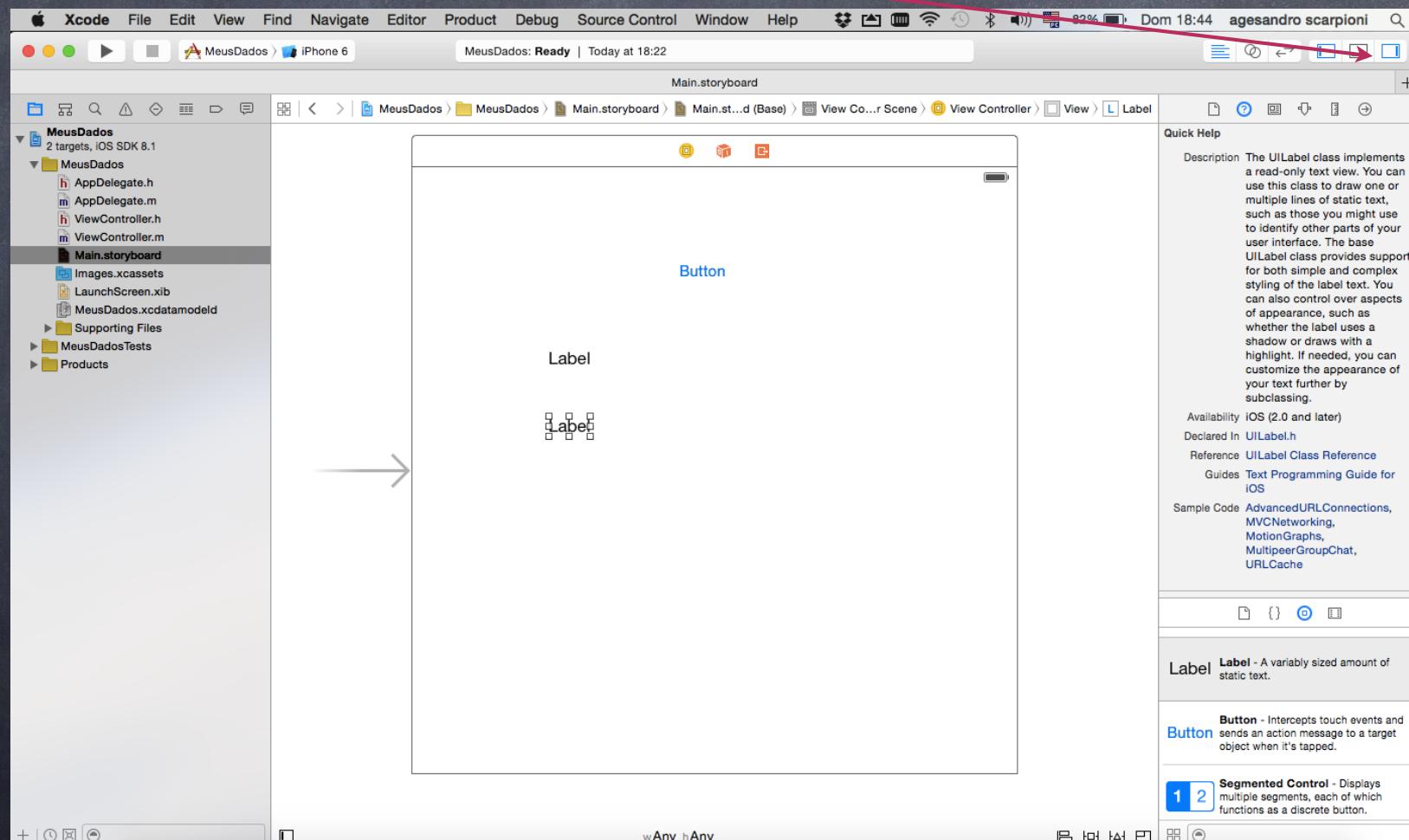
# Executando

- Verifique se seu simulador é para iPhone e execute seu App clicando em Run ou utilizando o atalho command + R.



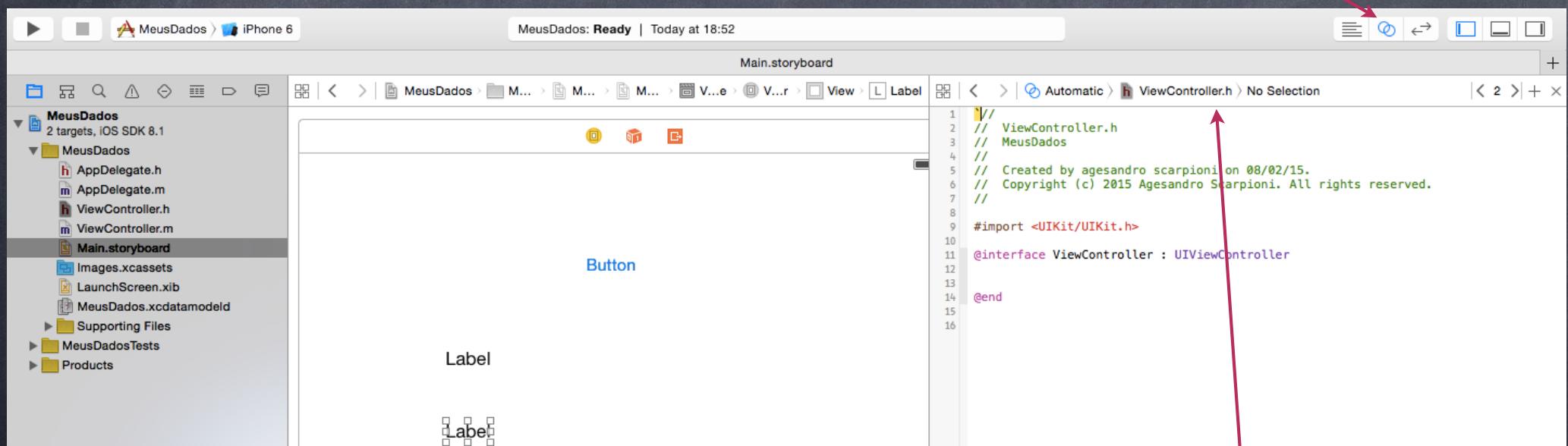
# Declarando os outlet's no .h

- Primeiramente vamos organizar nosso ambiente escondendo a janela da esquerda clicando neste ícone.



# Declarando os outlet's no .h

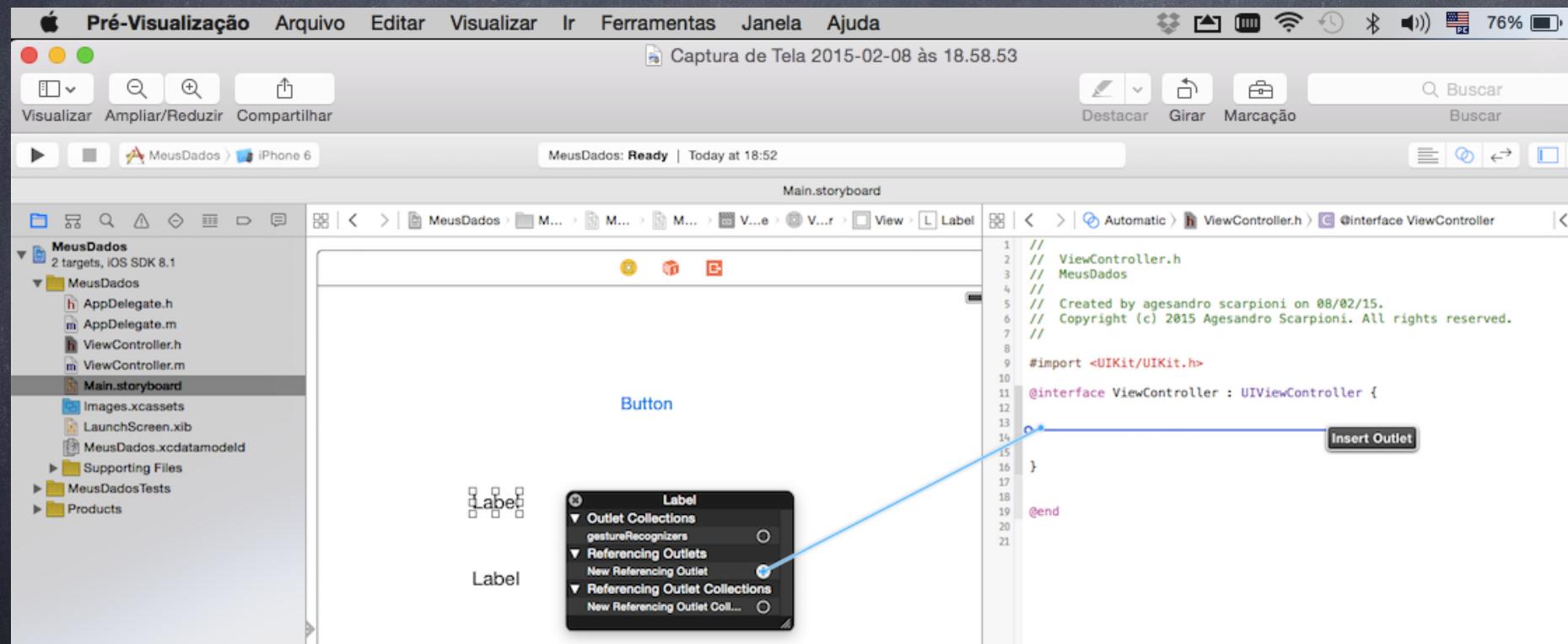
- Depois vamos compartilhar a tela de Storyboard com a viewController.h obtendo as duas simultaneamente clicando neste ícone.



