

App Nome e Idade

X-Code com Swift

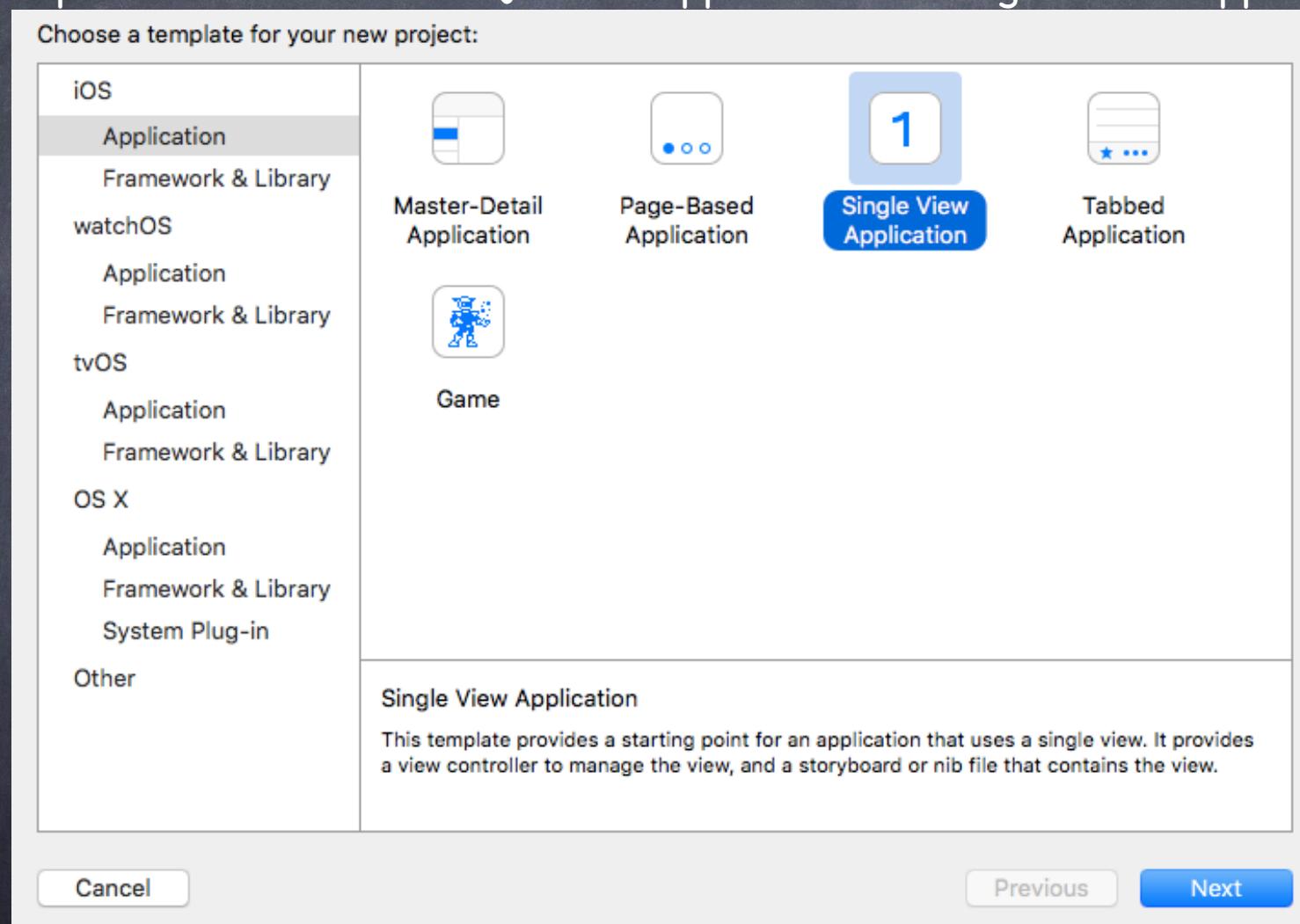
Prof. Agesandro Scarpioni
agesandro@fiap.com.br

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

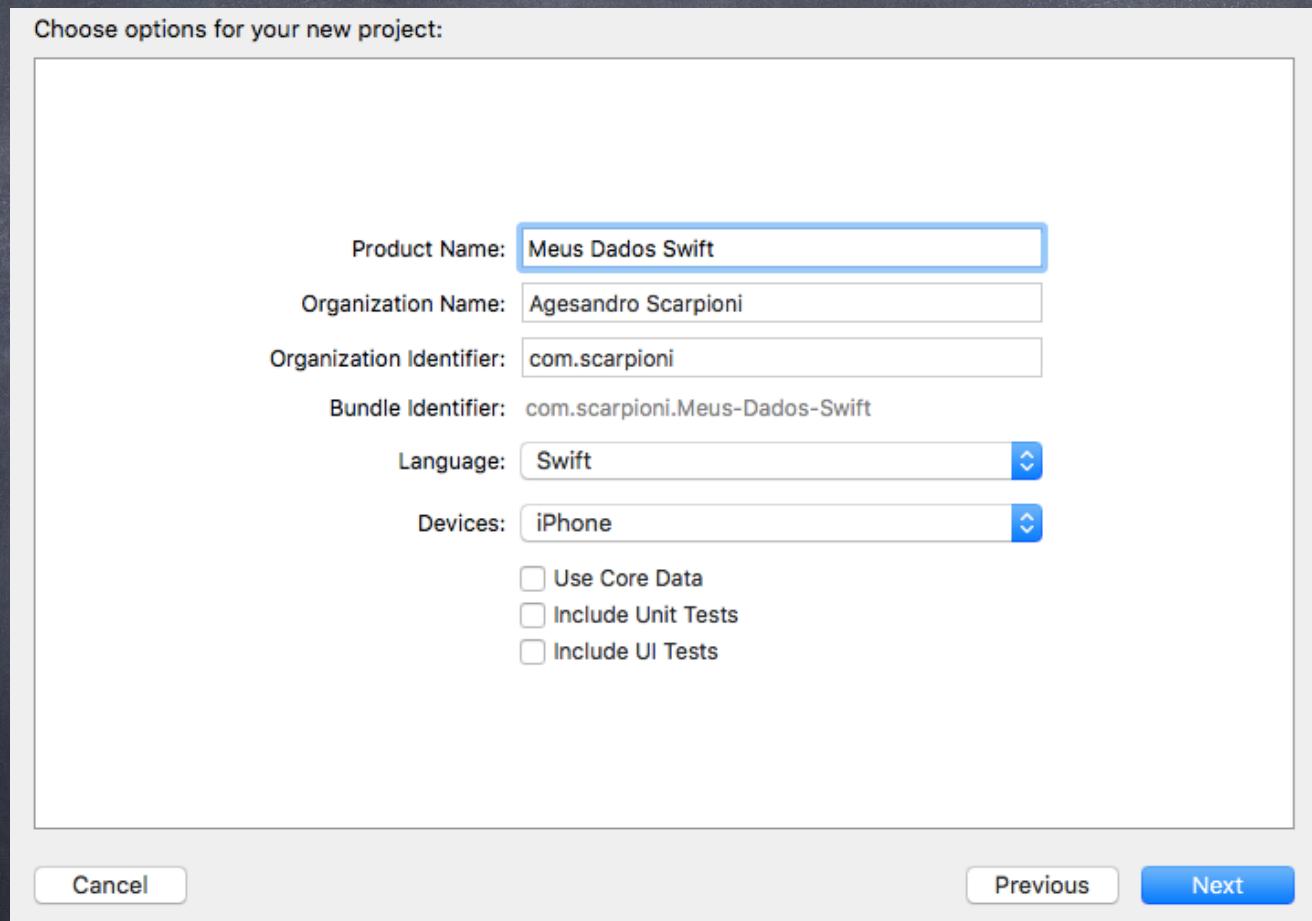
- Clique em File -> New Project -> Application -> Single View Application.



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

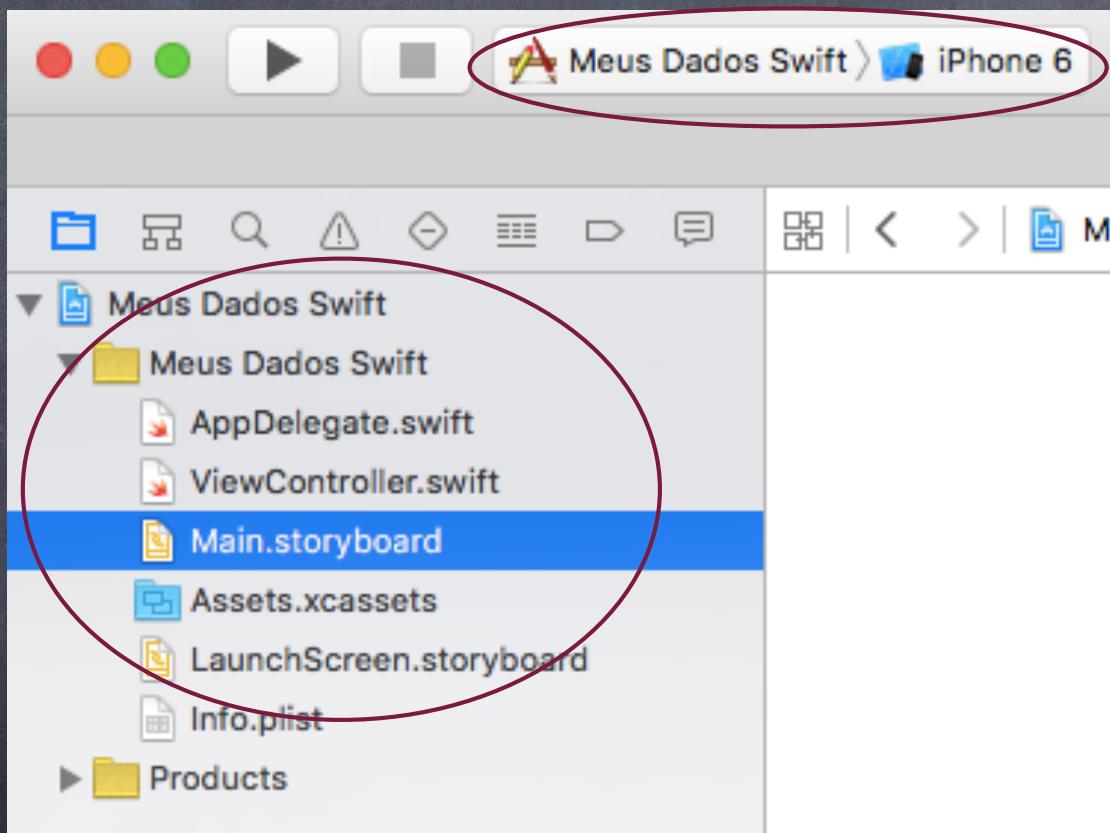
- Preencha com os dados abaixo, lembre-se que o Company Identifier é como se fosse o pacote no Java ou o namespace do VB, marque o 1º e 2º check box.



