FIAP

App com Nome e Idade

X-Code - Objective C

Prof. Agesandro Scarpioni
agesandro@fiap.com.br

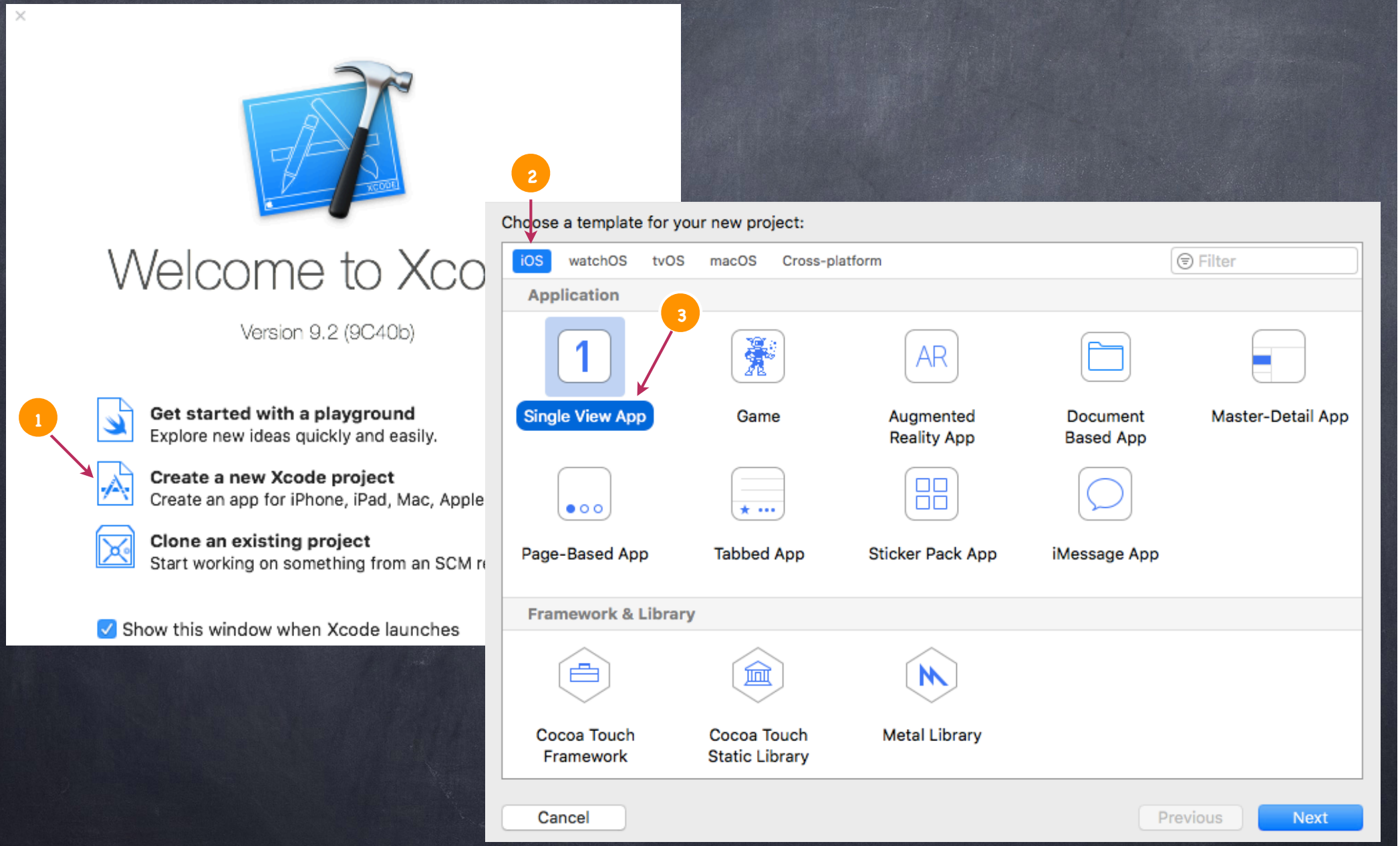
Nome e Idade

FIAP

- 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.