Obs: Verifique se a tela que aparece é ViewController.h, caso não seja basta clicar aqui e trocar.

# Declarando os outlet's no .h automaticamente

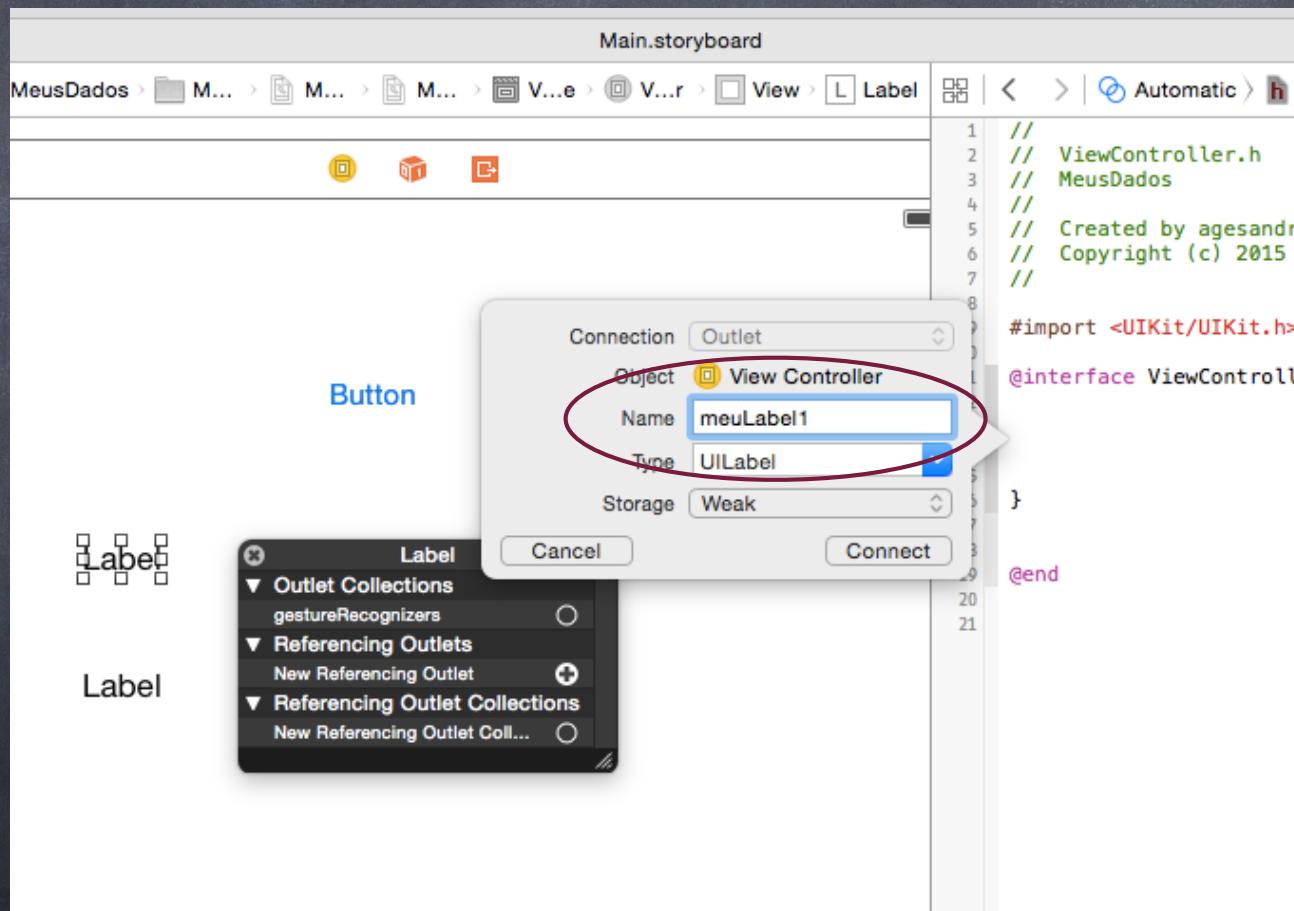
- Coloque duas chaves {} depois da declaração da interface, clique com o botão direito sobre o 1º label e escolha “New Referencing Outlet’s” clicando no local indicado na figura e arrastando até a área entre as chaves {}.



Obs: A área {} é uma área para declaração de variáveis.

# Declarando os outlet's no .h automaticamente

- Quando você soltar o botão do mouse aparecerá a janela abaixo, nomeie o outlet como "meuLabel1" e clique em connect. Repita os mesmos passos para o Label2.



# Declarando os outlet's no .h automaticamente

- Ao final essas duas linhas ( IBOutlet's ) são declaradas automaticamente no arquivo .h e os dois label's também já estão relacionados aos dois outlet's.



```
// ViewController.h
// MeusDados
//
// Created by agesandro scarpioni on 08/02/15.
// Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
//

#import <UIKit/UIKit.h>

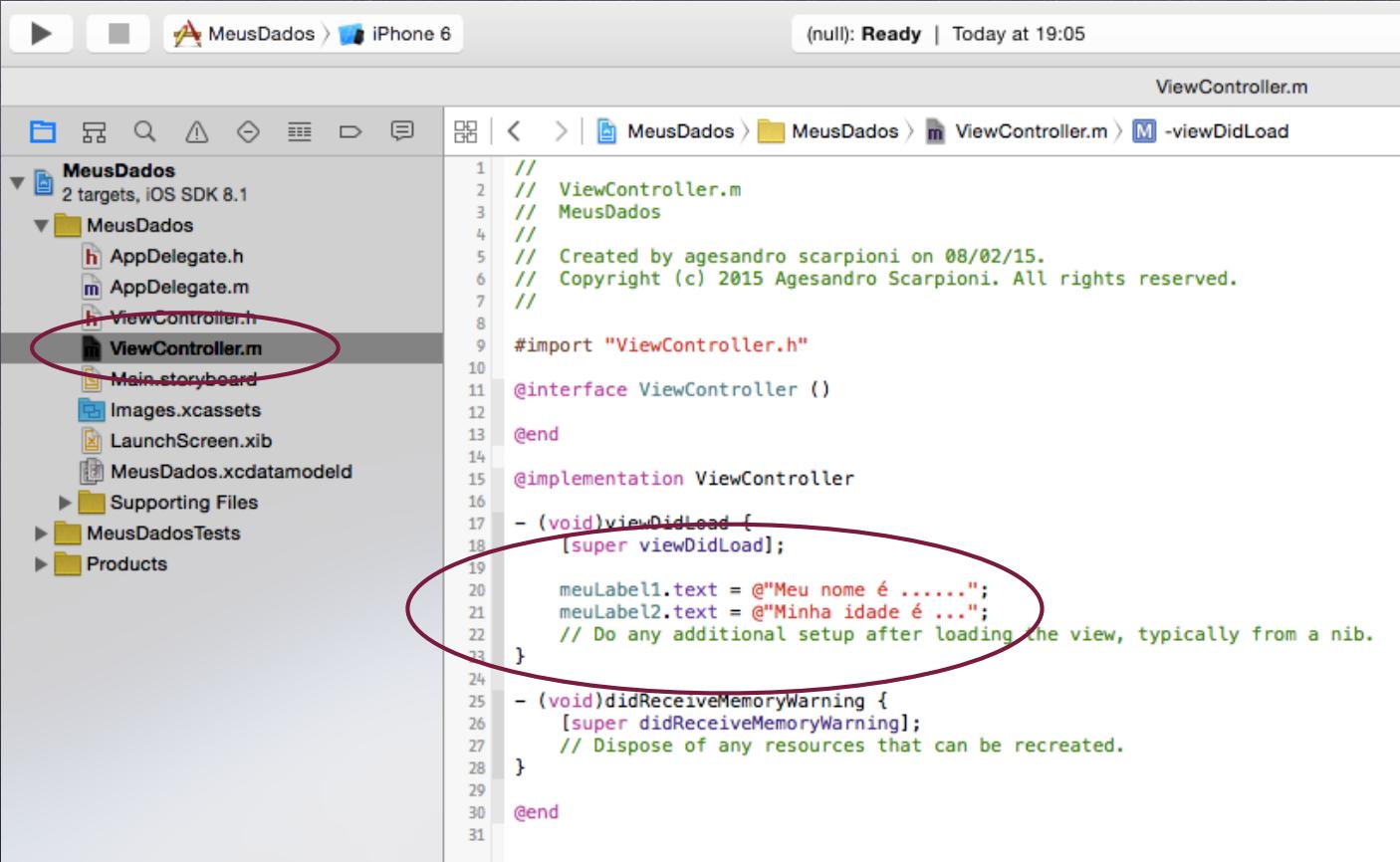
@interface ViewController : UIViewController {

    __weak IBOutlet UILabel *meuLabel1;
    __weak IBOutlet UILabel *meuLabel2;
}

@end
```