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

Tudo em seu lugar

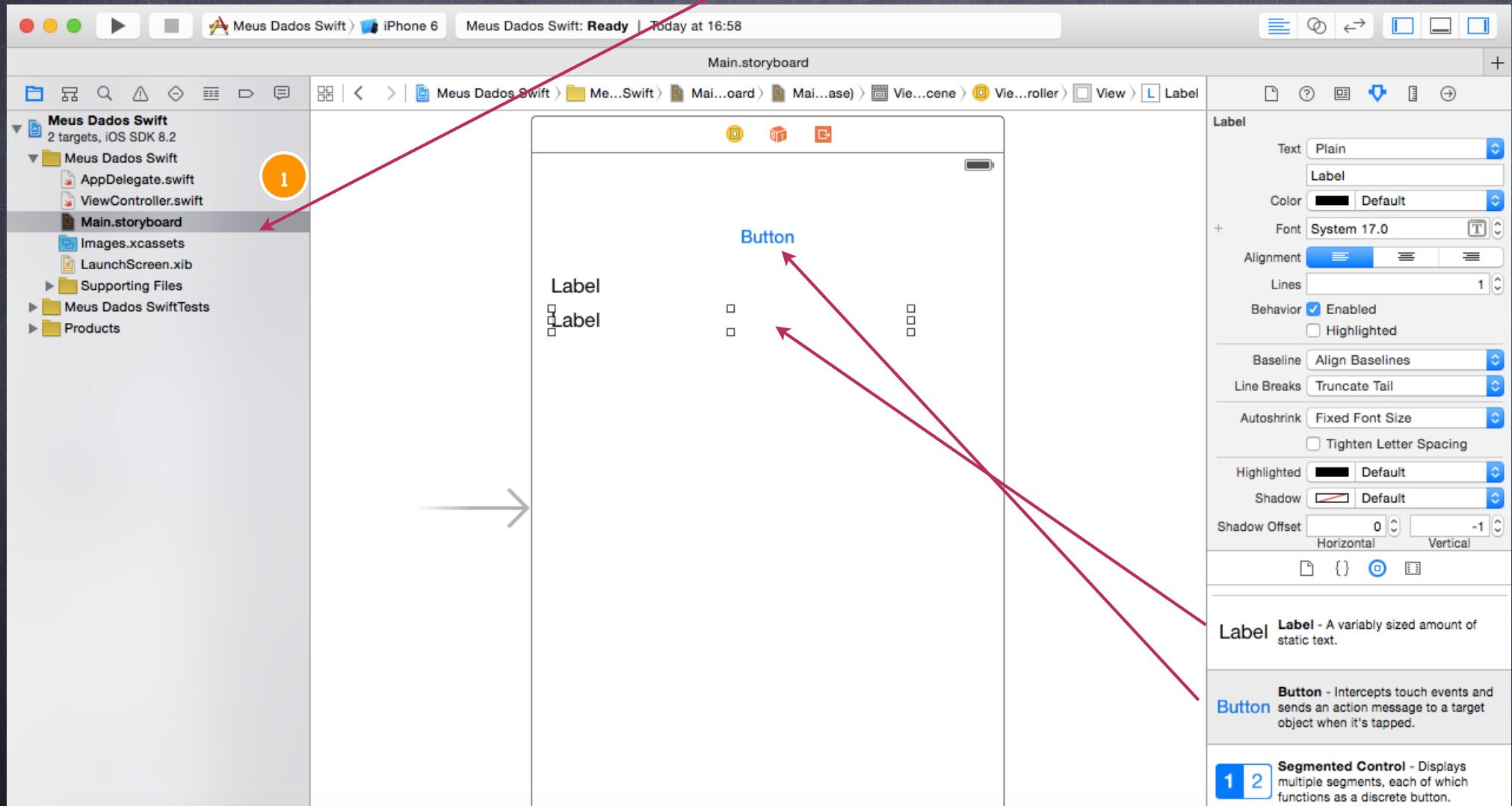
- Note que utilizando este template foi incluída uma classe chamada ViewController.swift, o delegate que já possui a classe referenciada e o Storyboard (iPhone e iPad).



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.