Iniciando Forma 1

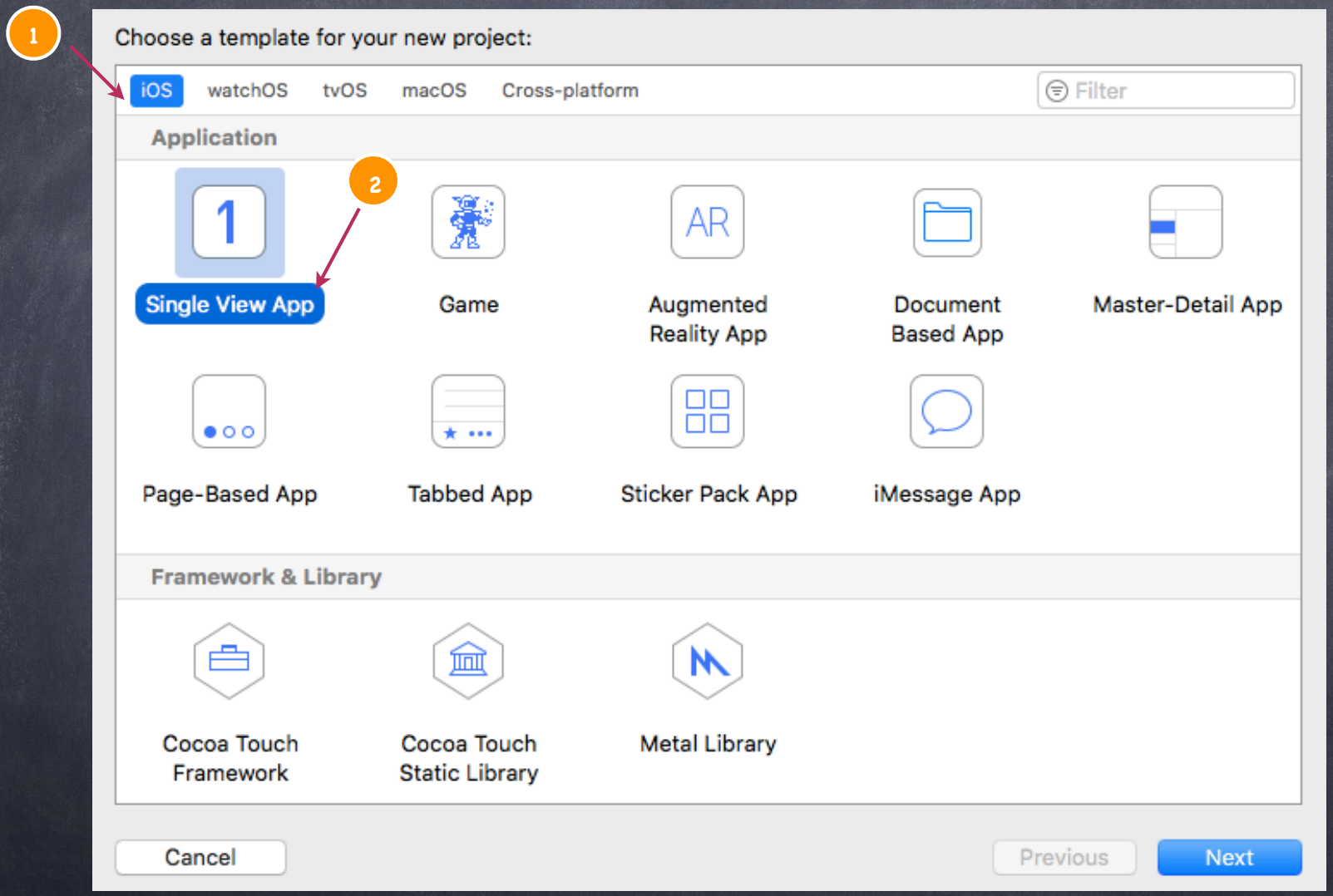
FIAP



Iniciando Forma 2

FIAP

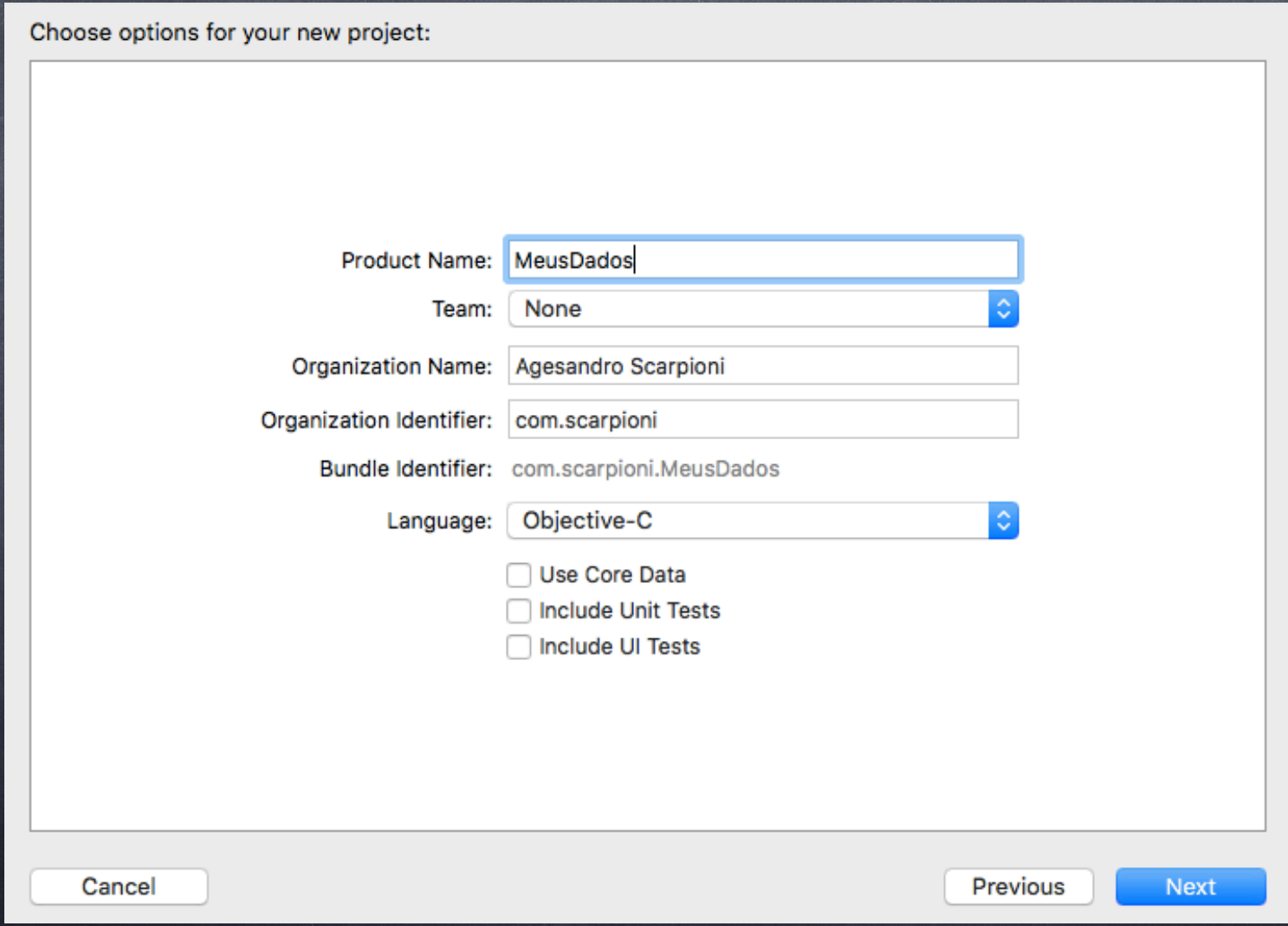
- Ou Clique em File -> New Project -> Application -> Single View Application.



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

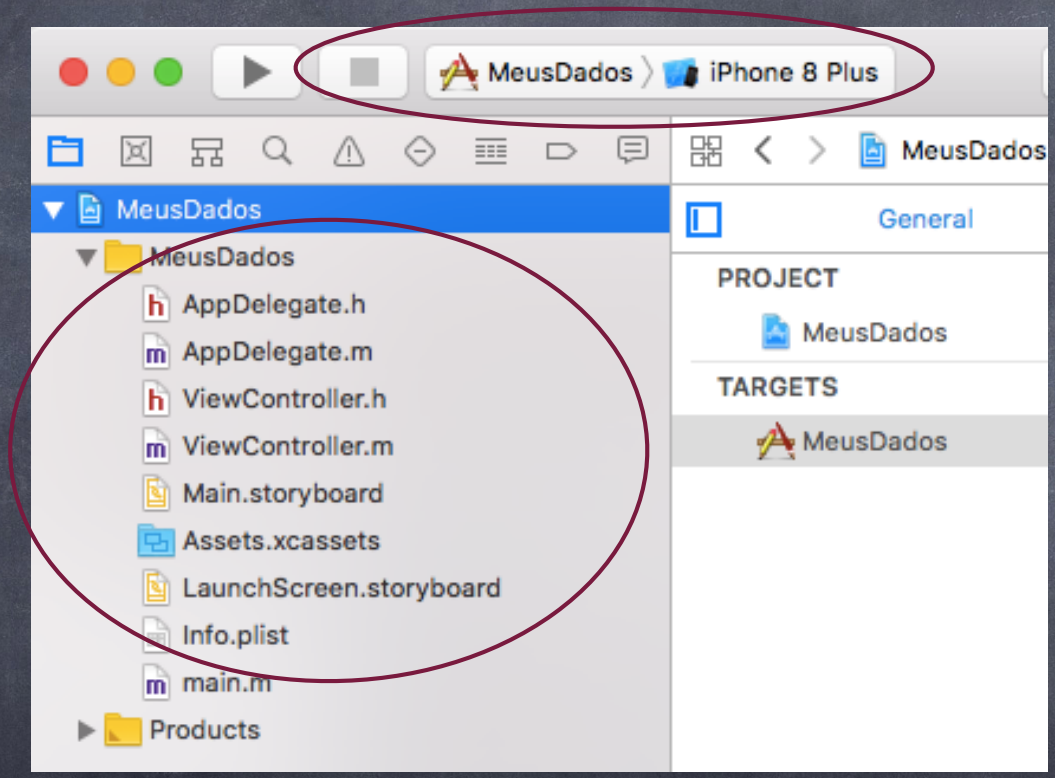
O App

- Preencha com os dados abaixo, lembre-se que o Organization Identifier é como se fosse o pacote no Java ou o namespace do VB, em Language use Objective-C.



Tudo em seu lugar

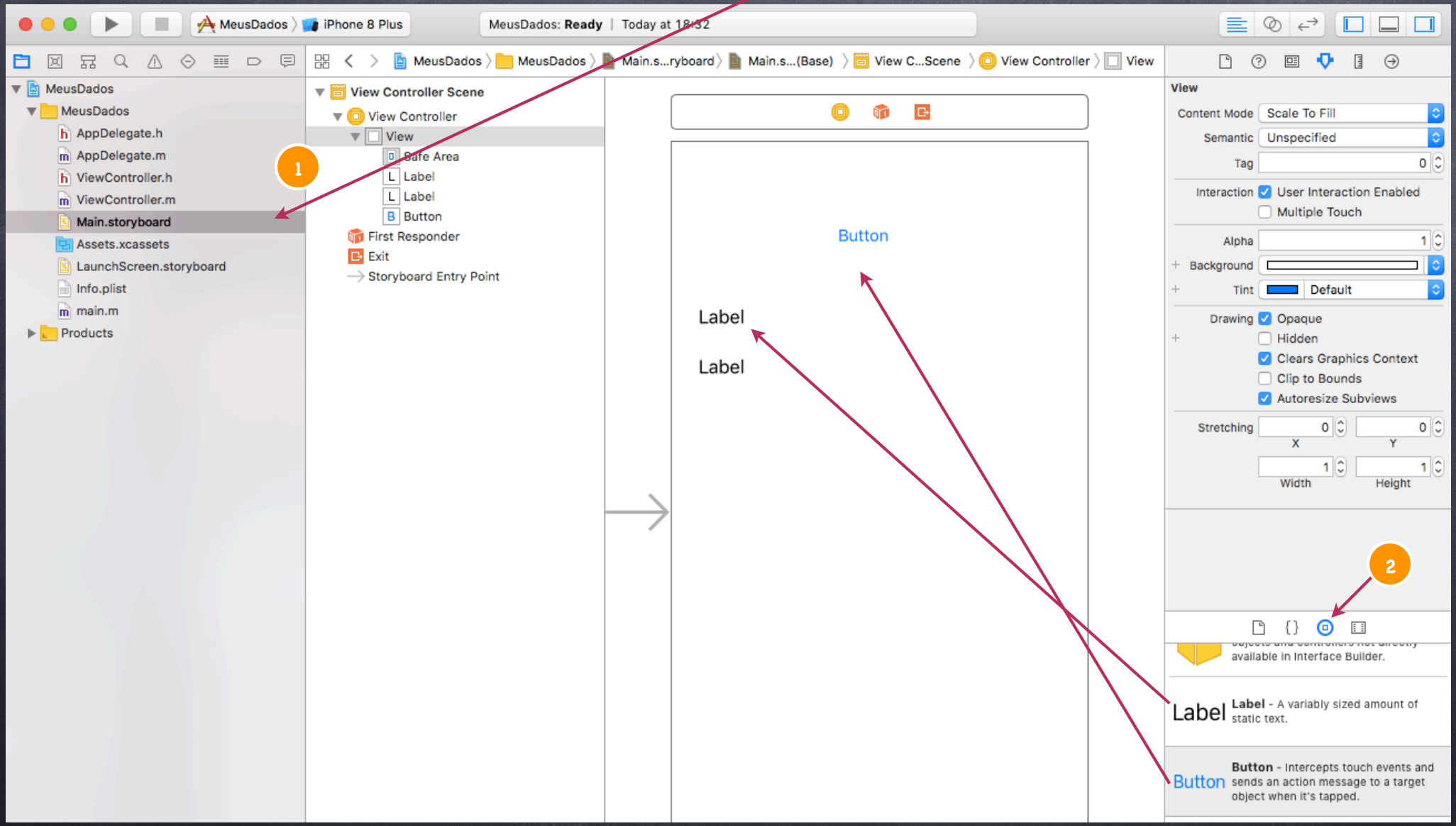
- Note que utilizando este template foi incluída uma classe chamada ViewController .h e .m, o delegate que já possui a classe referenciada e o Storyboard.