# Implementando os outlet's no .m

- Na Classe ViewController.m, existe um método chamado viewDidLoad que é o primeiro método que é executado quando a classe é chamada. Vamos carregar uma frase em cada label.



```
// ViewController.m
// MeusDados
//
// Created by agesandro scarpioni on 08/02/15.
// Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.

#import "ViewController.h"

@interface ViewController : UIViewController

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    meuLabel1.text = @"Meu nome é .....";
    meuLabel2.text = @"Minha idade é ...";
    // Do any additional setup after loading the view, typically from a nib.
}

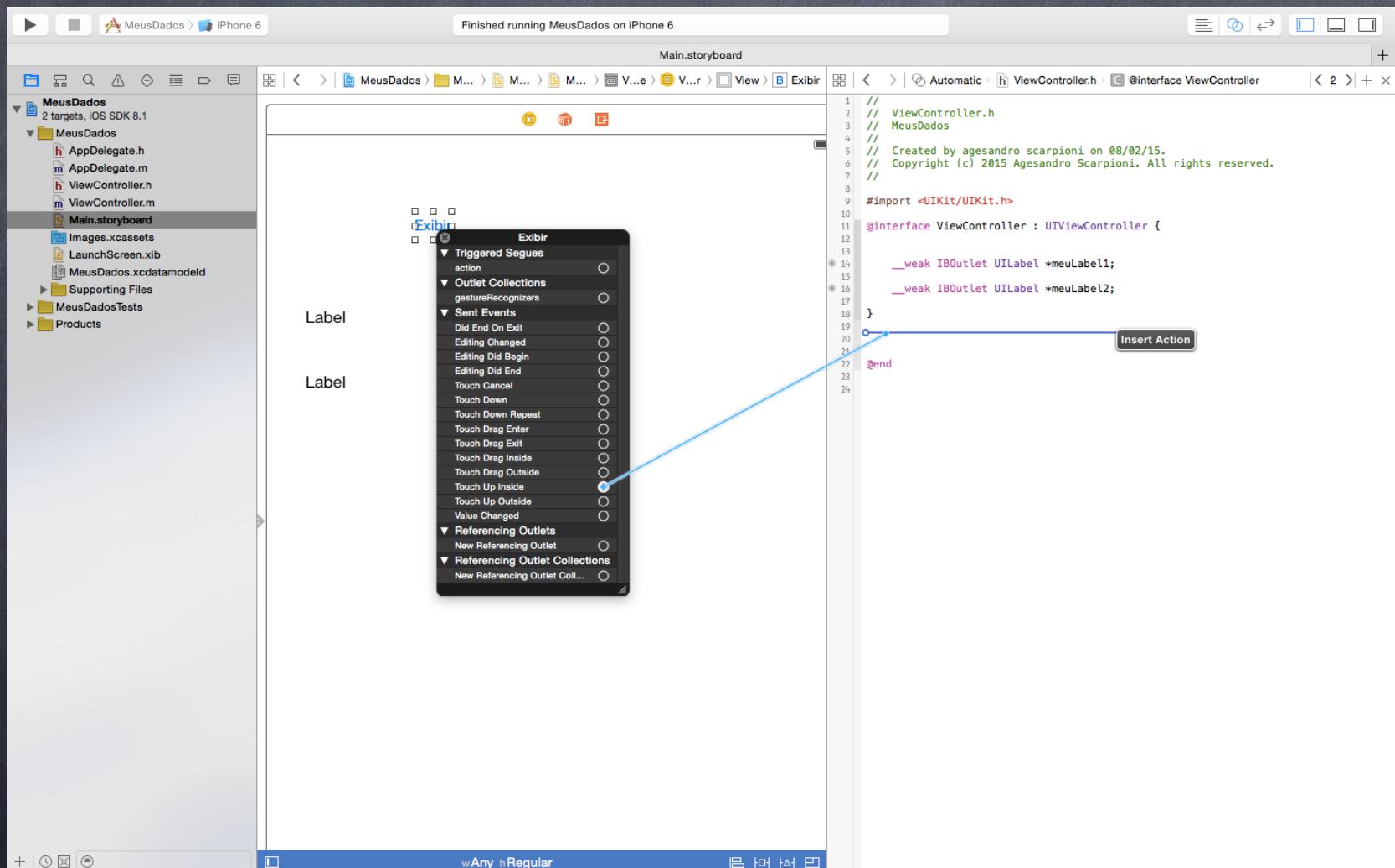
- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

Obs: viewDidLoad equivale ao Form\_Load do VB e ao windowOpened do JFrame no JAVA.

# Declarando IBAction no .h automaticamente

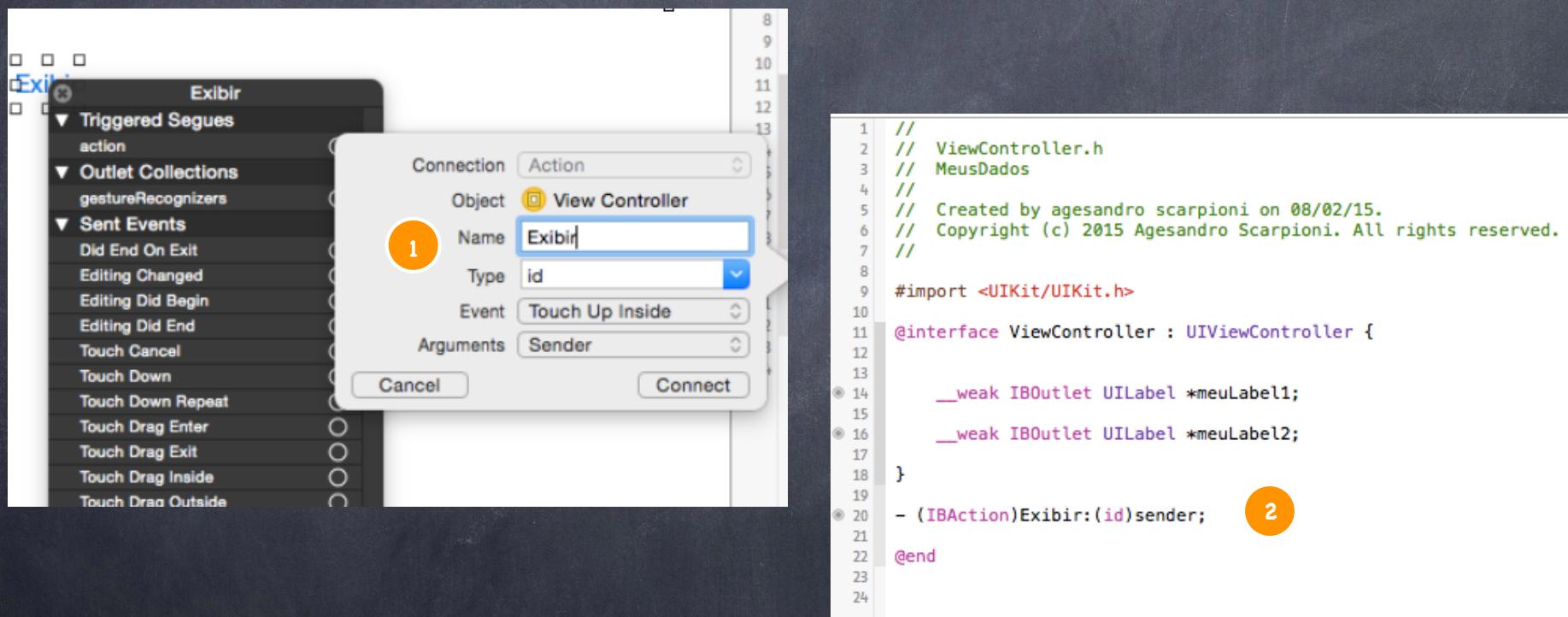
- Clique no storyboard do iPhone, clique com o botão direito do mouse sobre o Exibir, escolha o evento “Touch Up Inside” e arraste para fora da área das chaves {}.



Obs: A área fora do {} no arquivo .h, é uma área para declaração dos métodos.

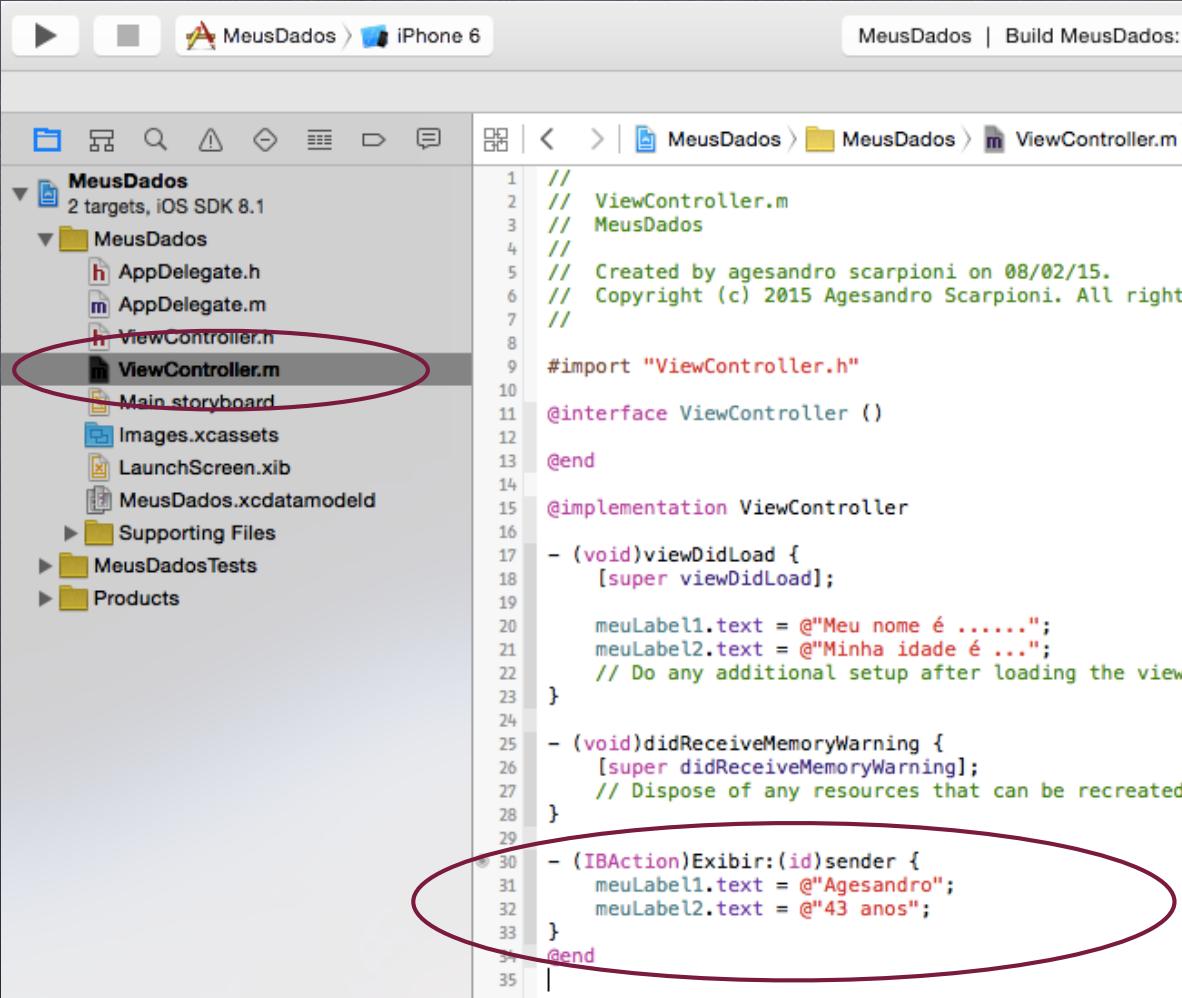
# Declarando IBAction no .h automaticamente

- Preencha o IBAction com o nome “Exibir” e clique em connect, a declaração irá aparecer automaticamente como mostra a figura 2 e já estará pronta para implementação no arquivo .m.



# Implementando IBAction .m

- Note que a área para a implementação do botão “Exibir” já aparece automaticamente, basta escrever as 2 linhas de comando.



The screenshot shows the Xcode interface with the project 'MeusDados' selected. The left sidebar shows the file structure, and the main editor shows the code for 'ViewController.m'. Two sections of the code are highlighted with red ovals:

- The first oval highlights the implementation block, starting with '@implementation ViewController' and ending with '@end'.
- The second oval highlights the implementation of the 'Exibir' IBAction method, starting with '- (IBAction)Exibir:(id)sender {'. This method sets the text of two labels: 'meuLabel1' to 'Meu nome é .....' and 'meuLabel2' to 'Minha idade é ...'. It also contains a comment about additional setup after loading the view.