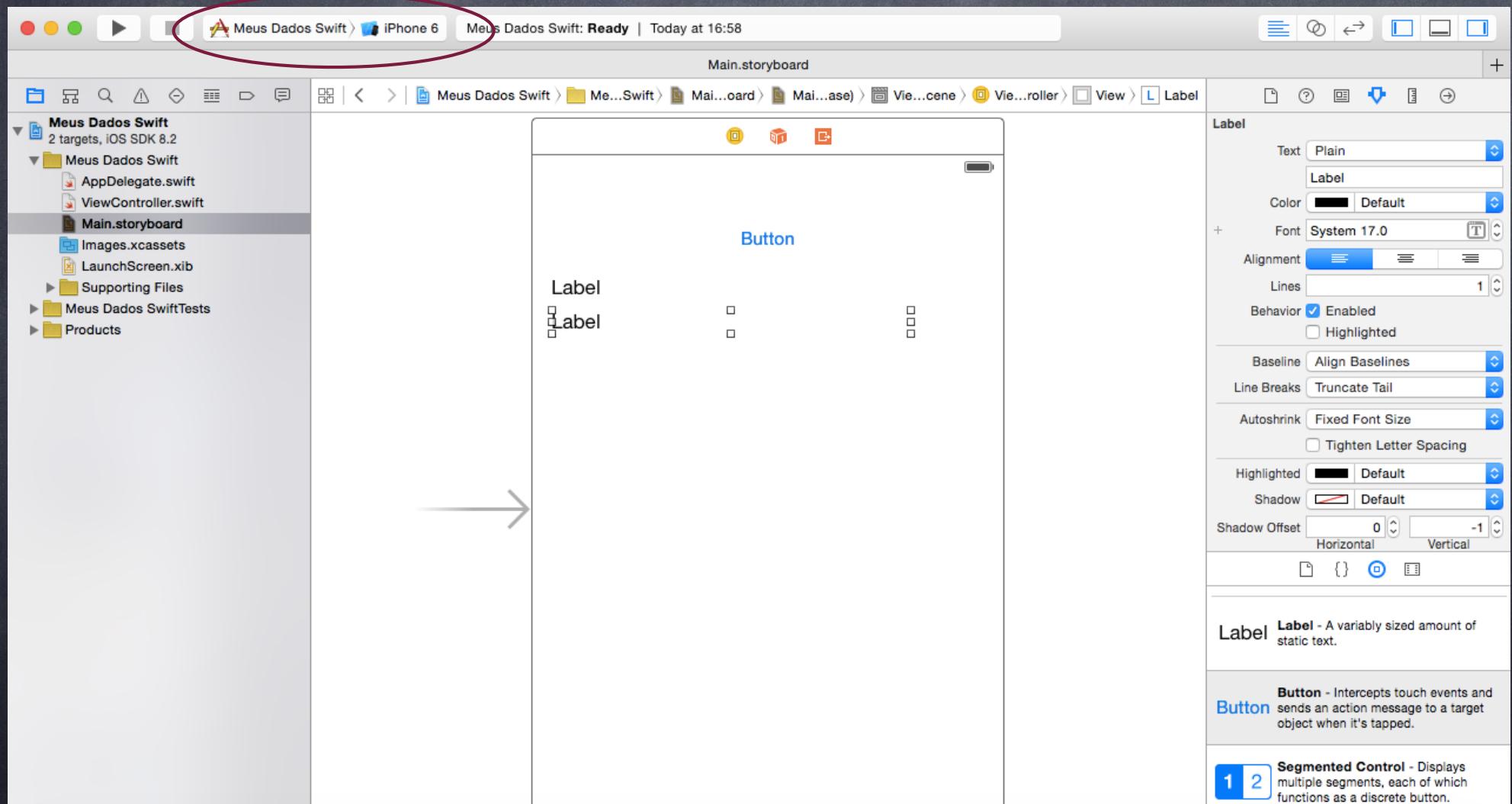
Adicionando objetos

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



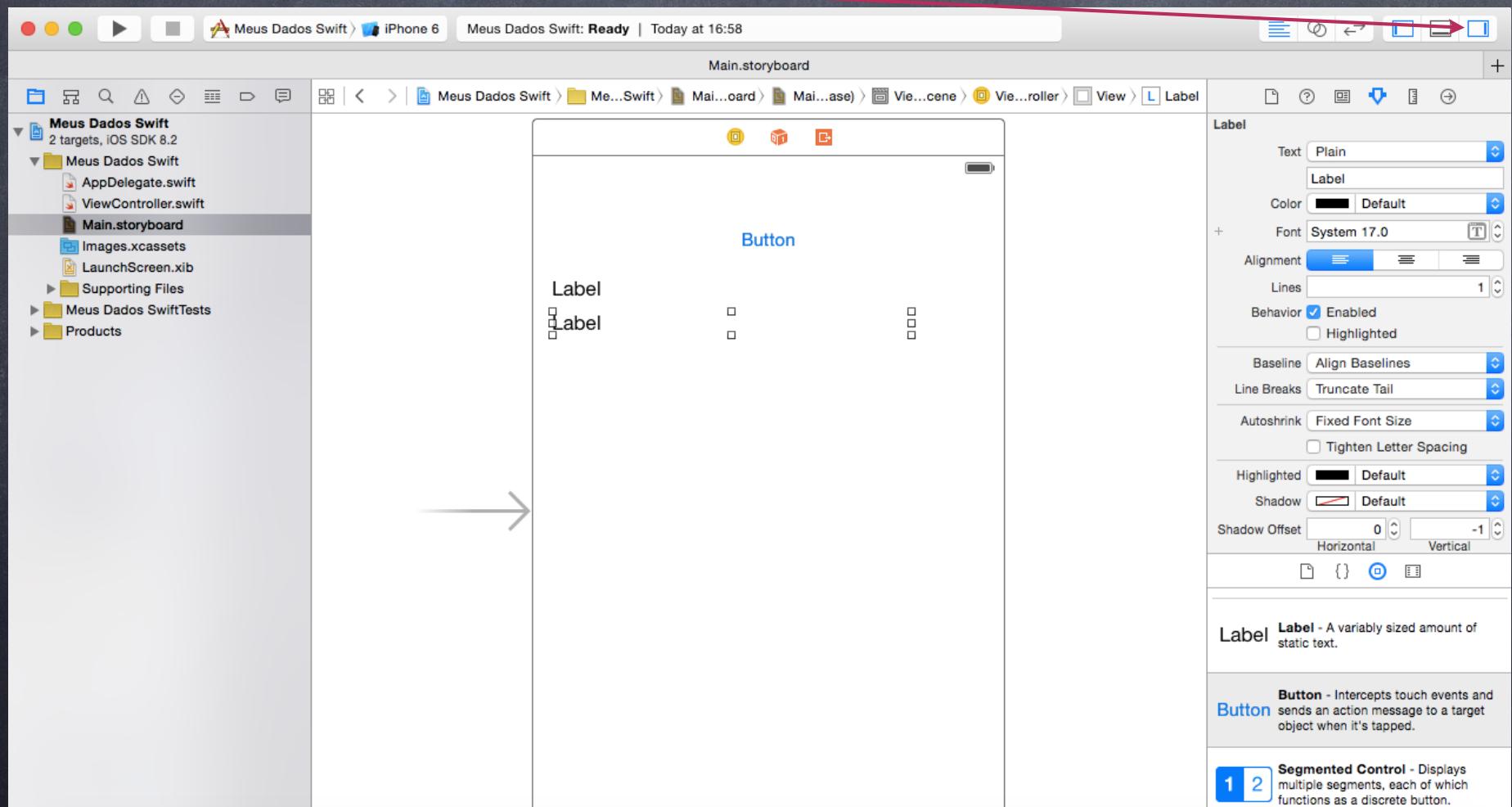
Executando

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



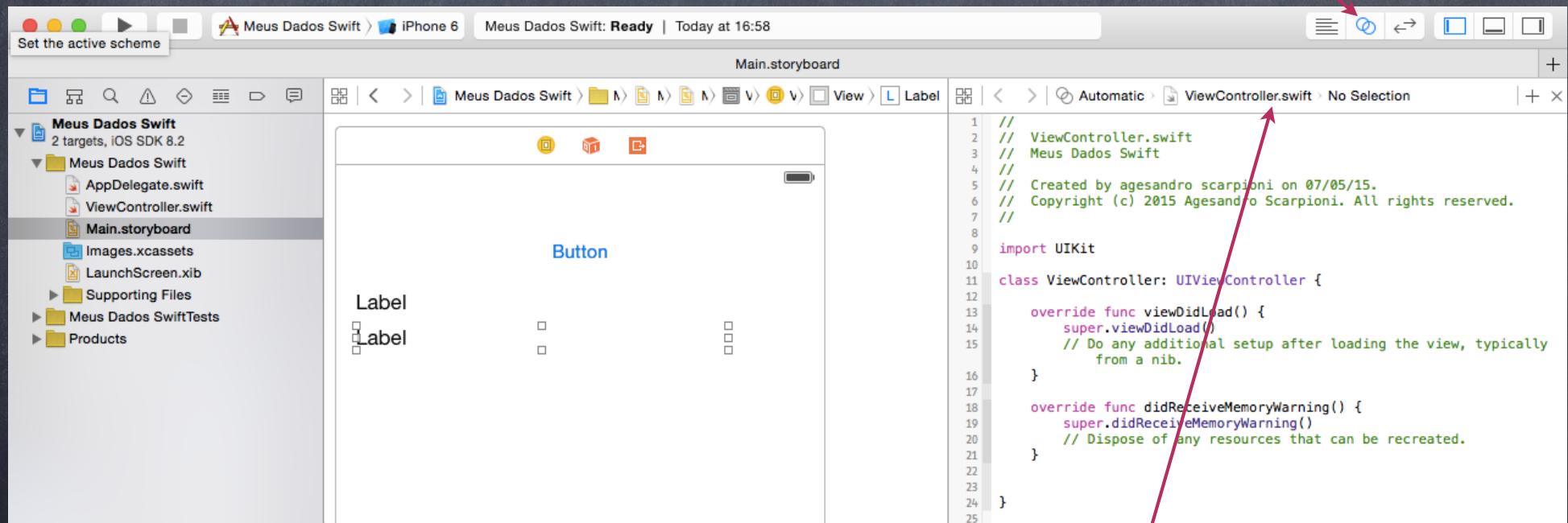
Declarando os outlet's no .h

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



Declarando os outlet's no .h

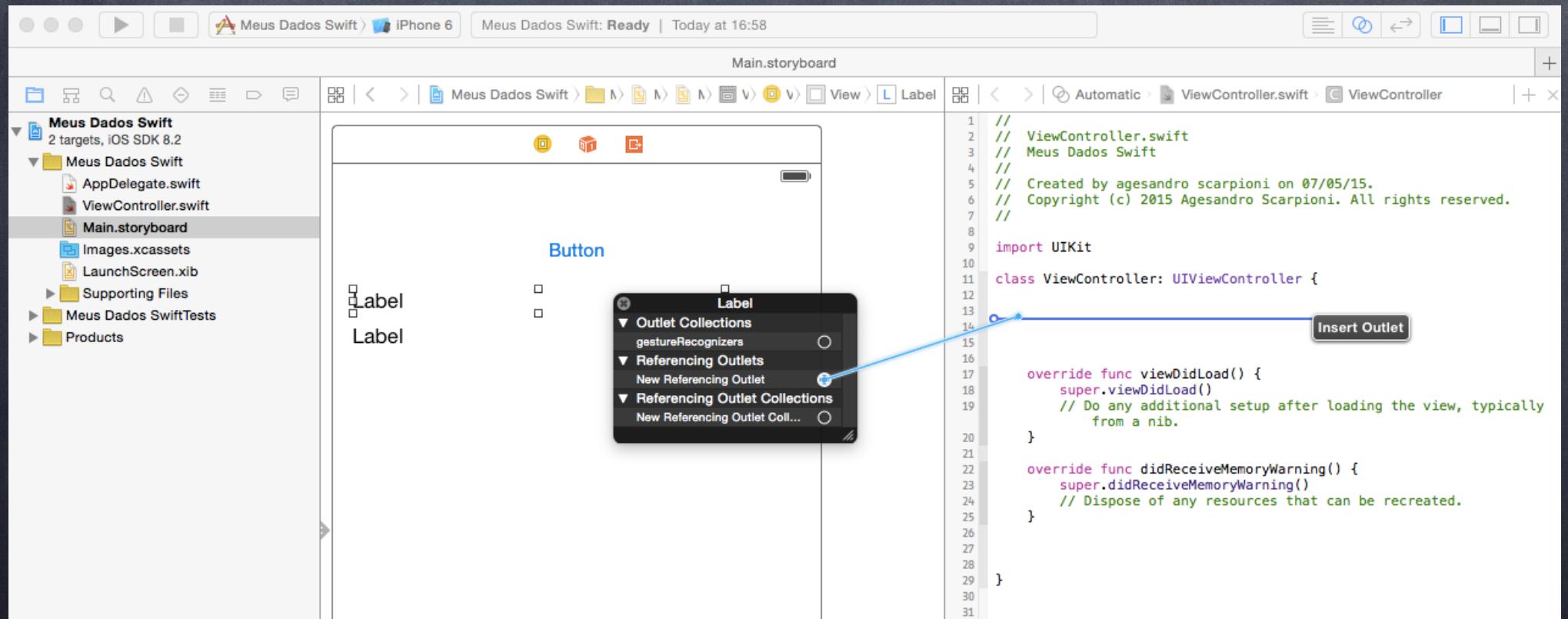
- Depois vamos compartilhar a tela de Storyboard com a viewController obtendo as duas telas simultaneamente clicando neste ícone.



Obs: Verifique se a tela que aparece é ViewController.swift

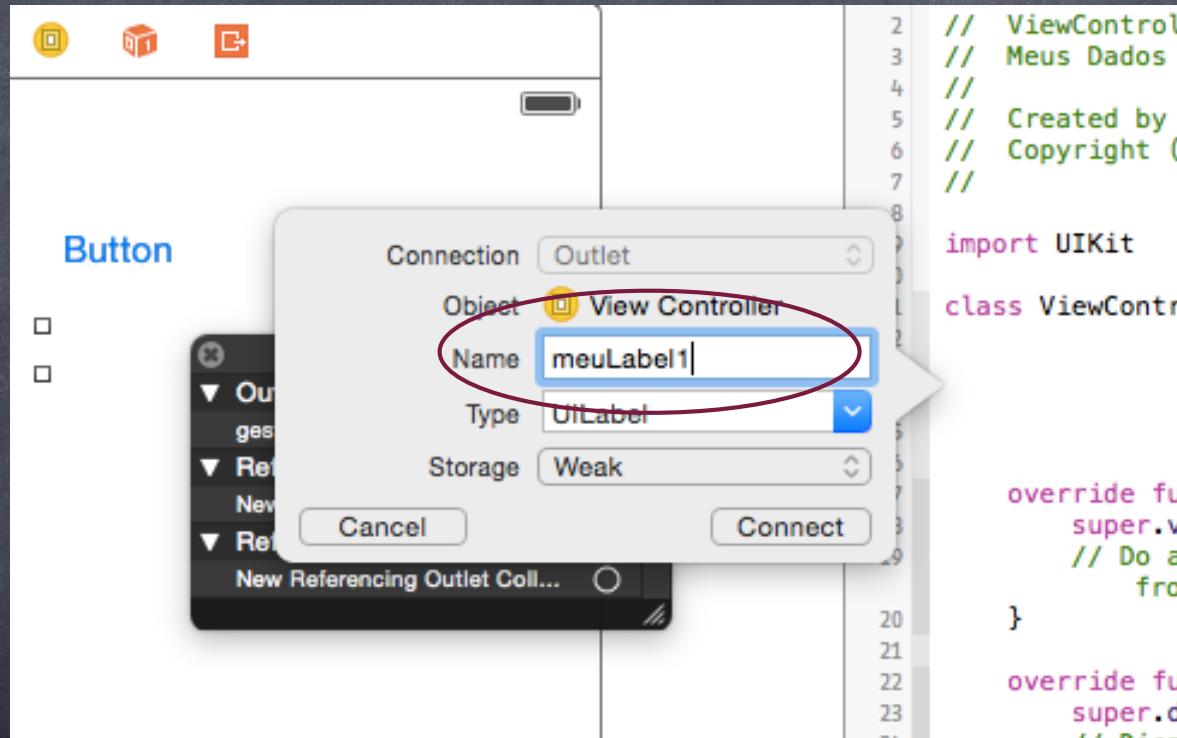
Declarando os outlet's automaticamente

- Clique com o botão direito sobre o 1º label e escolha "New Referencing Outlet's" clicando no local indicado na figura.



Declarando os outlet's 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.