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

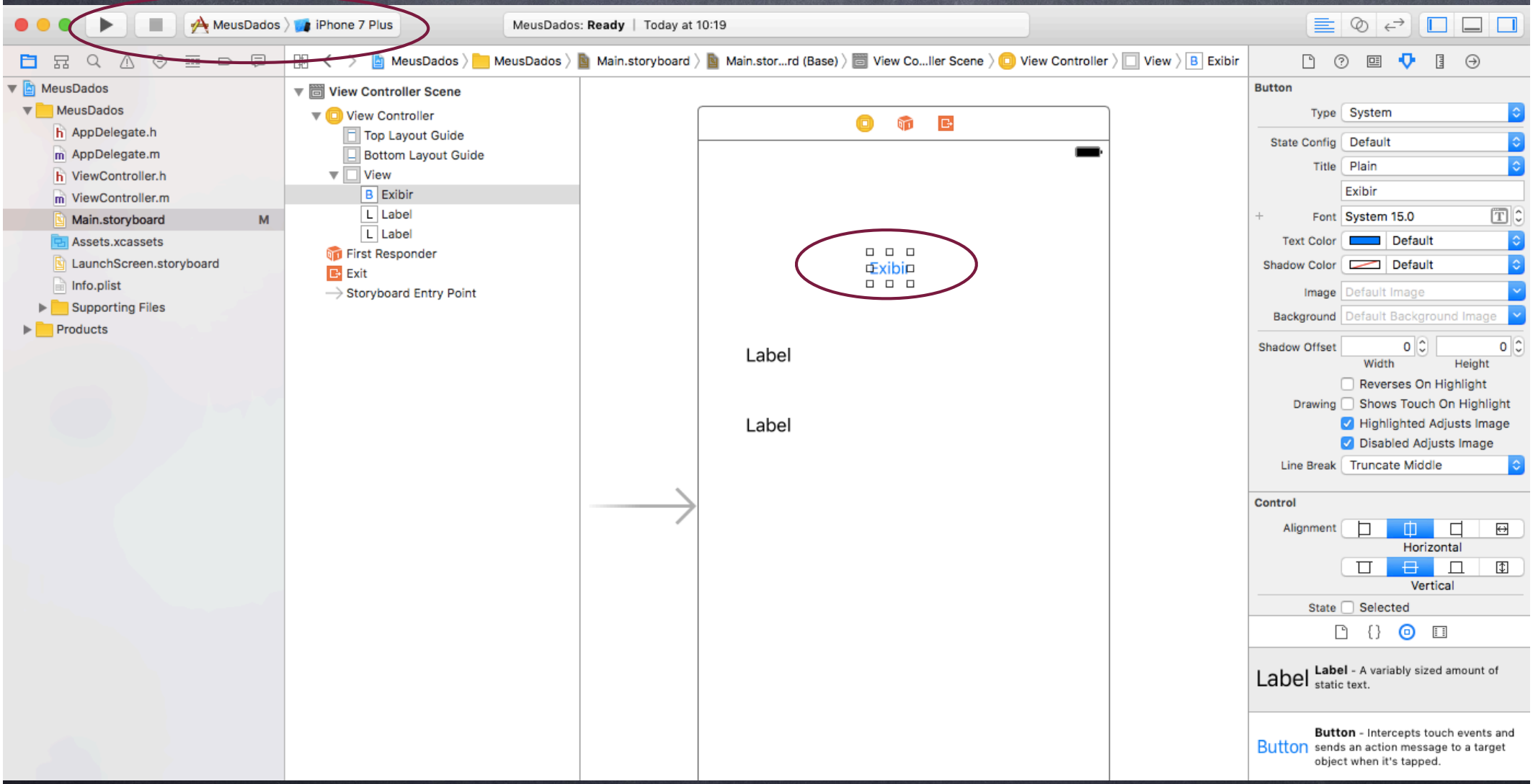
Adicionando objetos

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



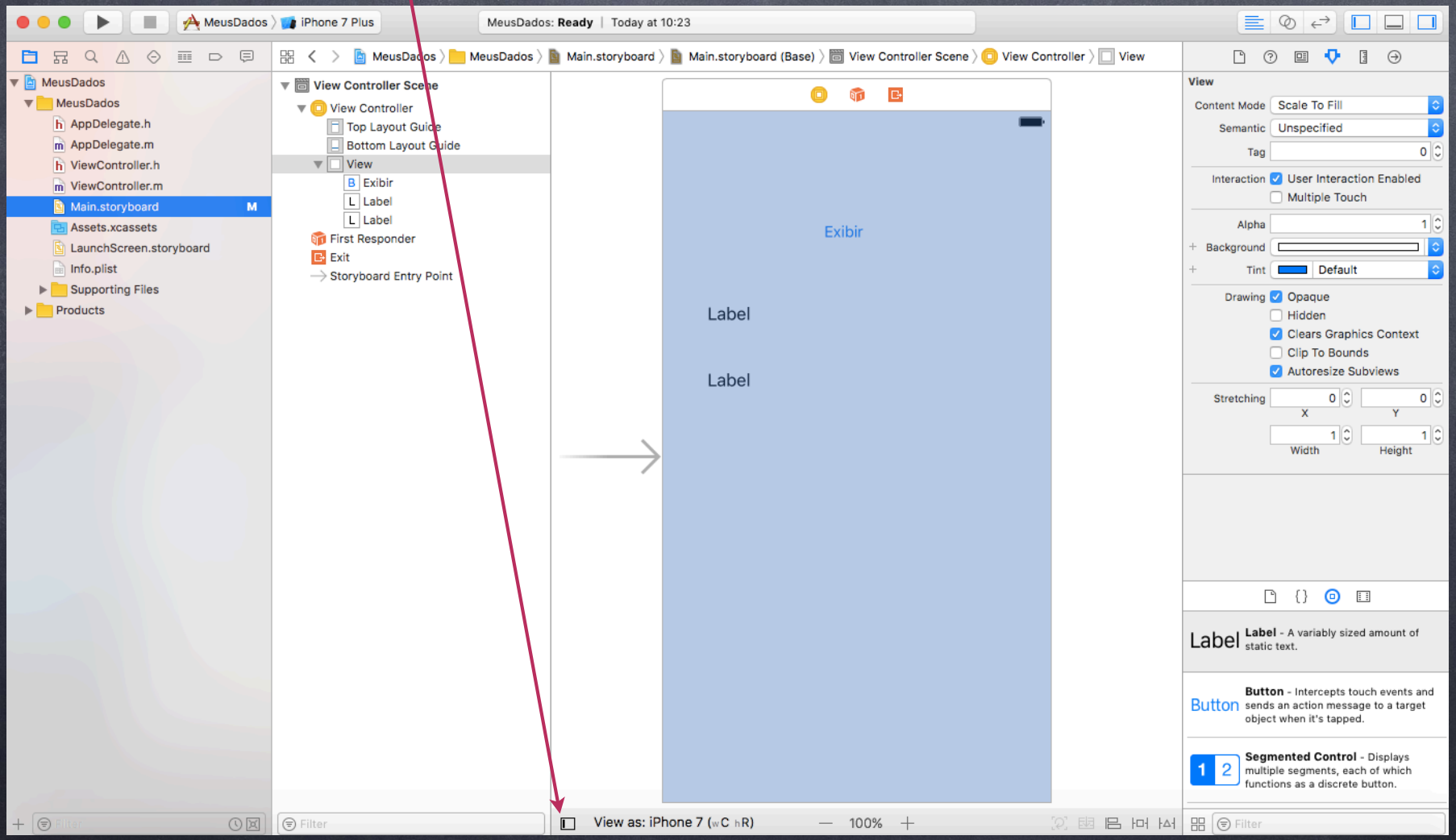
Executando

- Clique duas vezes sobre o botão e digite "Exibir" no texto, escolha um modelo de simulador ou deixe o modelo sugerido pelo Xcode e execute seu App clicando em Run ou utilizando o atalho command + R.