```
// ViewController.m
// MeusDados
//
// Created by agesandro scarpioni on 08/02/15.
// Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.

#import "ViewController.h"

@interface ViewController : UIViewController

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    meuLabel1.text = @"Meu nome é .....";
    meuLabel2.text = @"Minha idade é ...";
    // Do any additional setup after loading the view
}

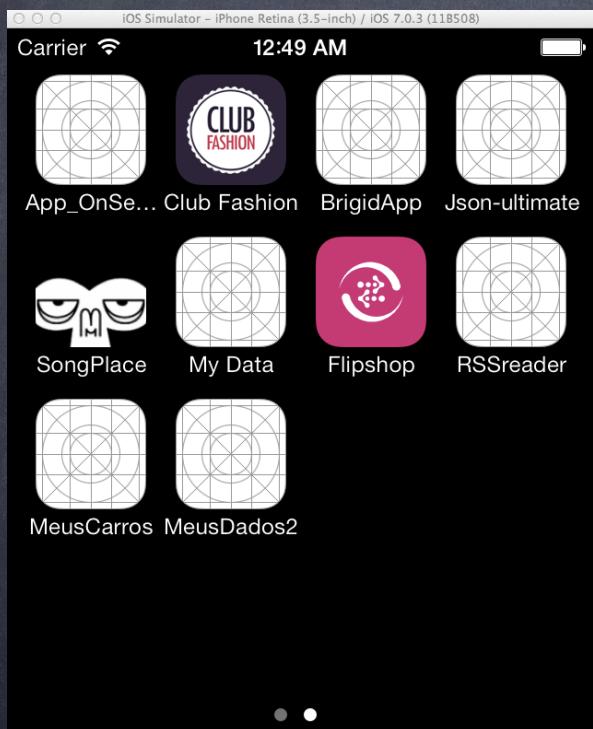
- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated
}