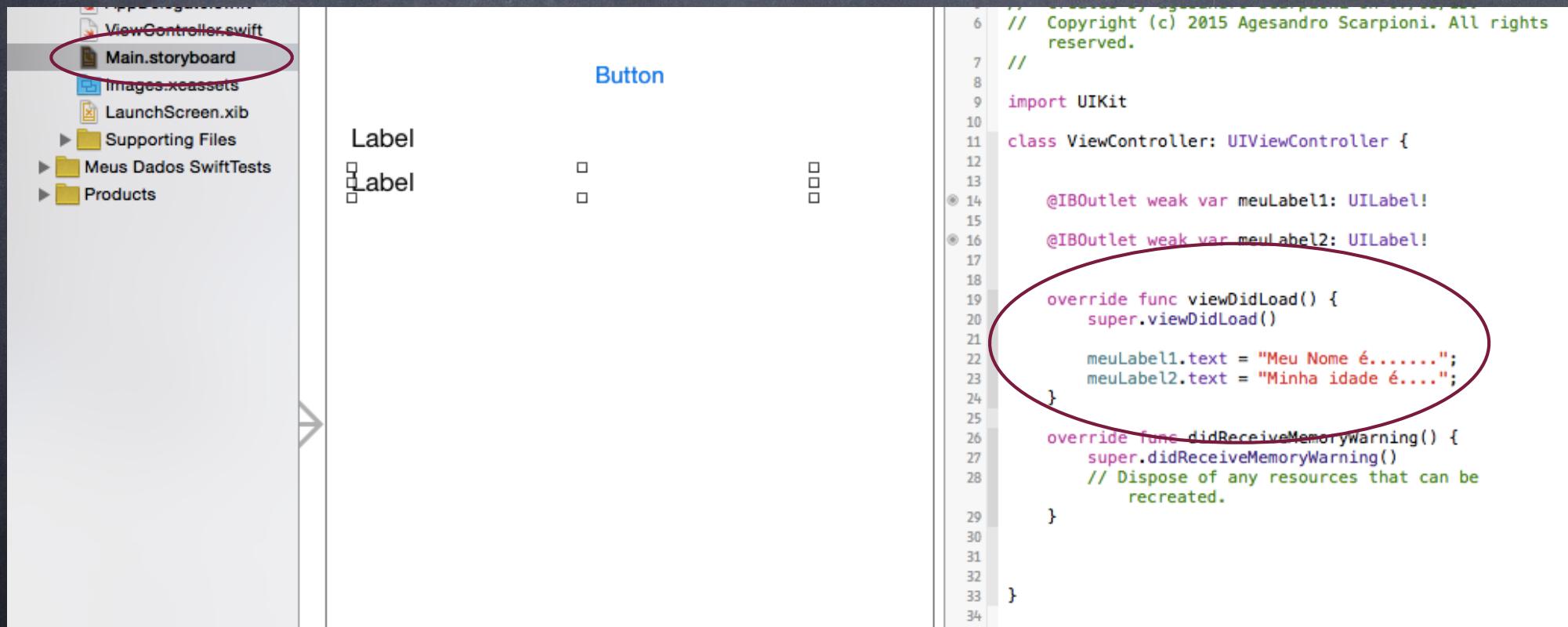
Declarando os outlet's automaticamente

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

```
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13
14     @IBOutlet weak var meuLabel1: UILabel!
15
16     @IBOutlet weak var meuLabel2: UILabel!
17
18
19     override func viewDidLoad() {
20         super.viewDidLoad()
21         // Do any additional setup after loading the view, typically
22         // from a nib.
23     }
24
25     override func didReceiveMemoryWarning() {
26         super.didReceiveMemoryWarning()
27         // Dispose of any resources that can be recreated.
28
29
30
31
32 }
```

Implementando

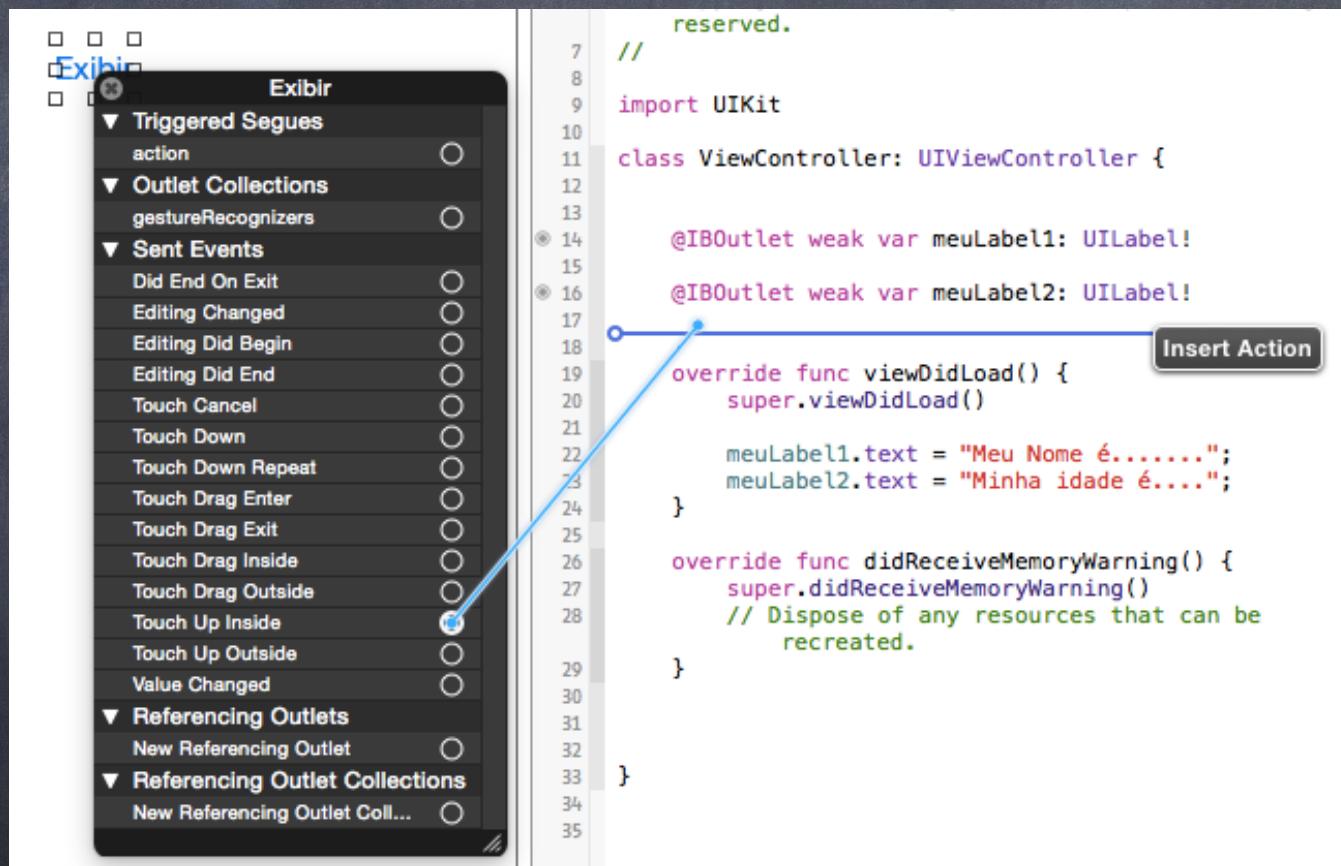
- Na Classe ViewController.swift, existe um método chamado viewDidLoad que é o primeiro método que é executado quando a classe é chamada. Vamos carregar uma frase em cada label como fizemos no primeiro programa.



Obs: viewDidLoad equivale ao Form_Load do VB e ao windowOpened do JFrame no JAVA.

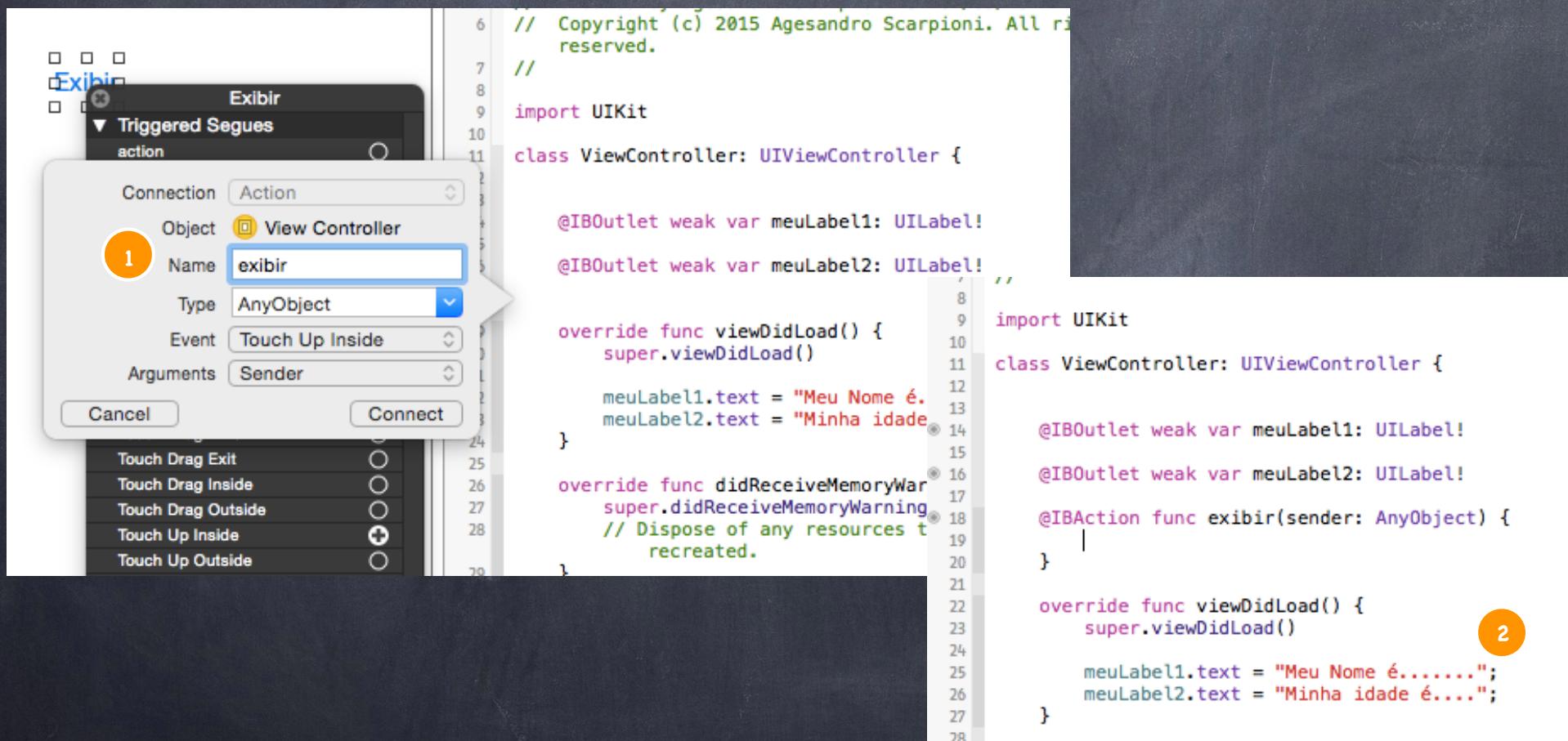
Declarando IBAction automaticamente

- Clique no storyboard, clique com o botão direito do mouse sobre o Exibir, escolha o evento “Touch Up Inside” e arraste para o ViewController.swift.