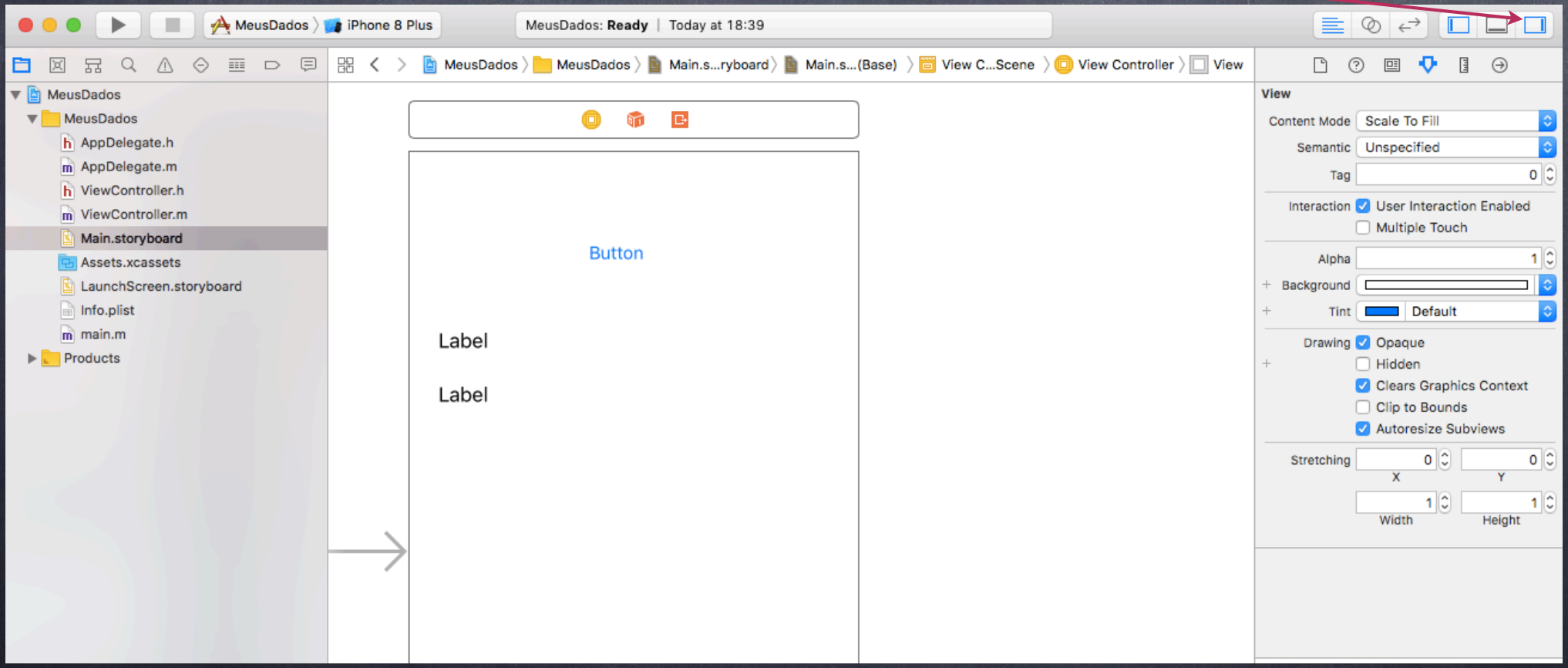
Continuando

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



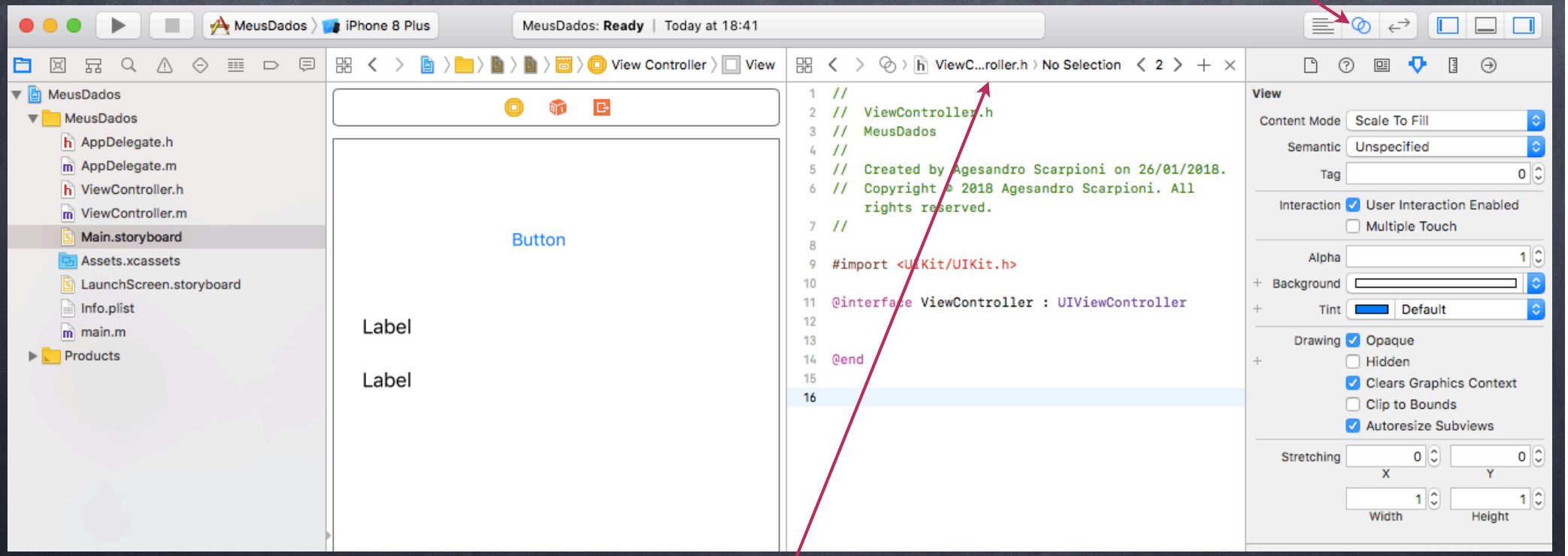
Declarando os outlet's no .h

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



Declarando os outlet's no .h

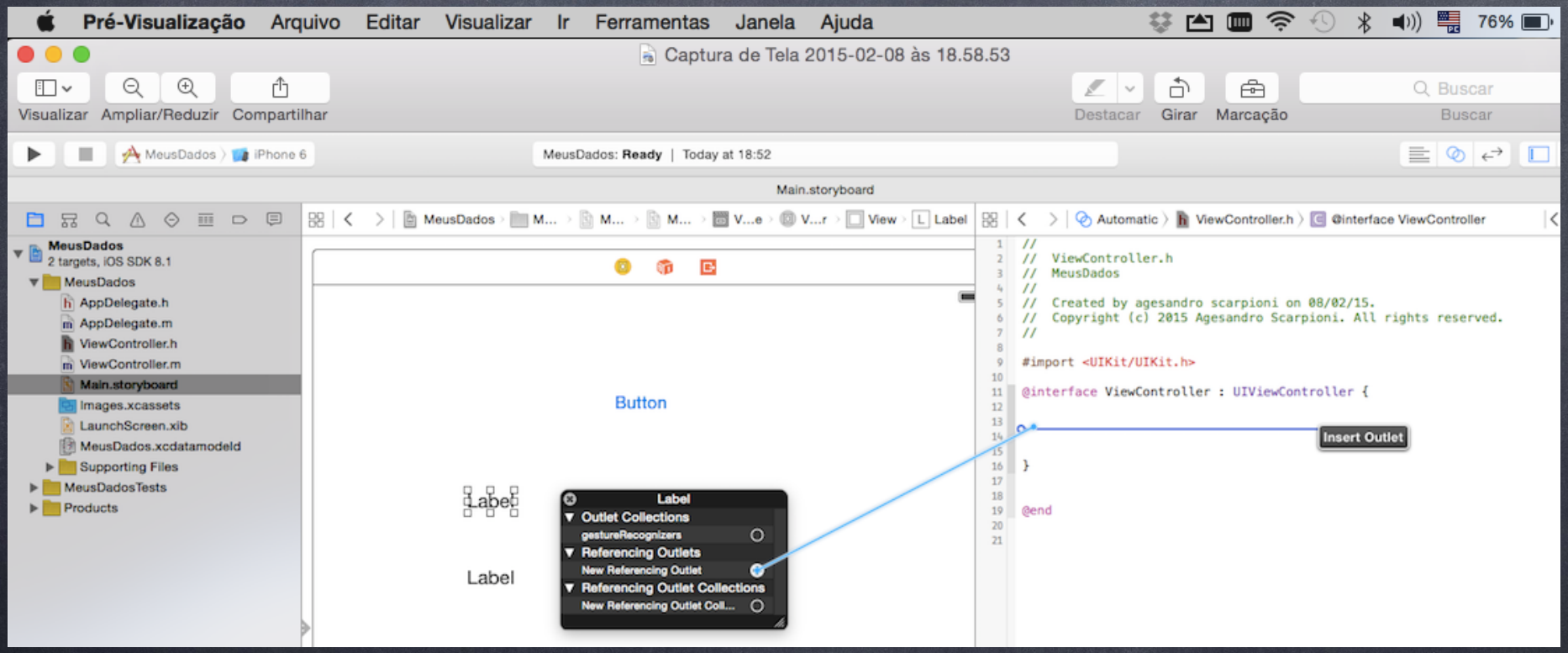
- Depois vamos compartilhar a tela de Storyboard com a viewController.h obtendo as duas telas simultaneamente clicando neste ícone.



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

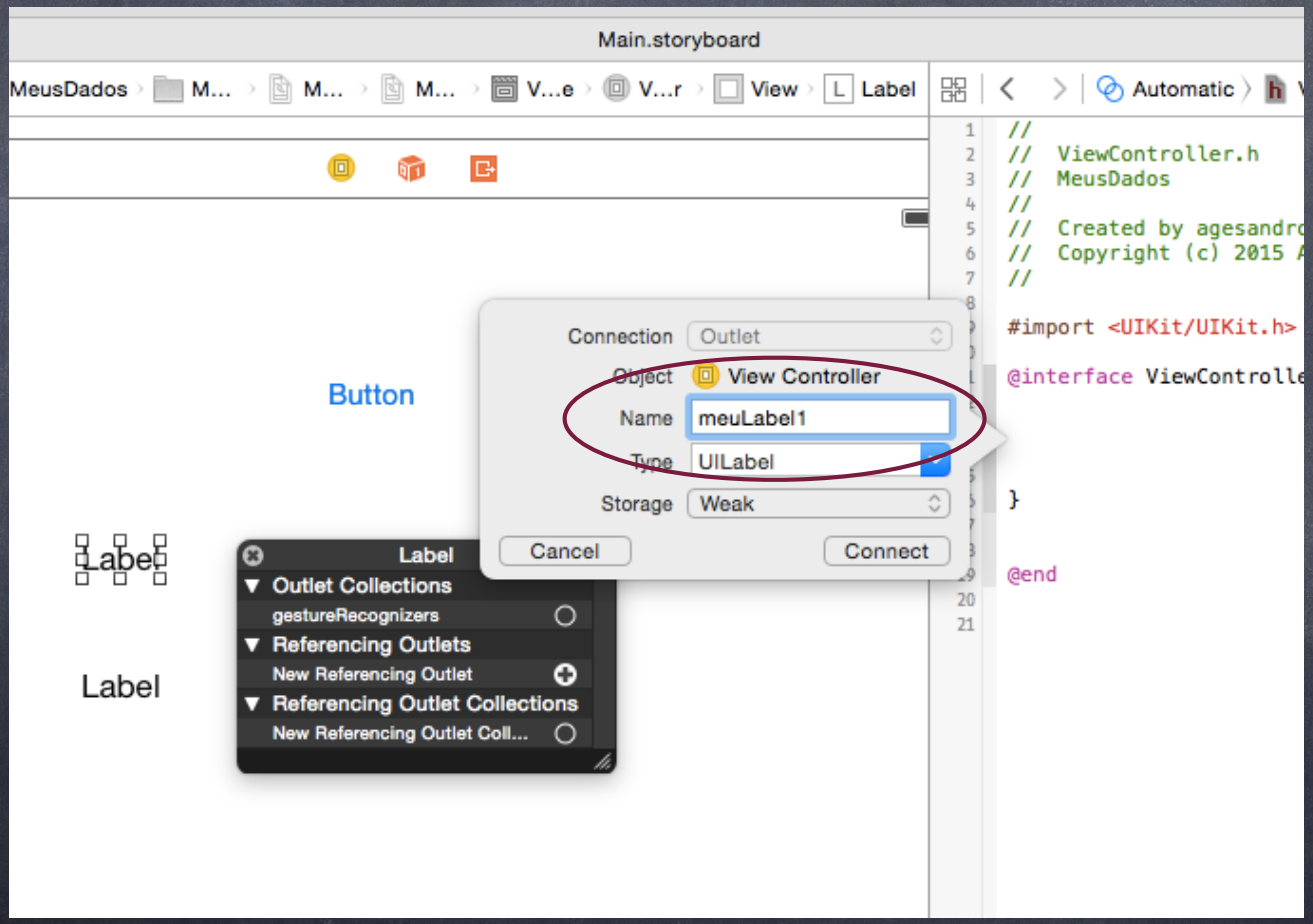
- Coloque duas chaves { } depois da declaração da interface, clique com o botão direito sobre o 1º label e escolha "New Referencing Outlet's" clicando no local indicado na figura e arrastando até a área entre as chaves { }.