- (IBAction)Exibir:(id)sender {
    meuLabel1.text = @"Agesandro";
    meuLabel2.text = @"43 anos";
}
@end
```

# O App no simulador

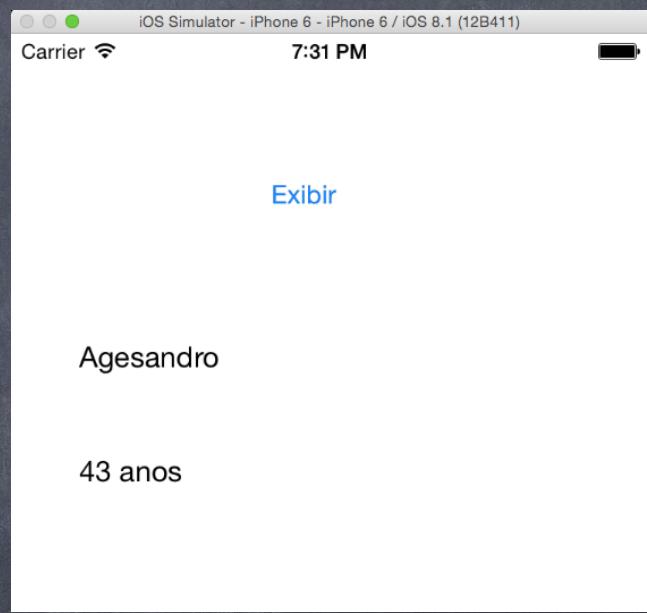
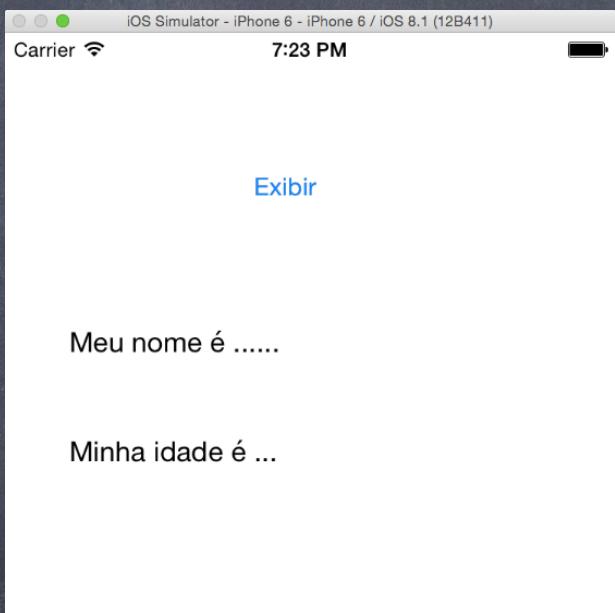
FIAP

Quando executamos



Antes de clicar no botão com  
informações carregadas pelo  
viewDidLoad

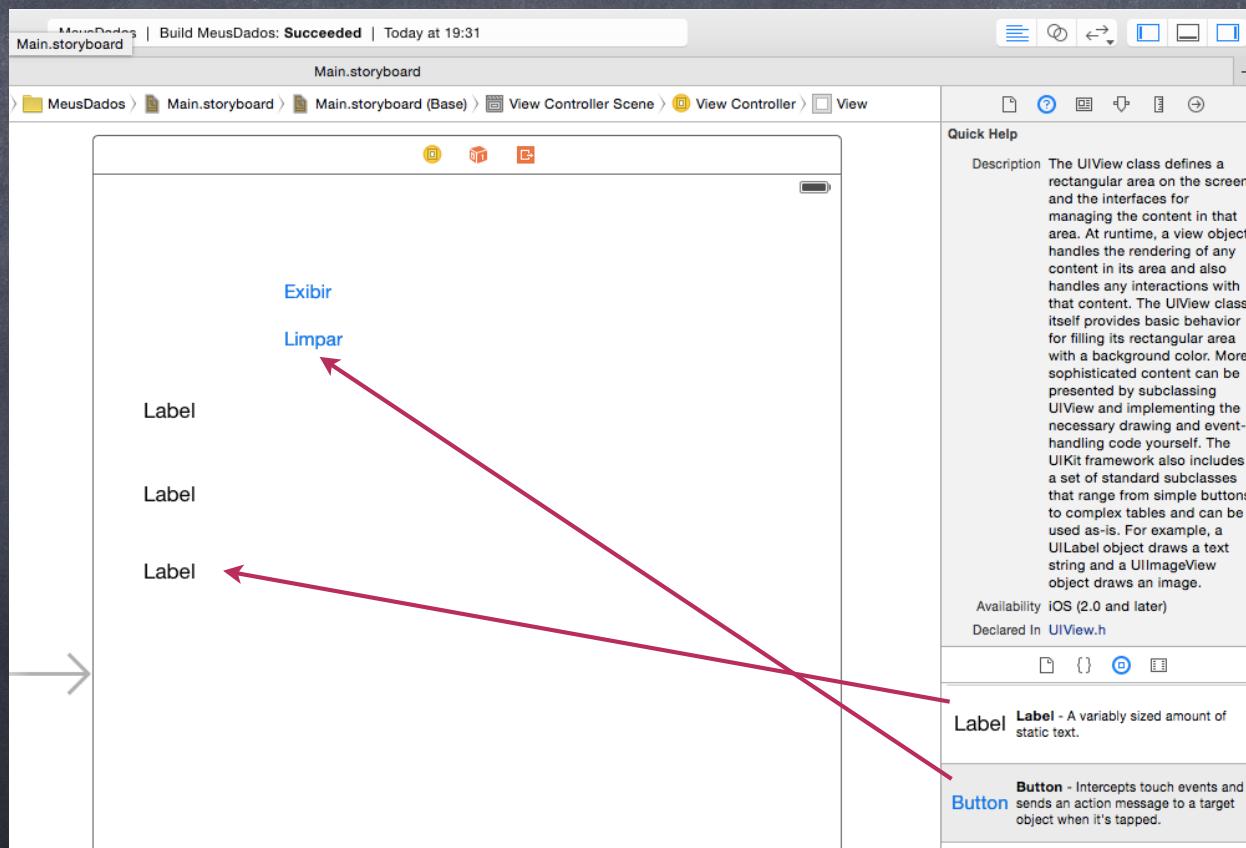
Após clicar no botão



Dica: O Command + R é o atalho para executarmos nosso App.

# Implementando IBAction e IBOulet de forma manual

- Inclua um botão Limpar e mais um label em seu Storyboard.



# Implementando IBAction e IBOulet de forma manual

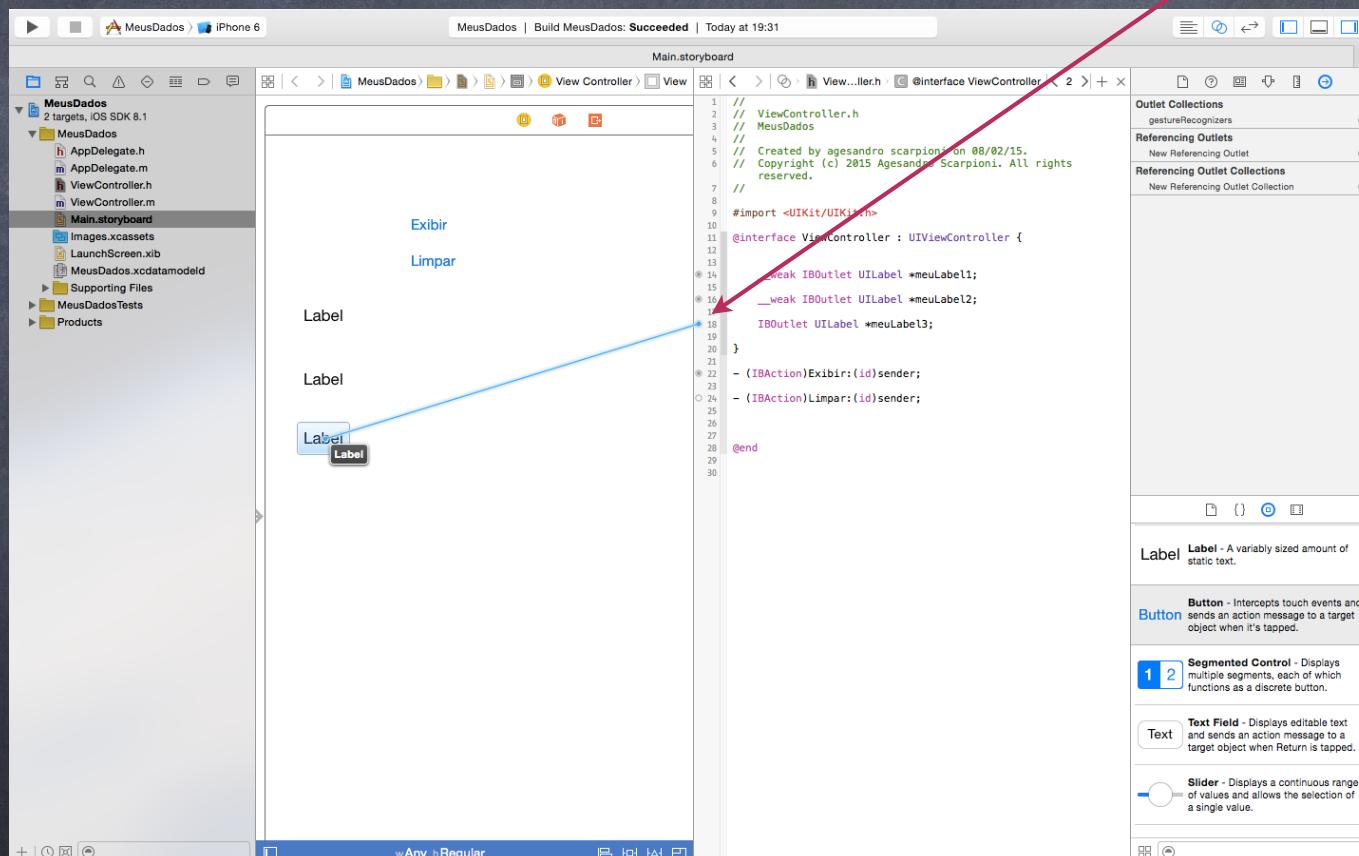
- Digite a linha do Outlet e a linha do IBAction abaixo:

• Note que os mesmos não estão conectados com a View, observe que os círculos brancos em frente aos números das linhas não estão preenchidos.

```
1 // ViewController.h
2 // MeusDados
3 //
4 // Created by agesandro scarpioni on 08/02/15.
5 // Copyright (c) 2015 Agesandro Scarpioni. All rights
6 // reserved.
7 //
8 #import <UIKit/UIKit.h>
9
10 @interface ViewController : UIViewController {
11
12
13
14     __weak IBOutlet UILabel *meuLabel1;
15
16     __weak IBOutlet UILabel *meuLabel2;
17
18     IBOutlet UILabel *meuLabel3;
19 }
20
21
22 - (IBAction)Exibir:(id)sender;
23 - (IBAction)Limpar:(id)sender;
24
25
26
27
28 @end
29
30
```

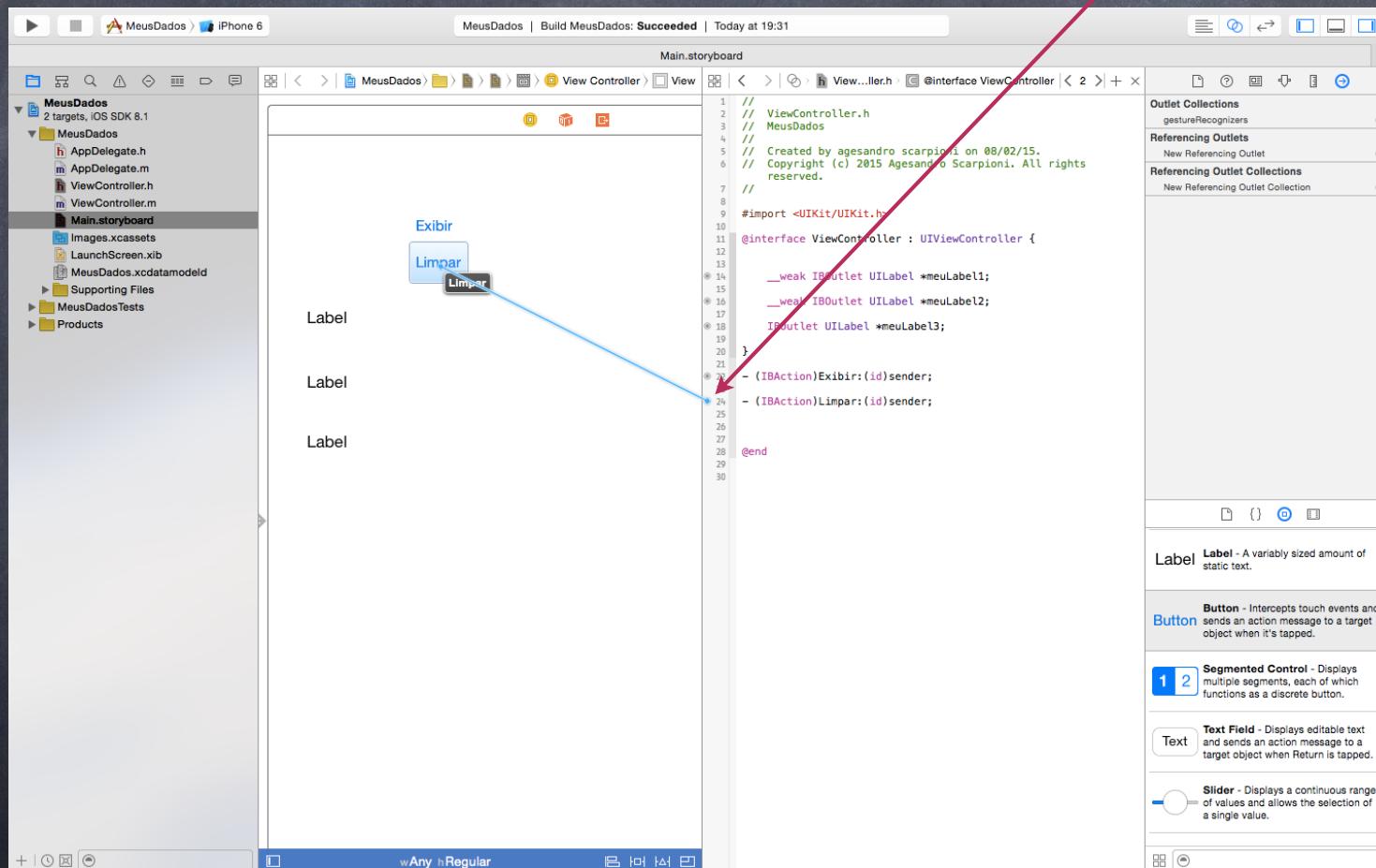
# Implementando IBAction e IBOulet de forma manual

- Trace uma linha do ponto indicado do Outlet até o Label para criarmos um link entre o Label no controller ao Label da View.



# Implementando IBAction e IBOutlet de forma manual

- Faça o mesmo com o IBAction Limpar.



# Implementando IBAction e IBOulet de forma manual

- Note que após essas ligações nosso novo IBOulet e IBAction estão conectados com a View.