Declarando IBAction 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.



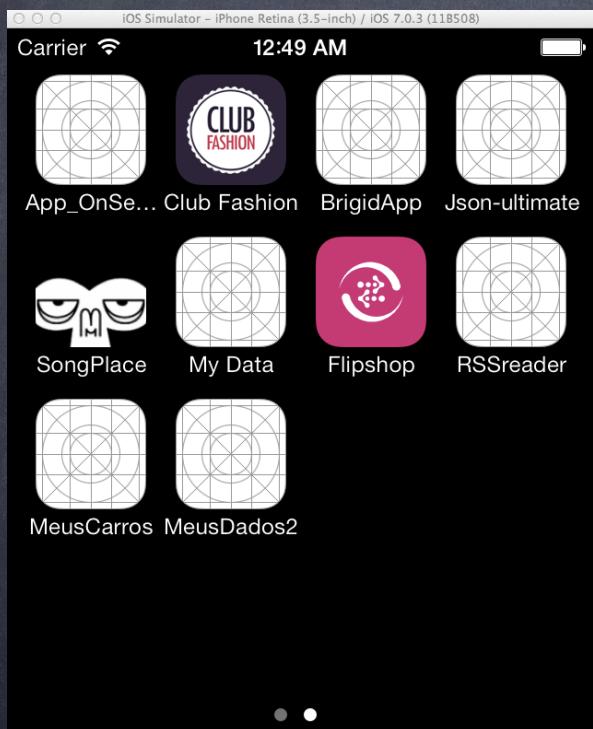
Implementando IBAction

- Escreva as 2 linhas de comando.

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13
14     @IBOutlet weak var meuLabel1: UILabel!
15
16     @IBOutlet weak var meuLabel2: UILabel!
17
18     @IBAction func exibir(sender: AnyObject) {
19         meuLabel1.text = "Agesandro"
20         meuLabel2.text = "43 anos"
21     }
22
23     override func viewDidLoad() {
24         super.viewDidLoad()
25
26         meuLabel1.text = "Meu Nome é.....";
27         meuLabel2.text = "Minha idade é....";
28     }
29
30     override func didReceiveMemoryWarning() {
31         super.didReceiveMemoryWarning()
32         // Dispose of any resources that can be
33         // recreated.
34     }
35
36
37 }
38
39
```

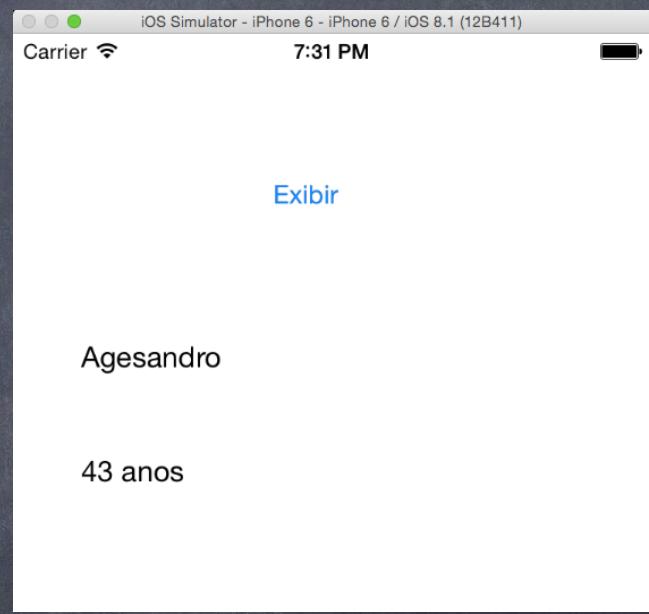
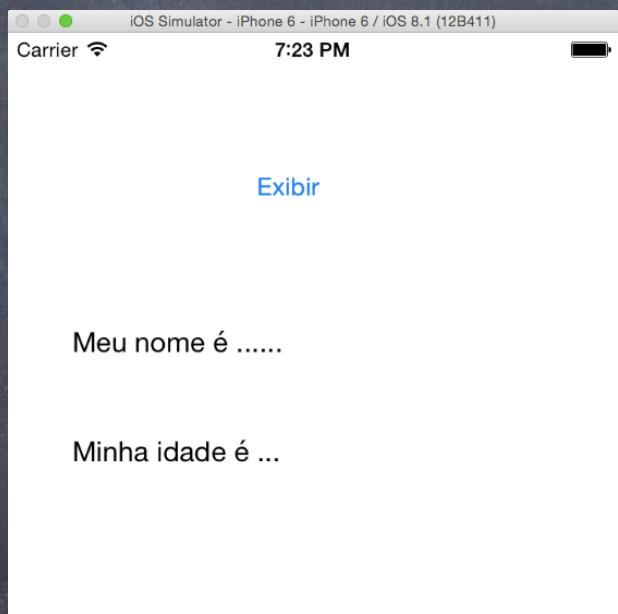
O App no simulador

Quando executamos



Antes de clicar no botão com
informações carregadas pelo
viewDidLoad

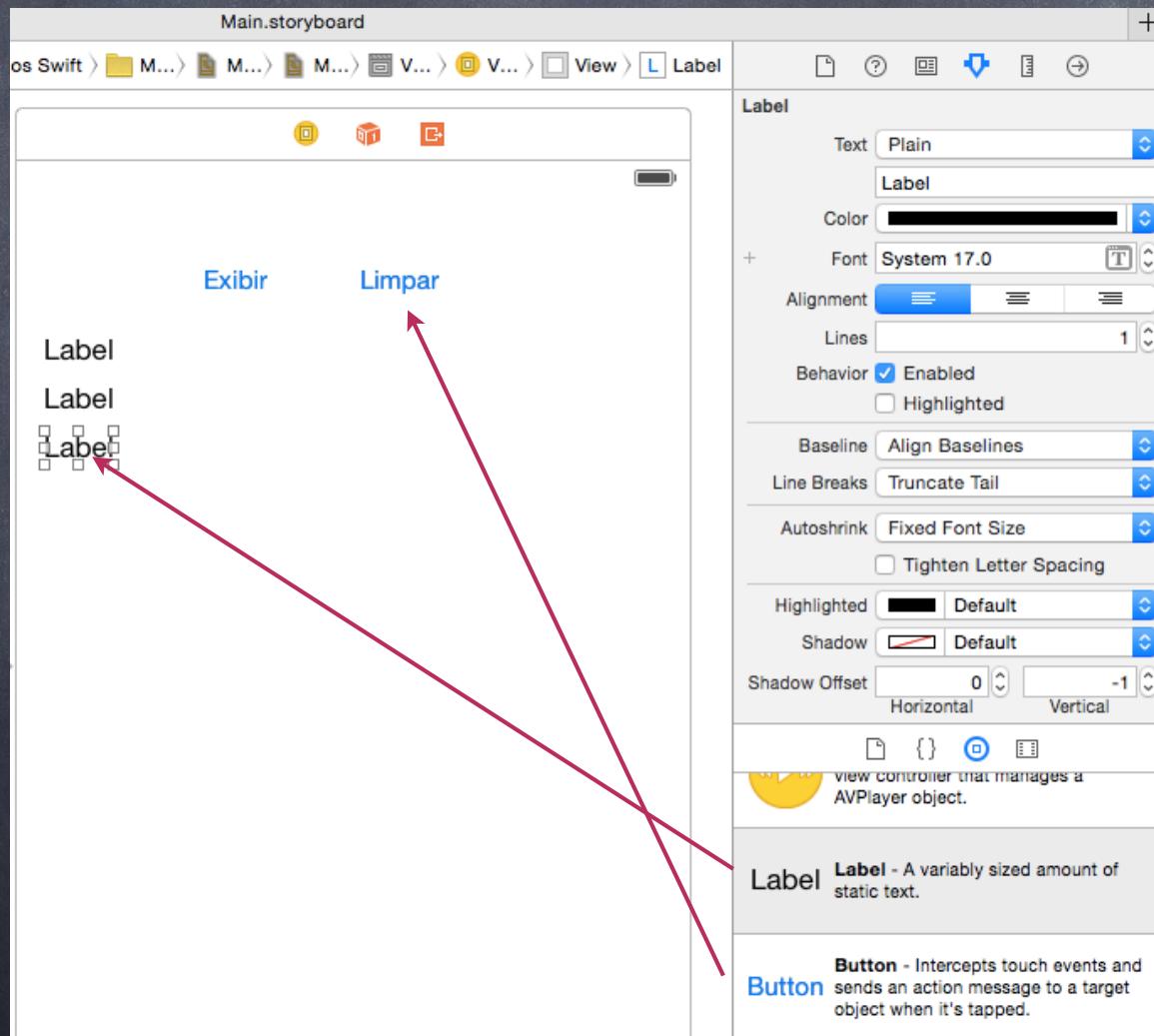
Após clicar no botão



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

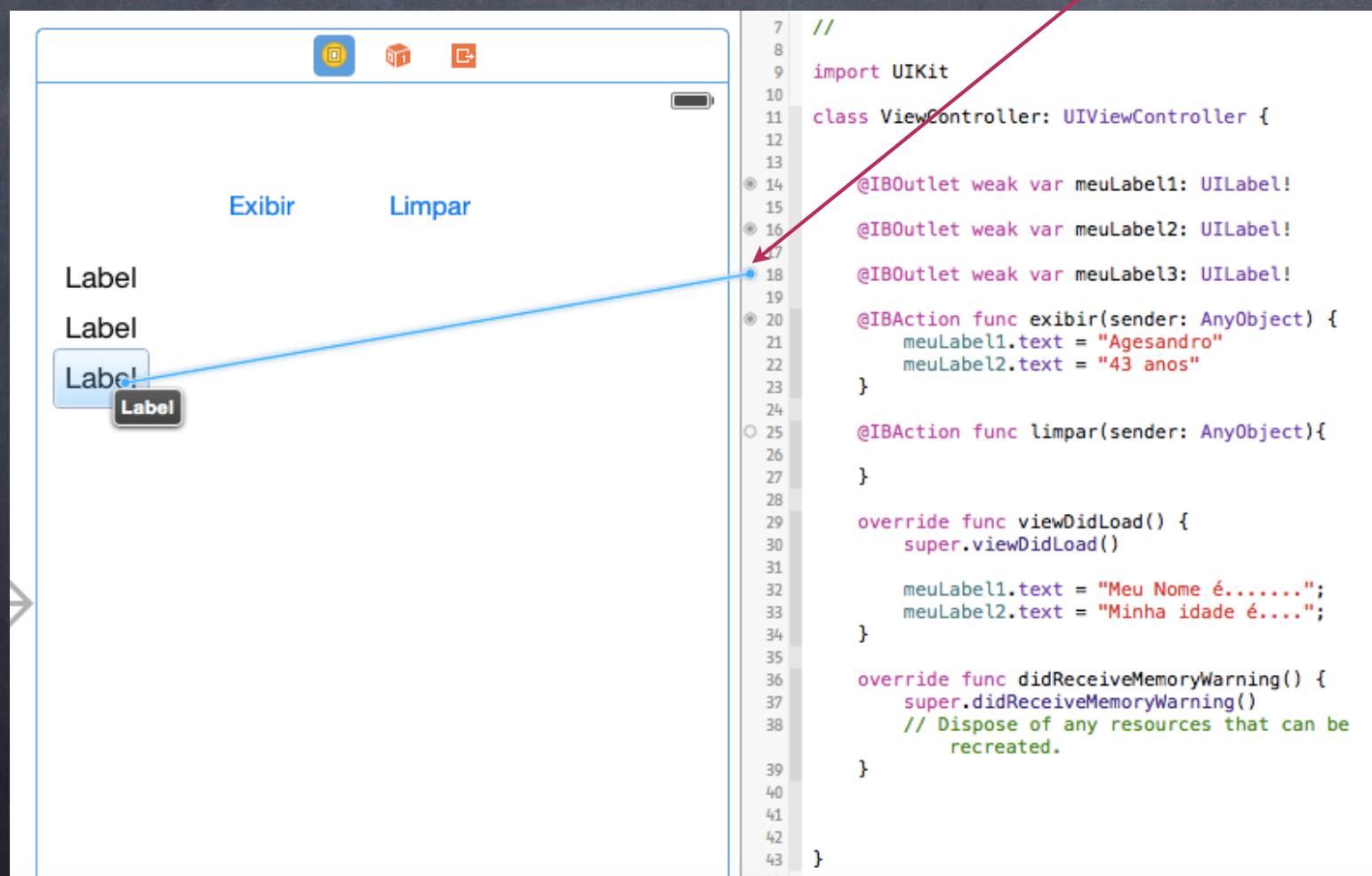
- Digite a linha do Outlet e a linha do IBAction abaixo:

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

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13
14     @IBOutlet weak var meuLabel1: UILabel!
15
16     @IBOutlet weak var meuLabel2: UILabel!
17
18     @IBOutlet weak var meuLabel3: UILabel!
19
20     @IBAction func exibir(sender: AnyObject) {
21         meuLabel1.text = "Agesandro"
22         meuLabel2.text = "43 anos"
23     }
24
25     @IBAction func limpar(sender: AnyObject){
26
27     }
28
29     override func viewDidLoad() {
30         super.viewDidLoad()
31
32         meuLabel1.text = "Meu Nome é.....";
33         meuLabel2.text = "Minha idade é....";
34     }
35
36     override func didReceiveMemoryWarning() {
37         super.didReceiveMemoryWarning()
38         // Dispose of any resources that can be
39         // recreated.
40     }
41
42
43
44 }
```