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

FIAP

- 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.

Automatic › ViewController.h › @interface ViewController

```
//
//  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.

MeusDados 〉 iPhone 6 | (null): **Ready** | Today at 19:05

ViewController.m

MeusDados 〉 MeusDados 〉 ViewController.m 〉 -viewDidLoad

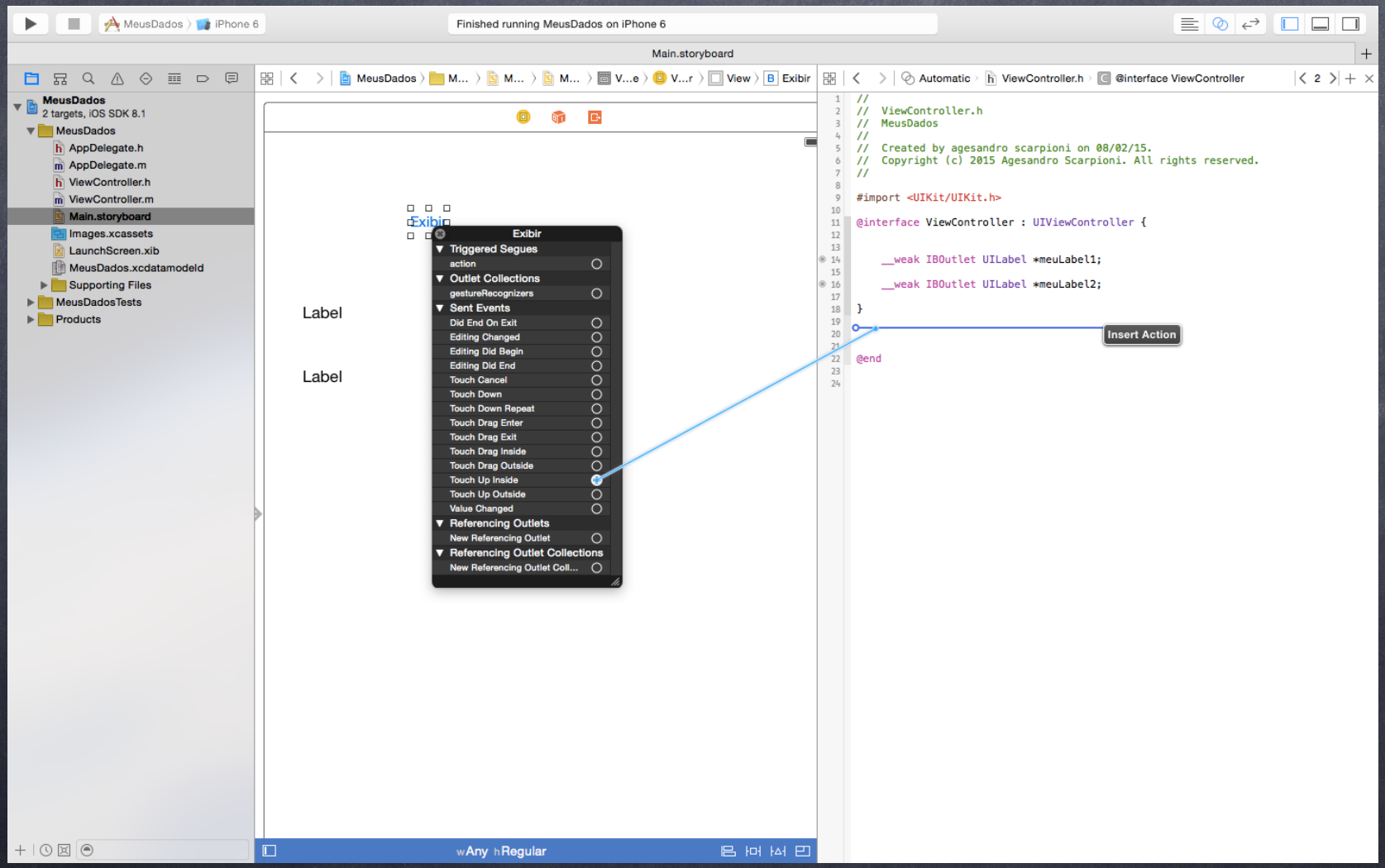
MeusDados
2 targets, iOS SDK 8.1
- MeusDados
  - AppDelegate.h
  - AppDelegate.m
  - ViewController.h
  - ViewController.m
  - Main.storyboard
  - Images.xcassets
  - LaunchScreen.xib
  - MeusDados.xcdatamodeld
  - Supporting Files
- MeusDadosTests
- Products