```
1 //  
2 // ViewController.h  
3 // MeusDados  
4 //  
5 // Created by agesandro scarpioni on 08/02/15.  
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights  
7 // reserved.  
8 //  
9 #import <UIKit/UIKit.h>  
10  
11 @interface ViewController : UIViewController {  
12  
13  
14     __weak IBOutlet UILabel *meuLabel1;  
15  
16     __weak IBOutlet UILabel *meuLabel2;  
17  
18     IBOutlet UILabel *meuLabel3;  
19  
20 }  
21  
22 - (IBAction)Exibir:(id)sender;  
23  
24 - (IBAction)Limpar:(id)sender;  
25  
26  
27  
28 @end  
29  
30
```

# Implementando IBAction e IBOulet de forma manual

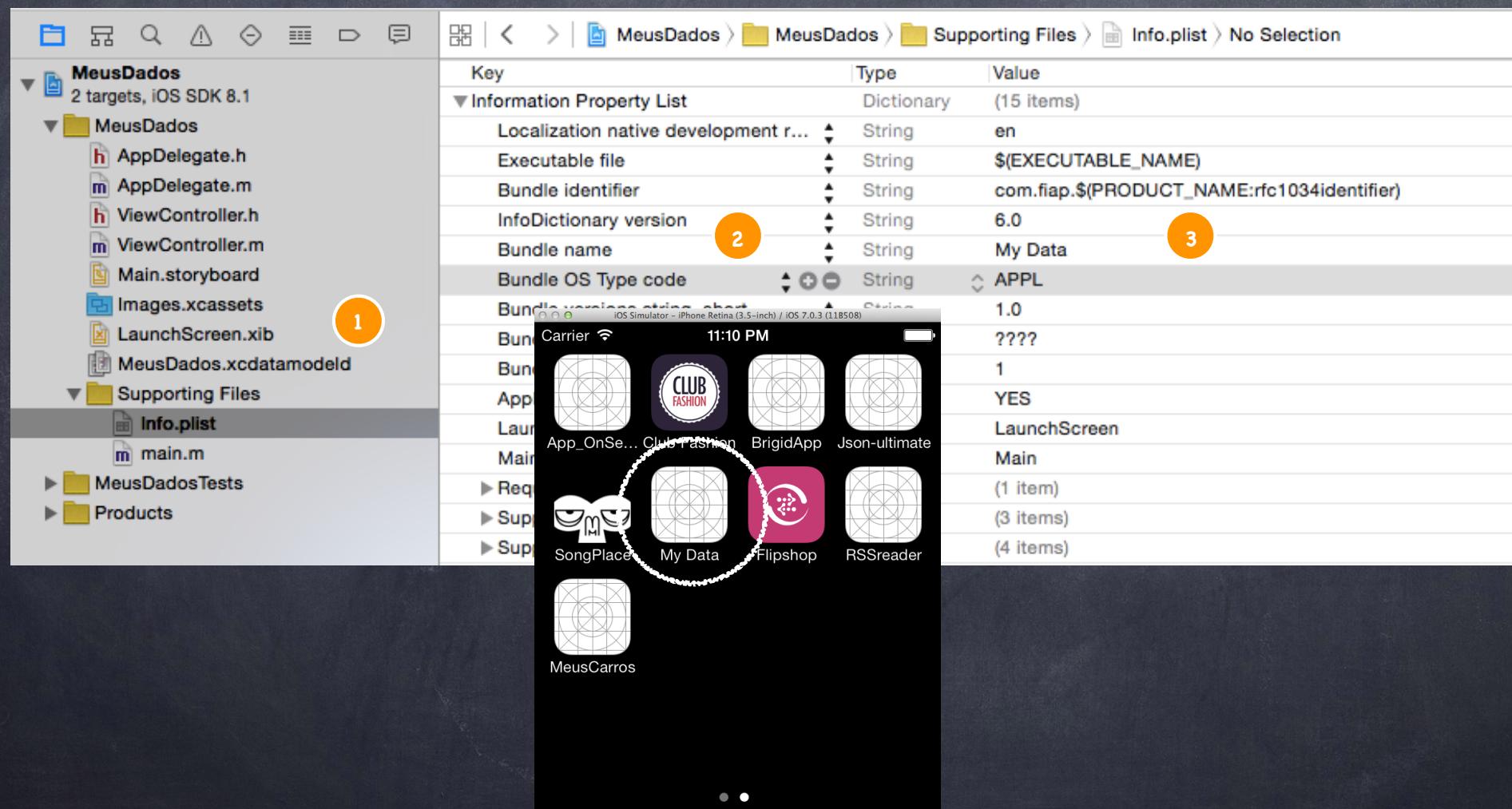
- Inclua as linhas do Label3 no DidLoad e no Exibir.

- Copie a linha do IBAction Limpar no arquivo ViewController.h e cole no ViewController.m, retire o (;) e coloque as chaves ({}), após essa etapa copie as linhas do DidLoad e cole no IBAction Limpar.