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 do controller ao Label da View.



Implementando IBAction e IBOulet de forma manual

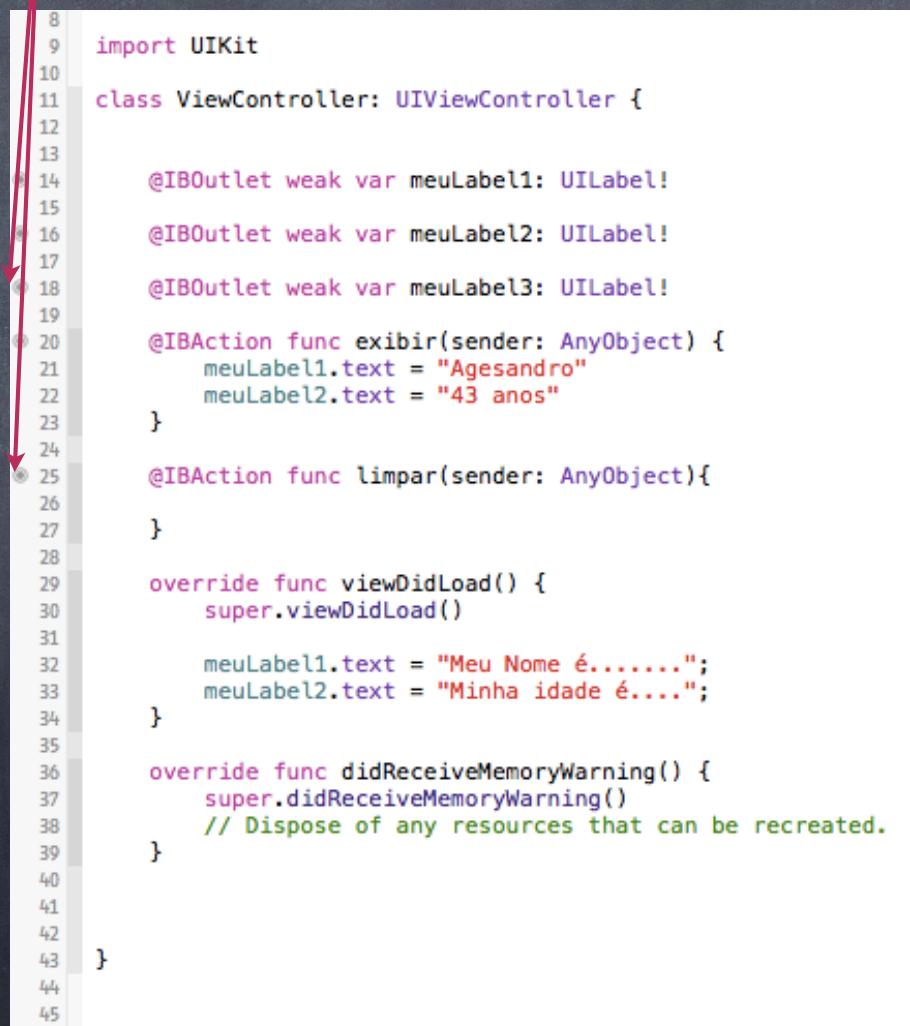
- Faça o mesmo com o IBAction Limpar.

The image shows a screenshot of the Xcode IDE. On the left, a storyboard is displayed with three labels and a button labeled "Limpar". A blue line connects the "Limpar" button to the code editor on the right, indicating a connection between the storyboard element and the implemented code. The code editor contains the following Swift code:

```
//  
import UIKit  
  
class ViewController: UIViewController {  
  
    @IBOutlet weak var meuLabel1: UILabel!  
    @IBOutlet weak var meuLabel2: UILabel!  
    @IBOutlet weak var meuLabel3: UILabel!  
  
    @IBAction func exibir(sender: AnyObject) {  
        meuLabel1.text = "Agesandro"  
        meuLabel2.text = "43 anos"  
    }  
  
    @IBAction func limpar(sender: AnyObject){  
    }  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        meuLabel1.text = "Meu Nome é.....";  
        meuLabel2.text = "Minha idade é....";  
    }  
}
```

Implementando IBAction e IBOOutlet de forma manual

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



```
8 import UIKit
9
10 class ViewController: UIViewController {
11
12
13
14     @IBOutlet weak var meuLabel1: UILabel!
15
16     @IBOutlet weak var meuLabel2: UILabel!
17
18     @IBOutlet weak var meuLabel3: UILabel!
19
20     @IBAction func exibir(sender: AnyObject) {
21         meuLabel1.text = "Agesandro"
22         meuLabel2.text = "43 anos"
23     }
24
25     @IBAction func limpar(sender: AnyObject){
26
27     }
28
29     override func viewDidLoad() {
30         super.viewDidLoad()
31
32         meuLabel1.text = "Meu Nome é.....";
33         meuLabel2.text = "Minha idade é....";
34     }
35
36     override func didReceiveMemoryWarning() {
37         super.didReceiveMemoryWarning()
38         // Dispose of any resources that can be recreated.
39     }
40
41
42
43
44
45 }
```