```
1  //
2  //  ViewController.m
3  //  MeusDados
4  //
5  //  Created by agesandro scarpioni on 08/02/15.
6  //  Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7  //
8
9  #import "ViewController.h"
10
11 @interface ViewController ()
12
13 @end
14
15 @implementation ViewController
16
17 - (void)viewDidLoad {
18     [super viewDidLoad];
19
20     meuLabel1.text = @"Meu nome é ......";
21     meuLabel2.text = @"Minha idade é ...";
22     // Do any additional setup after loading the view, typically from a nib.
23 }
24
25 - (void)didReceiveMemoryWarning {
26     [super didReceiveMemoryWarning];
27     // Dispose of any resources that can be recreated.
28 }
29
30 @end
31
```

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

Declarando IBAction no .h automaticamente

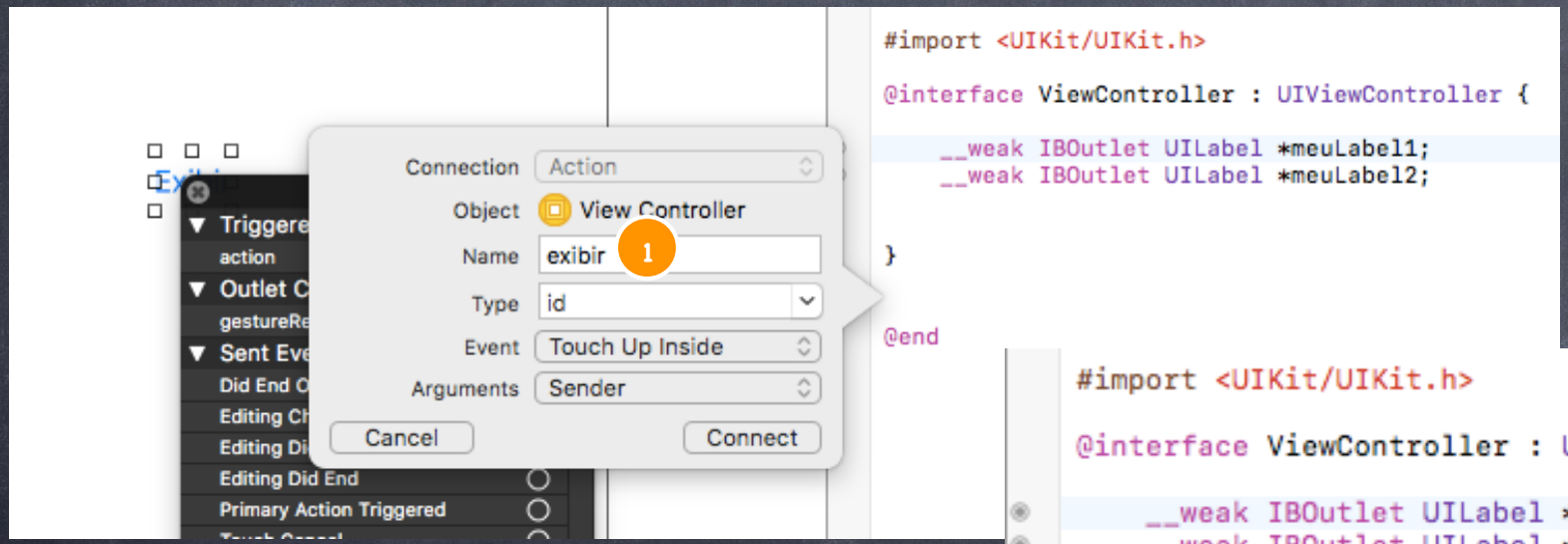
- Clique no storyboard do iPhone, clique com o botão direito do mouse sobre o Exibir, escolha o evento "Touch Up Inside" e arraste para fora da área das chaves { }.

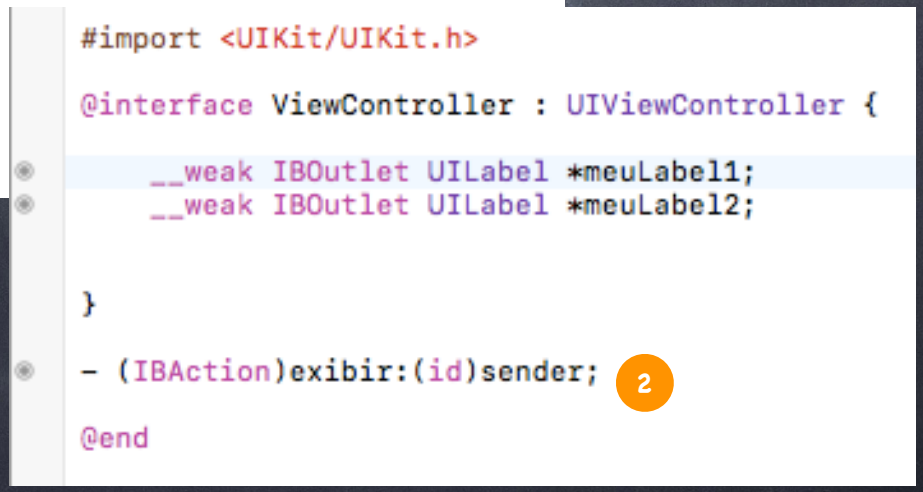


Obs: A área fora do { } no arquivo .h, é uma área para declaração dos métodos.

Declarando IBAction no .h automaticamente

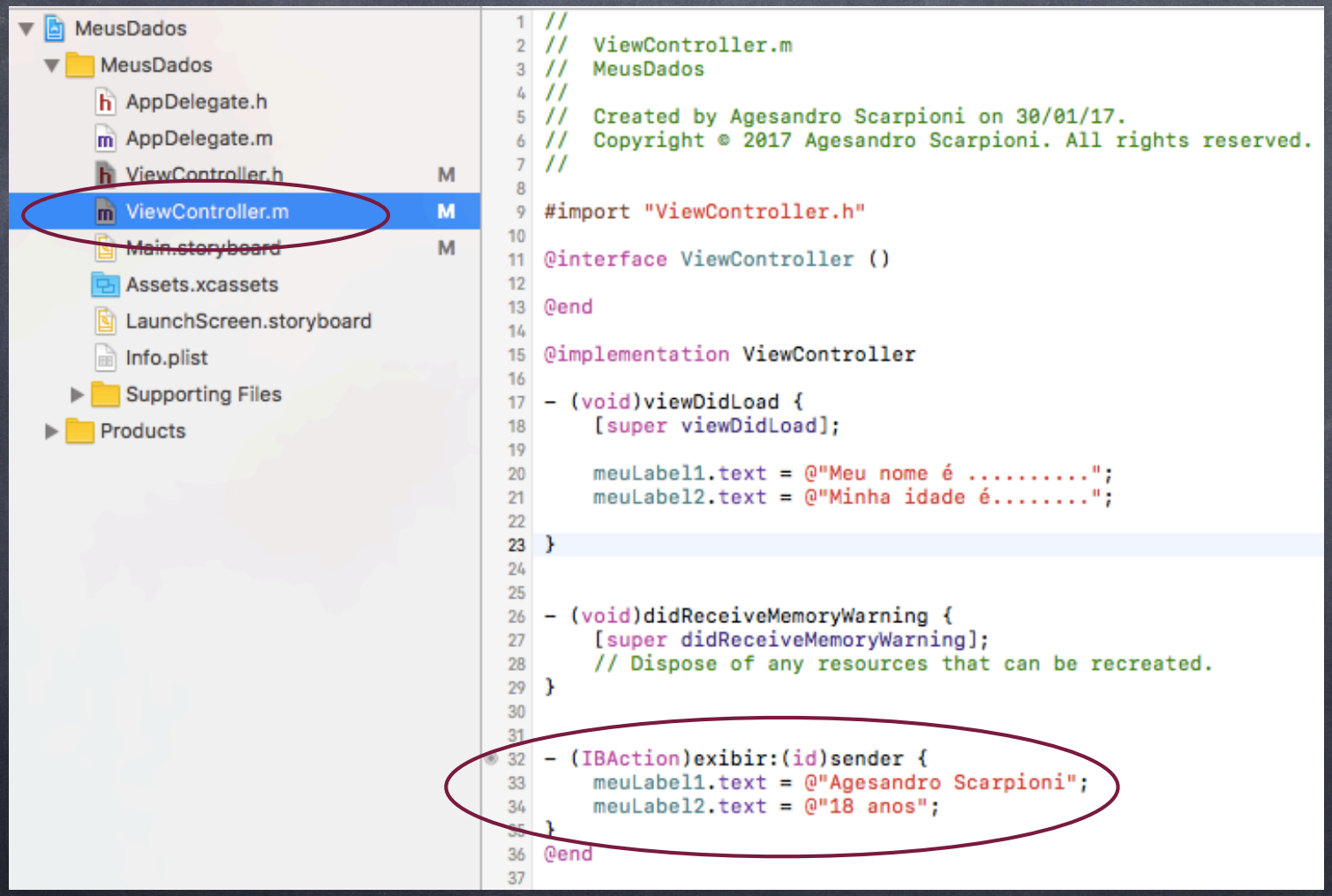
- Preencha o IBAction com o nome "Exibir" e clique em connect, a declaração irá aparecer automaticamente como mostra a figura 2 e já estará pronta para implementação no arquivo .m.





Implementando IBAction .m

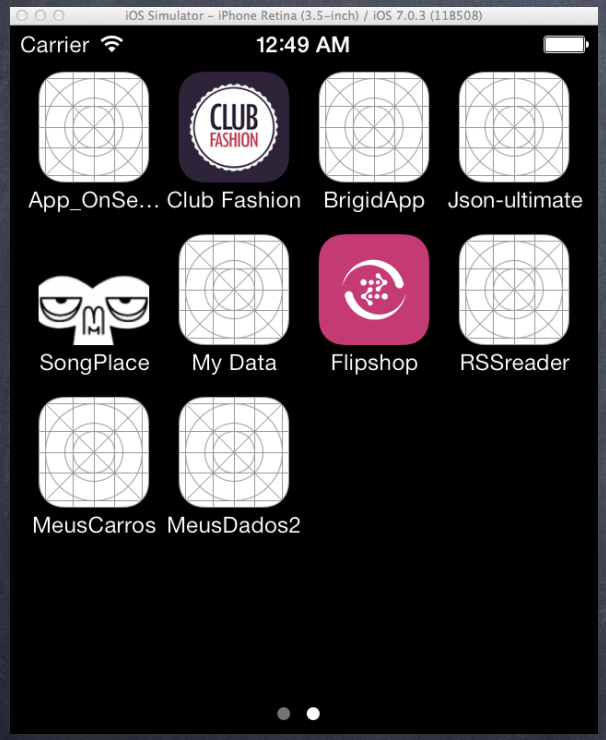
- Note que a área para a implementação do botão "Exibir" já aparece automaticamente no arquivo .m, basta escrever as 2 linhas de comando.



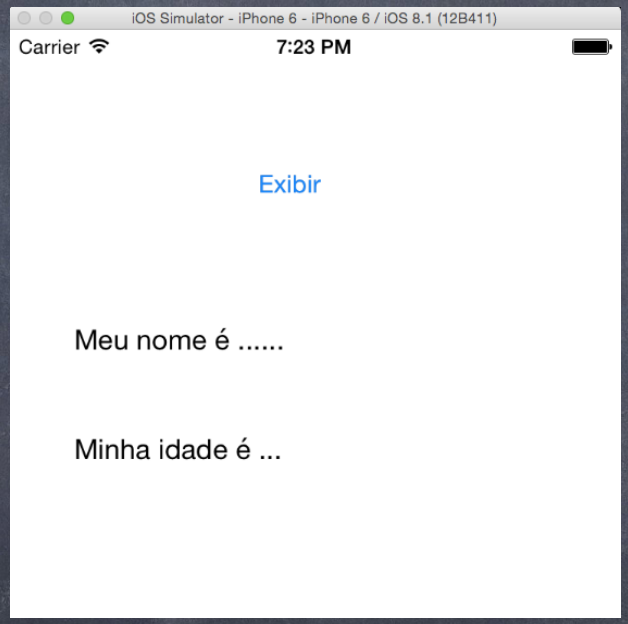
O App no simulador

FIAP

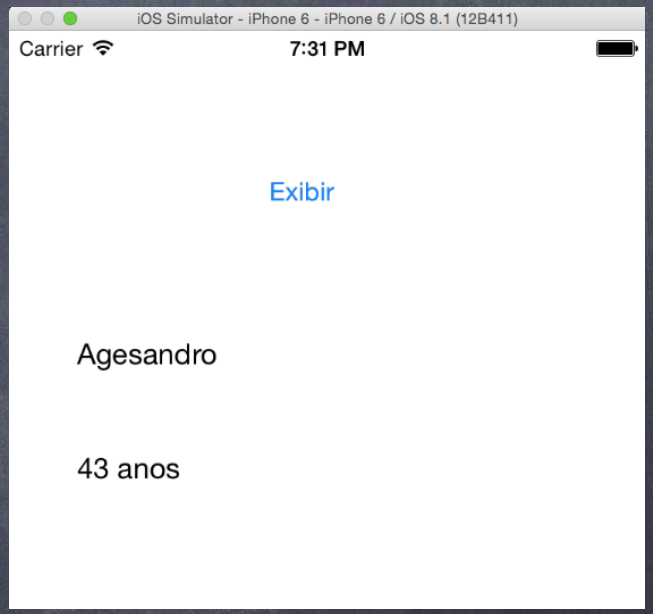
Quando executamos



Antes de clicar no botão com informações carregadas pelo viewDidLoad



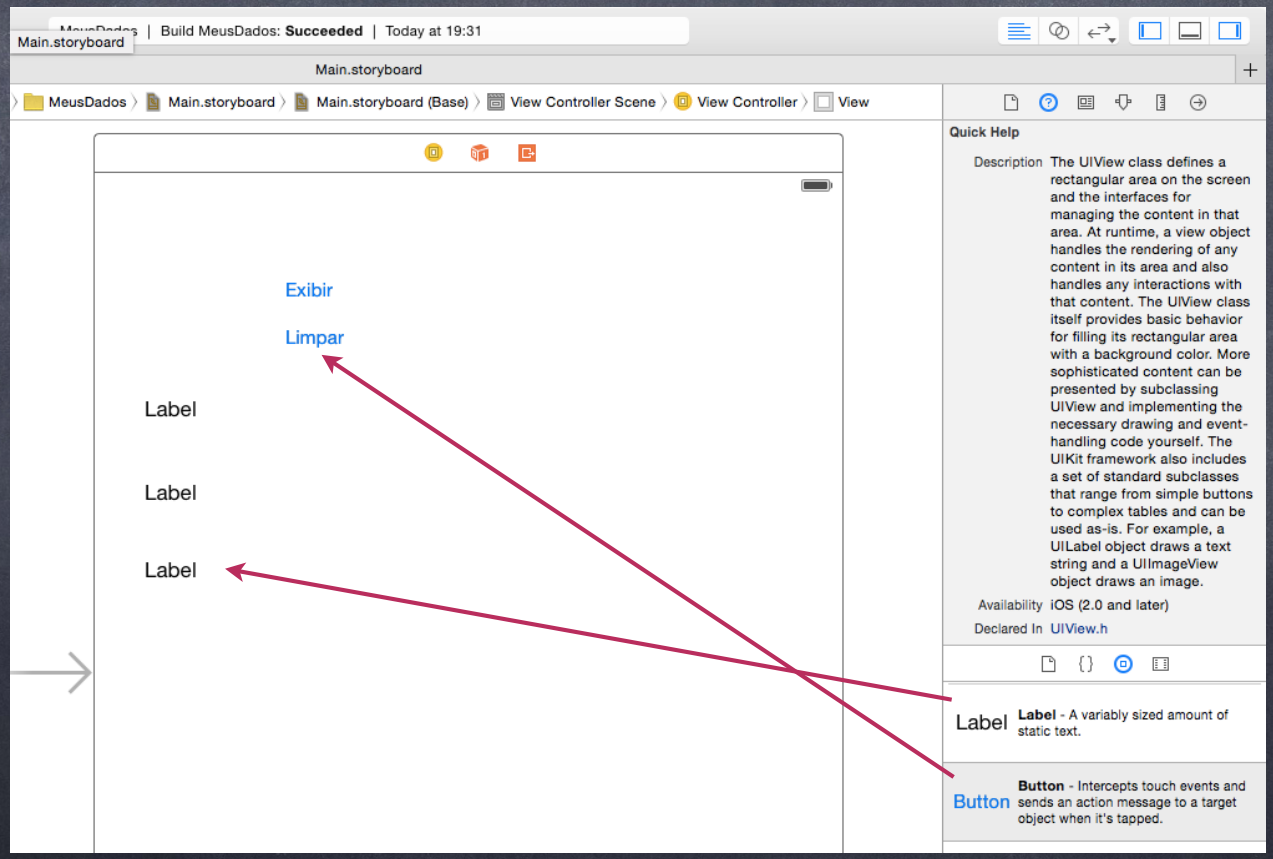
Após clicar no botão



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

Implementando IBAction e IBOutlet de forma manual

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



Implementando IBAction e IBOutlet de forma manual

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

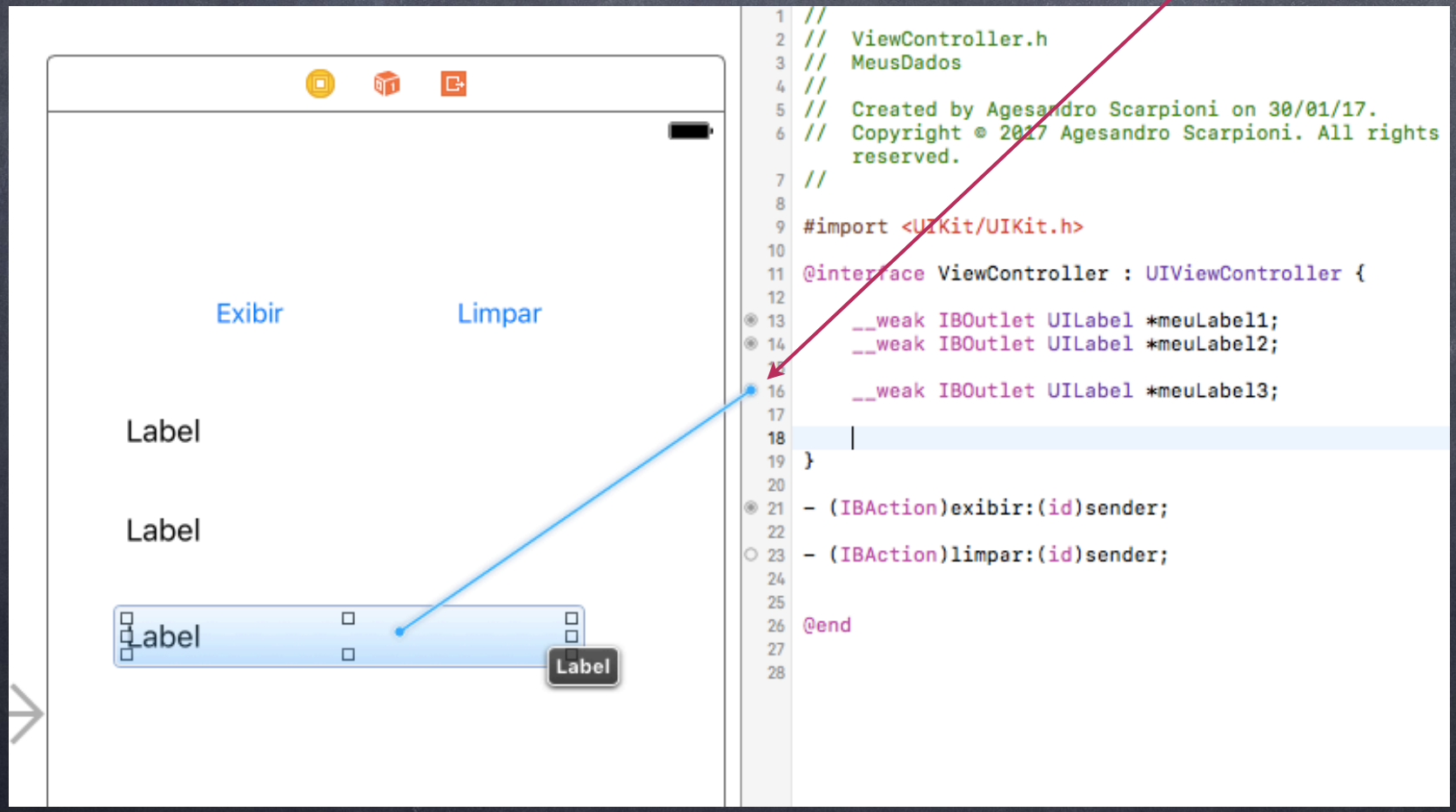
- Note que os mesmos não estão conectados com a View, observe que os círculos brancos em frente aos números das linhas não estão preenchidos.