```
1 //ViewController.m
2 // MeusDados
3 //
4 // Created by agesandro scarpioni on 08/02/15.
5 // Copyright (c) 2015 Agesandro Scarpioni. All rights
6 // reserved.
7 //
8
9 #import "ViewController.h"
10 @interface ViewController : UIViewController
11 @end
12 @implementation ViewController
13
14 - (void)viewDidLoad {
15     [super viewDidLoad];
16
17     meuLabel1.text = @"Meu nome é .....";
18     meuLabel2.text = @"Minha idade é ...";
19     meuLabel3.text = @"Minha cidade é ..";
20     // Do any additional setup after loading the view,
21     // typically from a nib.
22 }
23
24
25 - (void)didReceiveMemoryWarning {
26     [super didReceiveMemoryWarning];
27     // Dispose of any resources that can be recreated.
28 }
29
30
31 - (IBAction)Exibir:(id)sender {
32     meuLabel1.text = @"Agesandro";
33     meuLabel2.text = @"43 anos";
34     meuLabel3.text = @"São Paulo";
35 }
36
37
38 - (IBAction)Limpar:(id)sender{
39     meuLabel1.text = @"Meu nome é .....";
40     meuLabel2.text = @"Minha idade é ...";
41     meuLabel3.text = @"Minha cidade é ..";
42 }
43
44
45
46
47
```

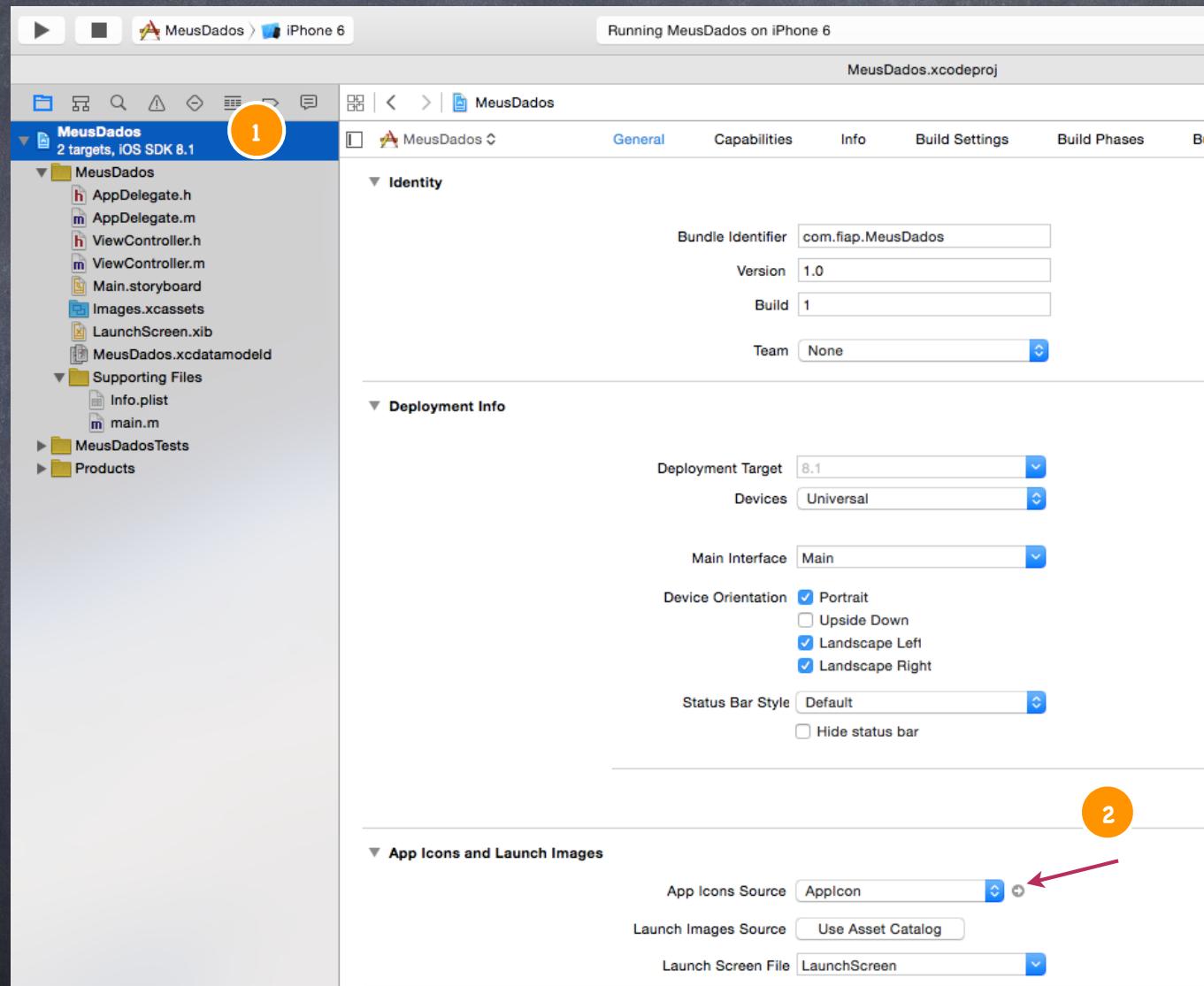
# Como trocar o nome do App

- Abra o Supporting Files(1) e altere o arquivo MeusDados-info.plist adicione o campo Bundle display name (2), digite My Data(3), veja o próximo slide.



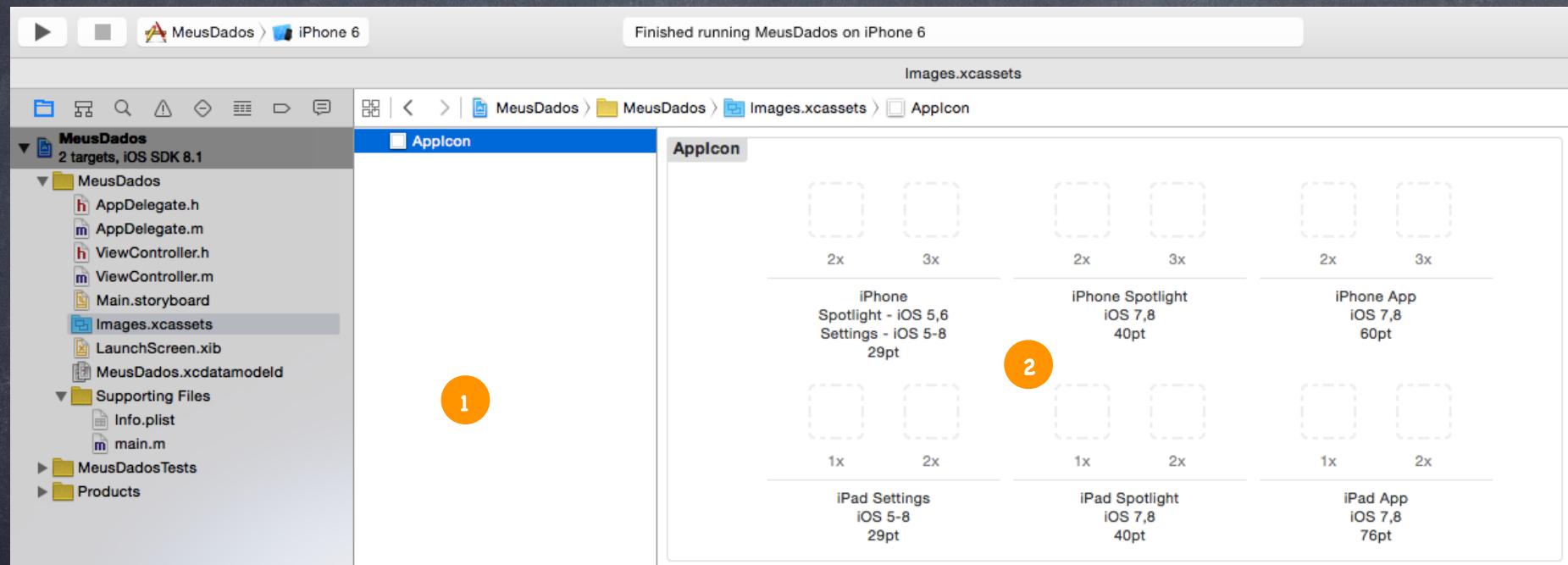
# Alterando o Ícone do App

- Clique em seu projeto (1), acesse o passo 2



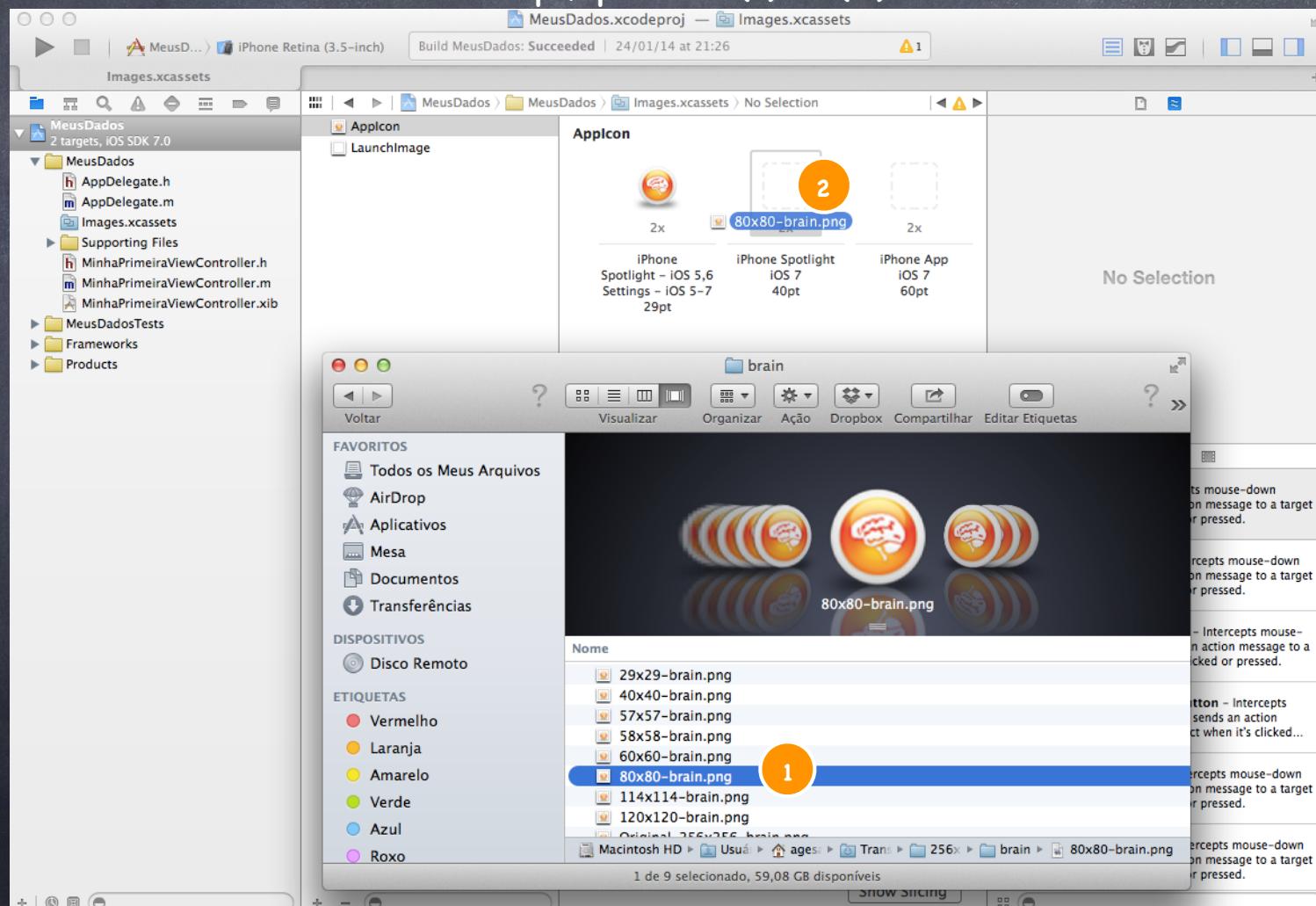
# Alterando o Ícone do App

- A tela abaixo será exibida. O ítem (1) exibe o catálogo de imagens, o ítem (2) exibe os padrões de imagens e suas respectivas visualizações.



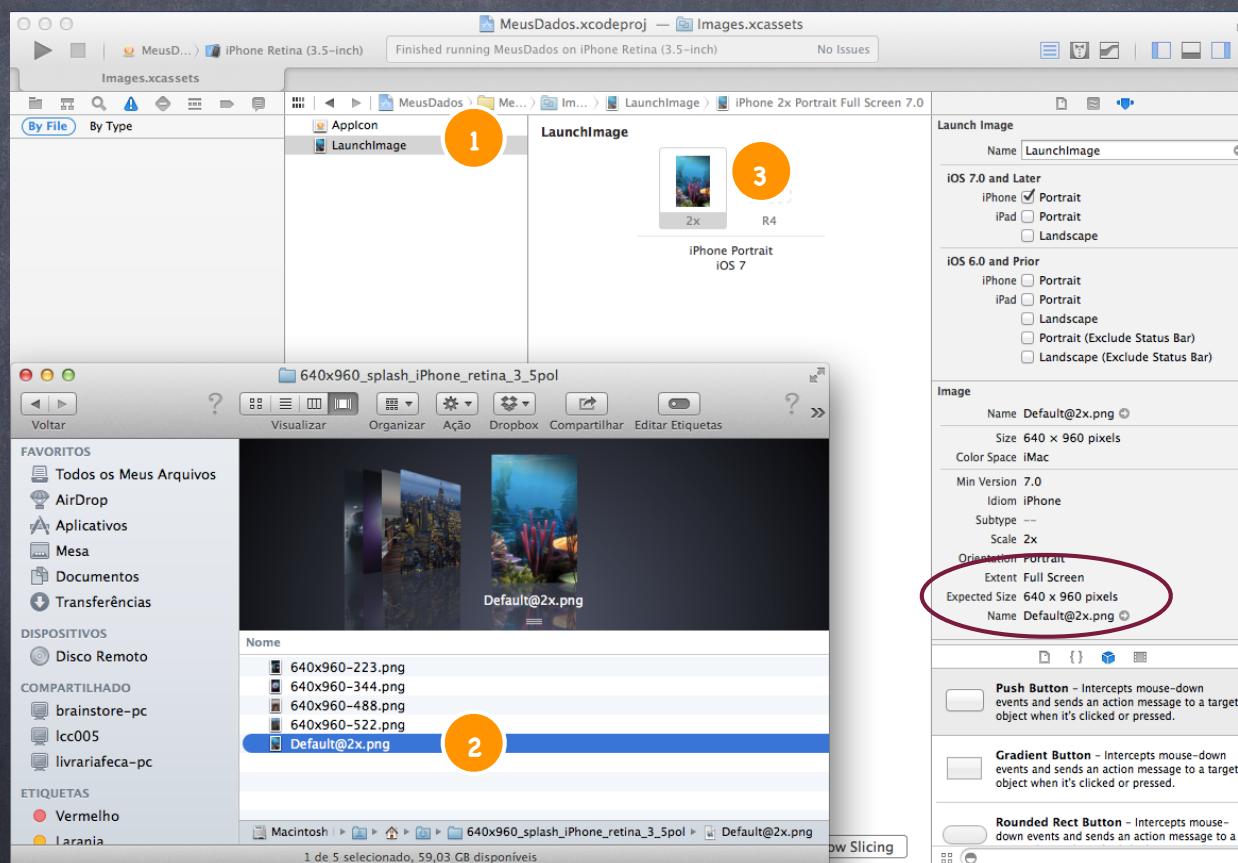
# Alterando o Ícone do App

- Abra uma das pastas disponibilizadas pelo professor com o finder e arraste a imagem apropriada para o espaço exclusivo da imagem escolhida, exemplo: A imagem 80x80 entra no 2x - 40pt, passos (1) e (2).



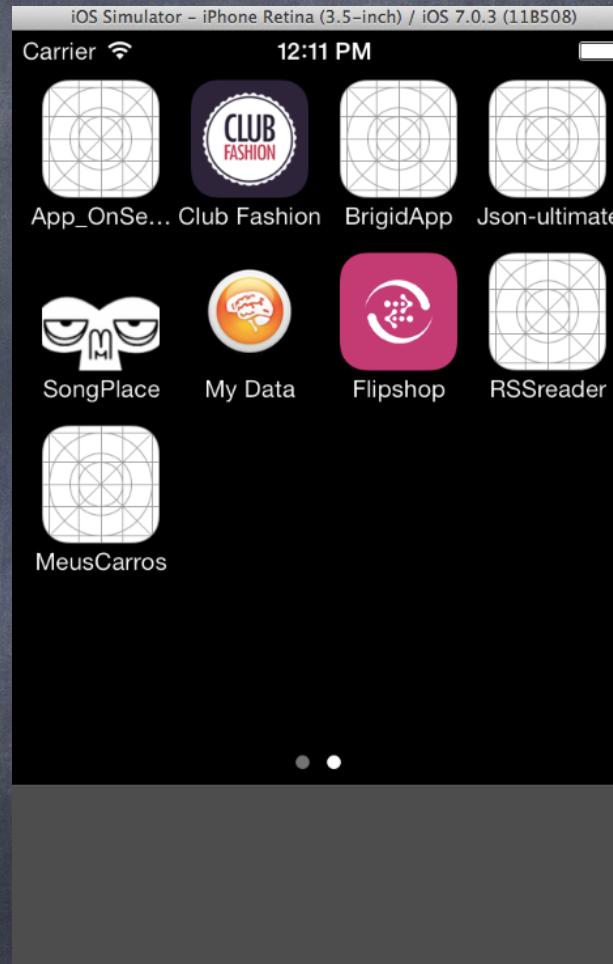
# Imagen de splash screen

- Existe um padrão para os nomes das imagens para todos os tipos de devices da Apple (iPhone, iPad, iPhone com tela de retina, etc.), veja a tabela dos padrões de nomes por dispositivo nos próximos slides tanto para splash quanto para ícones, se você tiver uma pasta com estes nomes em formato padrão elas são úteis para iOS anterior ao 7.



# Imagens

- Execute seu programa e verifique as imagens adicionadas.



Dica: Sites que ajudam a manipular imagens para Mobile: a) <http://ticons.fokkezb.nl> b) <http://makeappicon.com>

# Tabela de padrões de nomes de imagem

- A tabela abaixo é para imagens splash screen.

Default.png	320x480 - iPhone, iPod
Default@2x.png	640x960 - iPhone retina 3,5 polegadas
Default-568h@2x.png	640x1136 - iPhone retina 4 polegadas
Default-Landscape~ipad.png	1024x768 - iPad
Default-Portrait~ipad.png	768x1024 - iPad
Default-Landscape@2x~ipad.png	2048x1536 - iPad retina
Default-Portrait@2x~ipad.png	1536x2048 - iPad retina