Implementando IBAction e IBOulet de forma manual

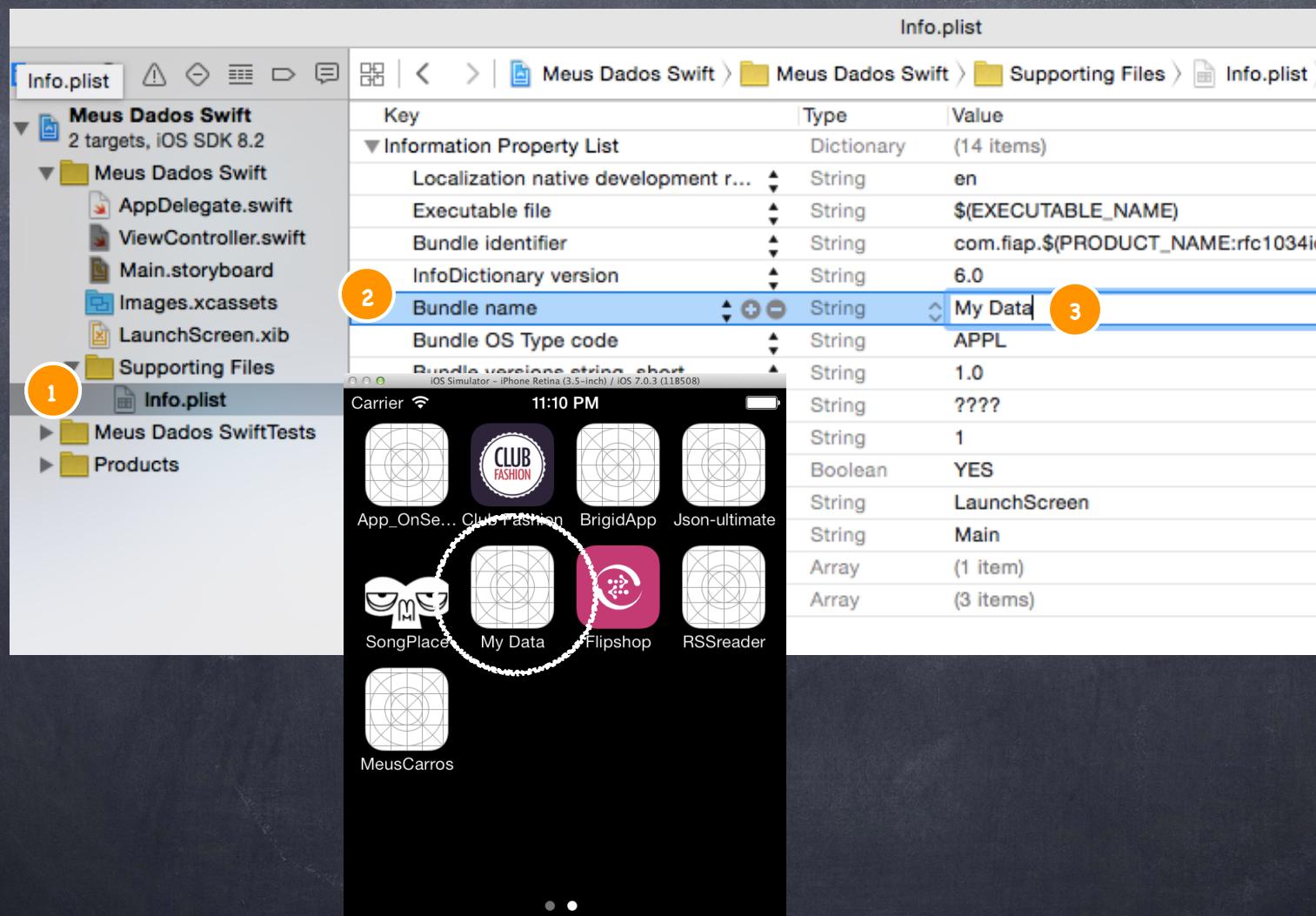
- Inclua as linhas do Label3 no DidLoad e no Exibir.

- No limpar escreva as linhas indicadas.