```
1  //
2  //  ViewController.h
3  //  MeusDados
4  //
5  //  Created by Agesandro Scarpioni on 30/01/17.
6  //  Copyright © 2017 Agesandro Scarpioni. All rights
       reserved.
7  //
8
9  #import <UIKit/UIKit.h>
10
11 @interface ViewController : UIViewController {
12
13     __weak IBOutlet UILabel *meuLabel1;
14     __weak IBOutlet UILabel *meuLabel2;
15
16     IBOutlet UILabel *meuLabel3;
17
18 }
19
20 - (IBAction)exibir:(id)sender;
21
22 - (IBAction)limpar:(id)sender;
23
24
25 @end
26
27
```

Dica: Para exibir os números de linha clique em Xcode -> Preferences, ou (Command + ,) e em Text Editing, clique no primeiro check box "Show Line Numbers".

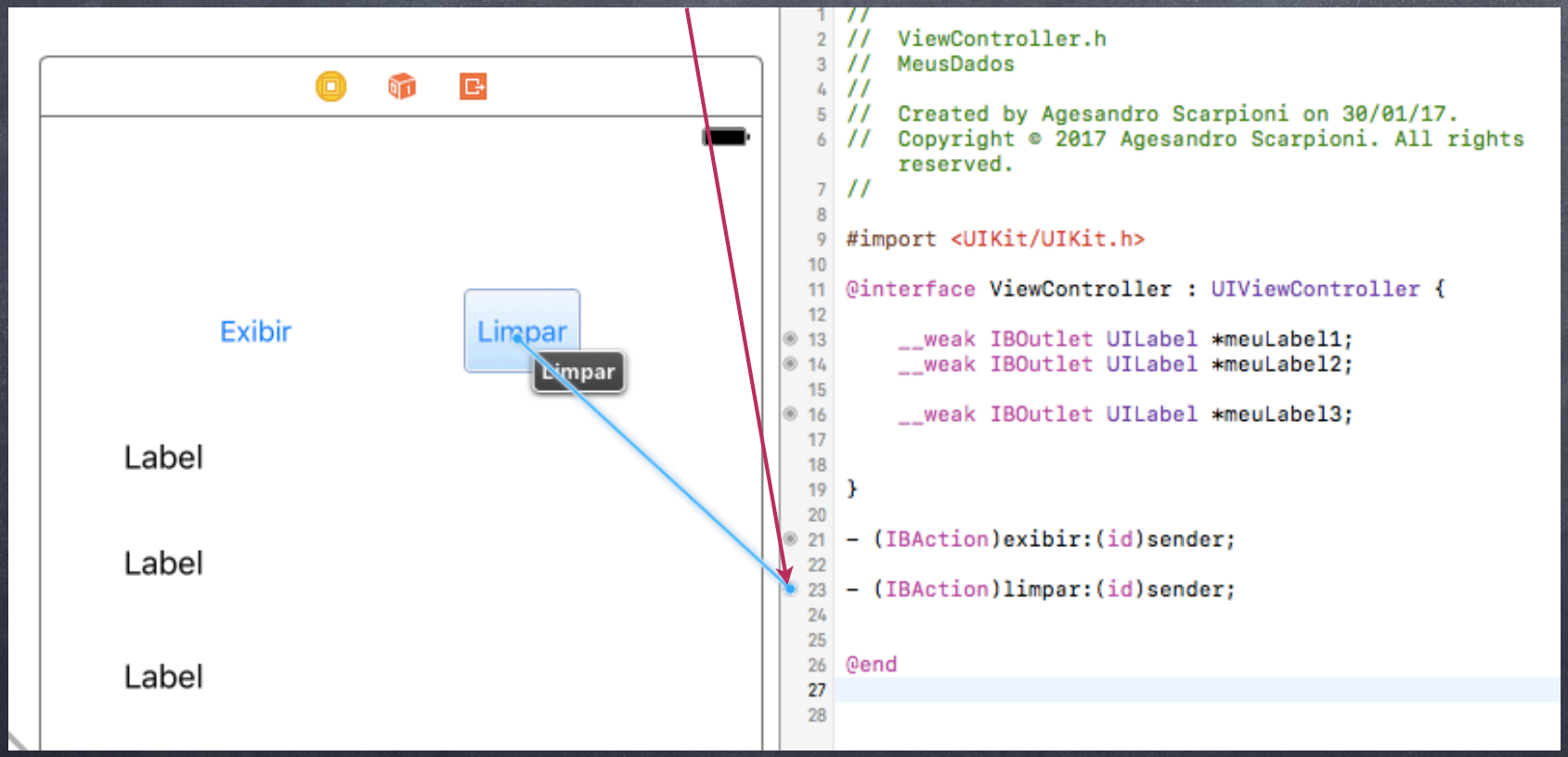
Implementando IBAction e IBOutlet de forma manual

- 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 IBOutlet de forma manual

- Note que após essas ligações nosso novo IBOutlet 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
       reserved.
7  //
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 IBOutlet de forma manual

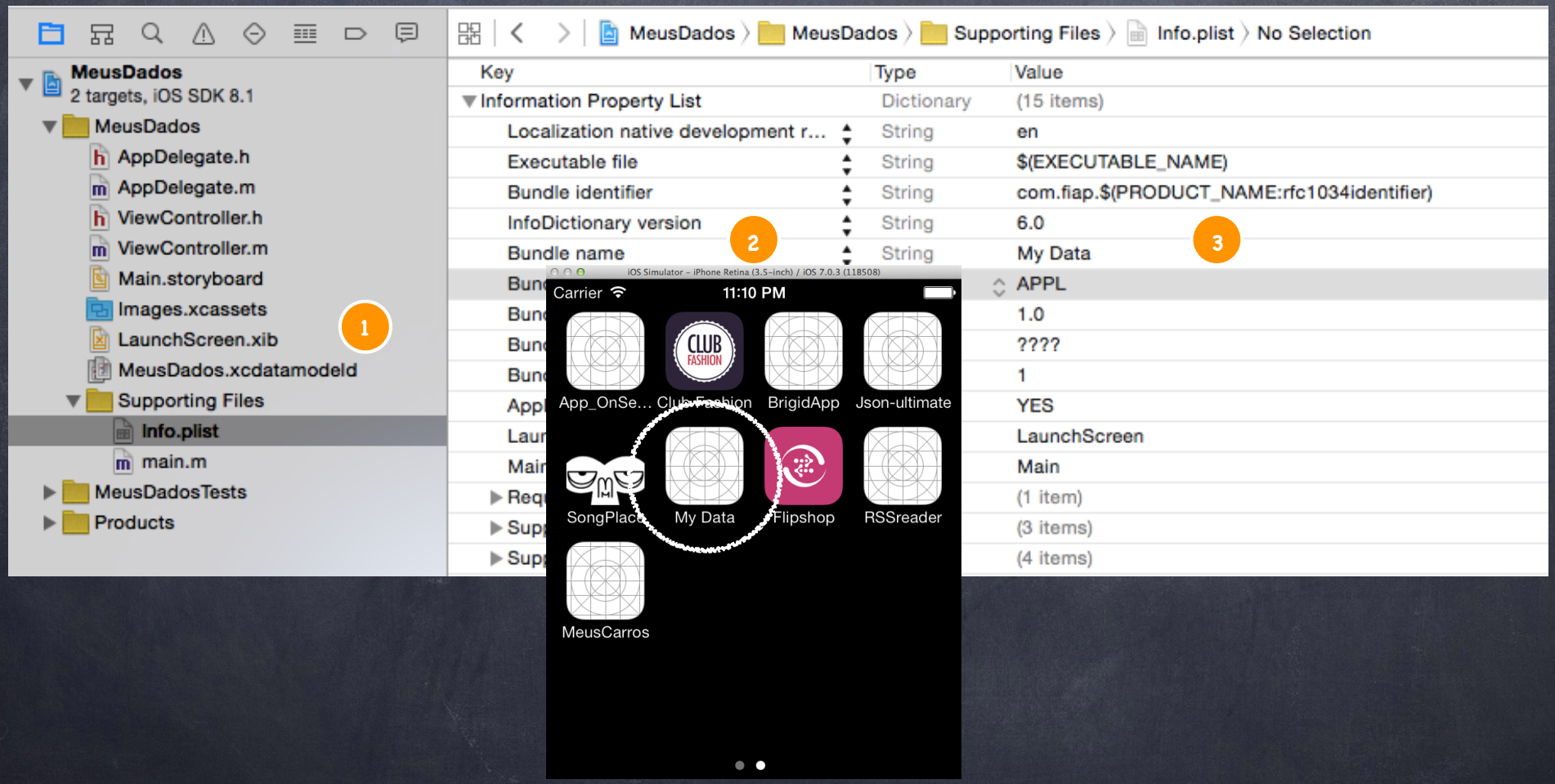
- Inclua as linhas do Label3 no DidLoad e no Exibir do arquivo .m.
- Copie a linha do IBAction Limpar do arquivo ViewController.h e cole no ViewController.m, retire o (;) e coloque as chaves ({}), após essa etapa copie as linhas do DidLoad e cole no IBAction Limpar.
- Outra possibilidade no limpar é chamar o método viewDidLoad.