# Tabela de padrões de nomes de imagem para ícones

## • Dimensões dos ícones para iOS 6.1 ou anterior

icon.png	57x57 ícone principal iPhone, iPod Touch
icon@2x.png	114x114 - ícone principal iPhone, iPod Touch (retina)
icon-72.png	72x72 - ícone principal iPad
icon-72@2x.png	144x144 - ícone principal iPad (retina)
icon-Small.png	29x29 iPhone - ícones para resultados de pesquisa e configurações do app
icon-Small@2x.png	58x58 iPhone - ícones para resultados de pesquisa e configurações do app (retina)
icon-Small-50.png	50x50 iPad - ícones para resultados de pesquisa e configurações do app
icon-Small-50@2x.png	100x100 iPad - ícones para resultados de pesquisa e configurações do app (retina)

# Tabela de padrões de ícones e resoluções

## • Dimensões dos ícones para iOS 7 ou posterior

60x60 - ícone principal iPhone, iPod Touch
120x120 - ícone principal iPhone, iPod Touch (retina)
76x76 - ícone principal iPad
152x152 - ícone principal iPad (retina)
40x40 - Todos os dispositivos - ícones para resultados de pesquisa
80x80 - Todos os dispositivos - ícones para resultados de pesquisa (retina)
29x29 - Todos os dispositivos - ícones para configurações do app
58x58 - Todos os dispositivos - ícones para configurações do app (retina)

**Obs:** O uso de nomes de arquivos fixos para os seus ícones de aplicativos é somente para compatibilidade com versões anteriores do iOS.

# Utilizando break points

- Ao clicar nessa área é inserido um break point

The screenshot shows the Xcode interface with the following details:

- Top Bar:** Shows "MeusDados > iPhone 6.1 Simulator" and "Running MeusDados on iPhone 6.1 Simulator".
- Scheme:** Set to "Breakpoints".
- Project:** "MeusDados" with 1 file open.
- Editor:** Shows the code for `MinhaPrimeiraViewController.m`. A red arrow points to the line `- (IBAction)Exibir:(id)sender{`, indicating where a breakpoint was set.
- Threads:** Thread 1 (Paused) is selected, showing the stack trace: `0 -[MinhaPrimeiraViewController ...] 1-[NSObject performSelector:withObjec...`.
- Output Window:** Shows the "All Output" tab with the following content:
  - Thread 1: step over
  - 2013-05-30 03:52:57.402 MeusDados[2142:c07] Atenção
  - (lldb)
- Bottom Bar:** Shows the "Step Over" button and other debugger controls.

Dica: Utilize o botão Step Over para avançar por entre as linhas após o break point, use a janela All Output para ver as informações printadas com NSLog.

# Prática 1

Criação de um programa para testarmos todos os conceitos deste tópico.

- Crie um projeto novo com 4 labels e 2 botões, faça aparecer nestes label's as seguintes informações:
- Botão 1: seu nome completo, sua cidade de nascimento, respectivamente nos 2 primeiros label's
- Botão 2: seu email e a data de nascimento nos outros dois label's.
- Altere o nome e o ícone do aplicativo, altere a imagem de splash screen.

# Próxima aula

- ➊ Novos objetos em novas interfaces.