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13
14     @IBOutlet weak var meuLabel1: UILabel!
15
16     @IBOutlet weak var meuLabel2: UILabel!
17
18     @IBOutlet weak var meuLabel3: UILabel!
19
20     @IBAction func exibir(sender: AnyObject) {
21         meuLabel1.text = "Agesandro"
22         meuLabel2.text = "43 anos"
23         meuLabel3.text = "São Paulo"
24     }
25
26     @IBAction func limpar(sender: AnyObject){
27         meuLabel1.text = "";
28         meuLabel2.text = "";
29         meuLabel3.text = "";
30     }
31
32     override func viewDidLoad() {
33         super.viewDidLoad()
34
35         meuLabel1.text = "Meu Nome é.....";
36         meuLabel2.text = "Minha idade é....";
37         meuLabel3.text = "Minha Cidade é..";
38     }
39
40     override func didReceiveMemoryWarning() {
41         super.didReceiveMemoryWarning()
42         // Dispose of any resources that can be
43         // recreated.
44     }
45
46 }
```

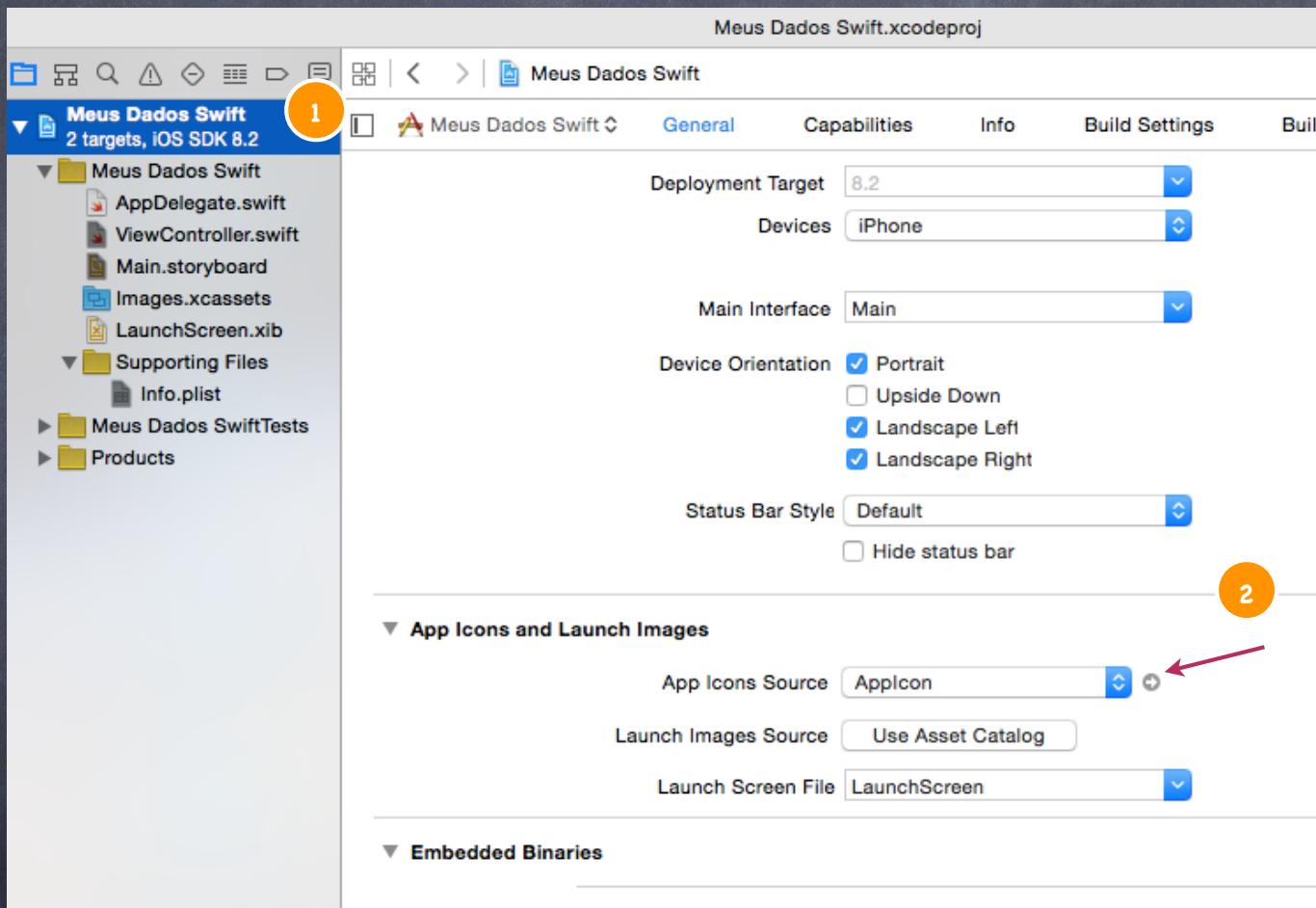
Como trocar o nome do App

- 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.



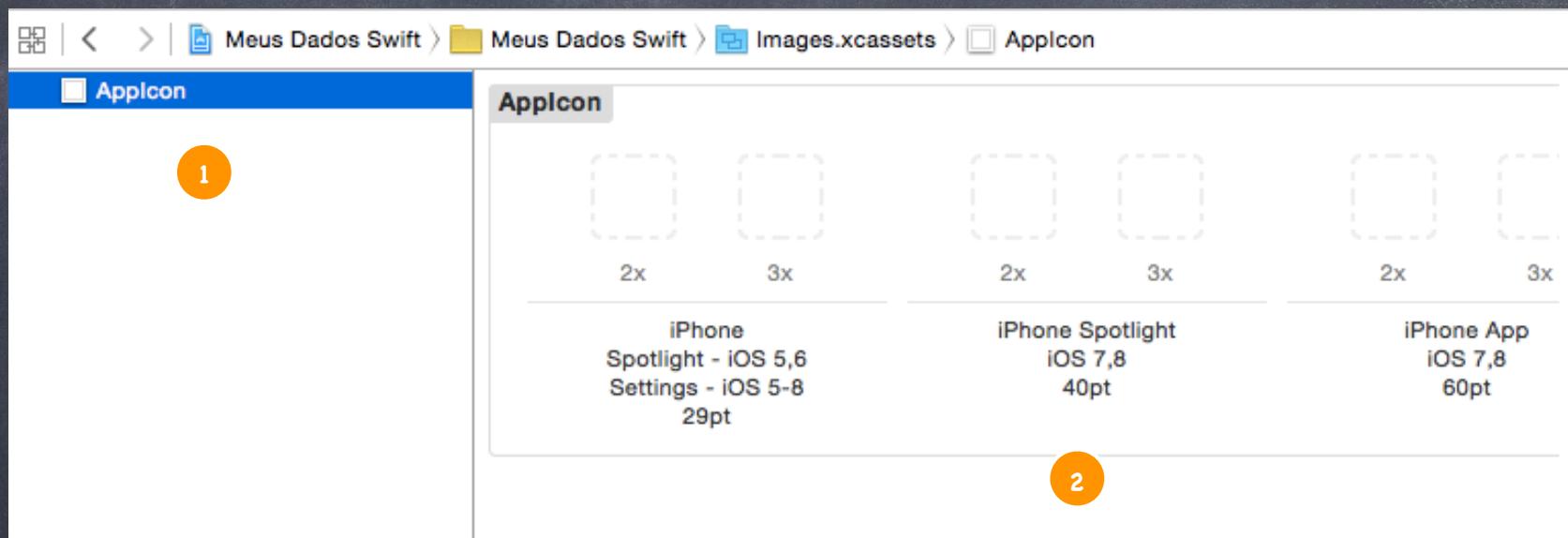
Alterando o Ícone do App

- Clique em seu projeto (1), clique na seta do 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).

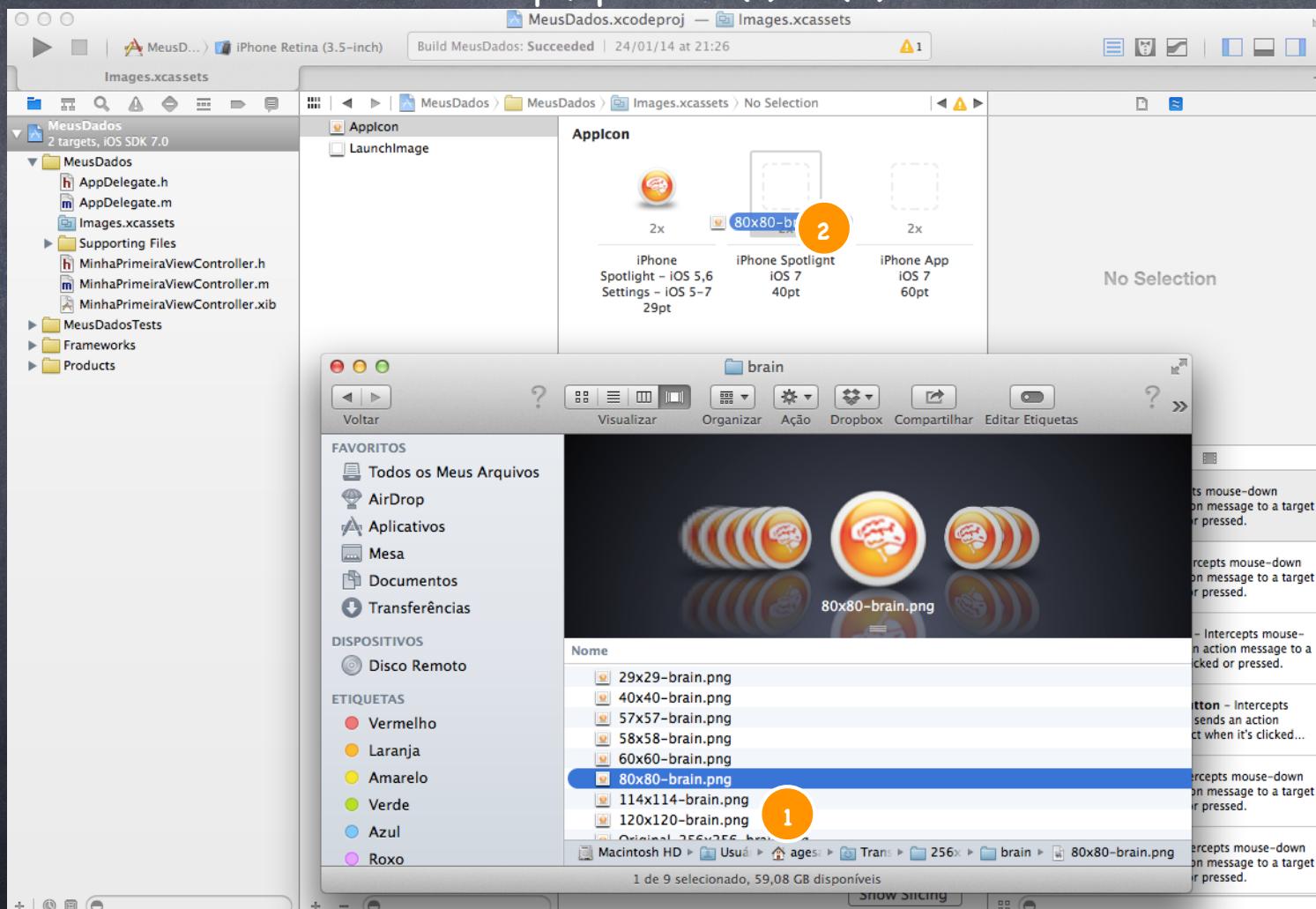
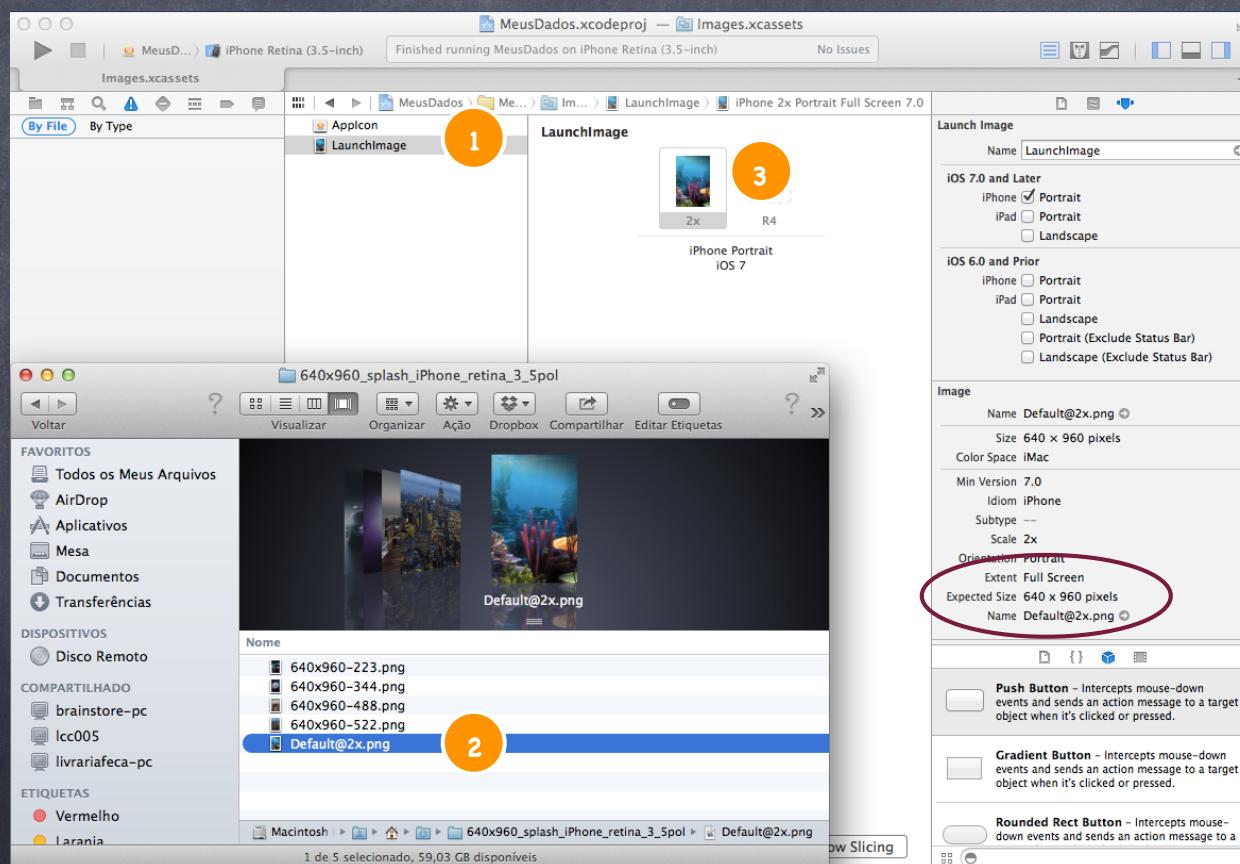


Imagen de splash screen

- 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.

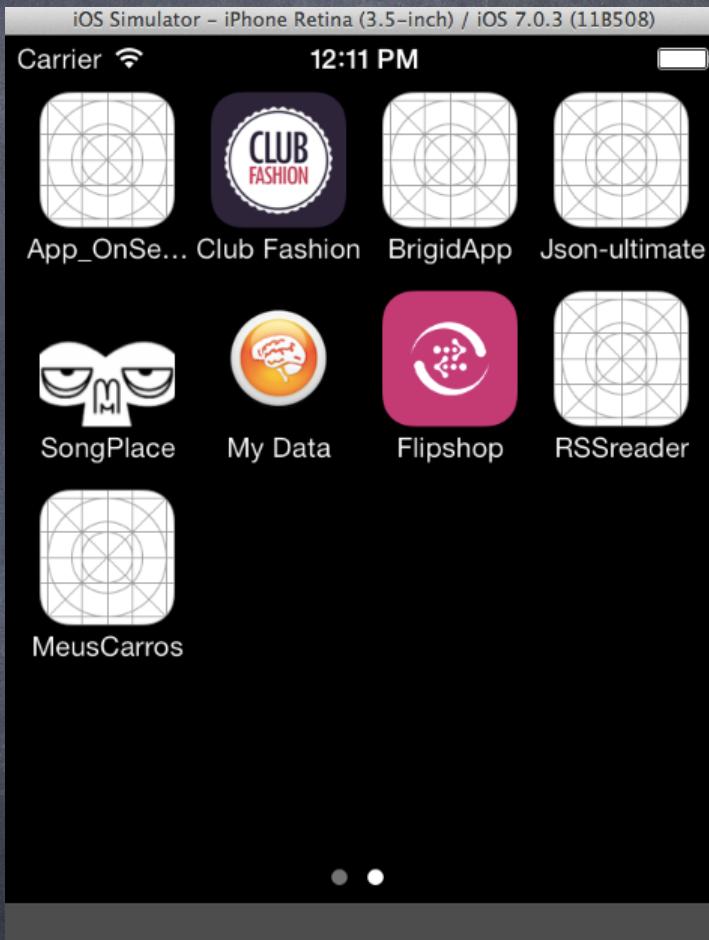


Tabela de padrões de nomes de imagem

- 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.

Veja os Links para Gerar Icons e Telas Splash

- ⦿ <http://makeappicon.com>
- ⦿ <http://ticons.fokkezb.nl>

Utilizando break points

- Existem duas formas de inserir um break point em um programa.

The screenshot shows the Xcode interface during a debugging session on an iPhone 6.1 Simulator. The main window displays the code for `MinhaPrimeiraViewController.m`. A red arrow points from the text "Existem duas formas de inserir um break point em um programa." to the line number 43 in the code, where a green breakpoint marker is placed. Another red arrow points from the text "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." to the "All Output" tab at the bottom of the Xcode window. The output pane shows the result of the `NSLog` statement at line 43: "2013-05-30 03:52:57.402 MeusDados[2142:c07] Atenção".

```

// MinhaPrimeiraViewController.m
//
// Created by agesandro scarpioni on 30/05/13.
// Copyright (c) 2013 Agesandro Scarpioni. All rights reserved.

#import "MinhaPrimeiraViewController.h"

@interface MinhaPrimeiraViewController : UIViewController

@end

@implementation MinhaPrimeiraViewController

- (id)initWithNibName:(NSString *)NibName bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view from its nib.
    meuLabel1.text = @"Meu nome é .....";
    meuLabel2.text = @"Minha idade é .....";
    //rode o programa e observe que o texto não aparece porque o UILabel não foi vinculado ao label que foi desenhado
}

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

-(IBAction)Exibir:(id)sender{
    NSLog(@"Atenção");
    meuLabel1.text = @"Agesandro";
    meuLabel2.text = @"41 anos";
    //depois desse ponto precisamos vincular o evento do botao com esse método;
}

-(void)dealloc{
    [meuLabel1 release];
    [meuLabel2 release];
    [super dealloc];
}

```

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, tente fazer sem olhar os slides.

- ⦿ 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.