```
- (IBAction)limpar:(id)sender {
//      meuLabel1.text = @"";
//      meuLabel2.text = @"";
//      meuLabel3.text = @"";
    [self viewDidLoad];
}
```

```
//
//  ViewController.m
//  MeusDados
//
//  Created by Agesandro Scarpioni on 30/01/17.
//  Copyright © 2017 Agesandro Scarpioni. All rights reserved.
//

#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    meuLabel1.text = @"Meu nome é ..........";
    meuLabel2.text = @"Minha idade é........";
    meuLabel3.text = @"Minha cidade é ......";
}

- (IBAction)limpar:(id)sender{
    meuLabel1.text = @"";
    meuLabel2.text = @"";
    meuLabel3.text = @"";
}


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


- (IBAction)exibir:(id)sender {
    meuLabel1.text = @"Agesandro Scarpioni";
    meuLabel2.text = @"18 anos";
    meuLabel3.text = @"São Palo";
}
@end
```

Como trocar o nome do App

- Encontre o arquivo Info.plist e altere o campo Bundle name (2), digite My Data(3), veja o próximo slide.



Dica: Shif + Command + H —> Aciona o botão Home no Simulador

Alterando o Ícone do App

- Clique em seu projeto (1), clique na seta do passo 2

MeusDados 〉 iPhone 7 Plus | MeusDados | Build MeusDados: **Succeeded** | Today at 10:58

1

MeusDados
- MeusDados
  - AppDelegate.h
  - AppDelegate.m
  - ViewController.h M
  - ViewController.m M
  - Main.storyboard M
  - Assets.xcassets M
  - LaunchScreen.storyboard
  - Info.plist M
  - Supporting Files
- Products

General | Capabilities | Resource Tags | Info | Build Settings | Build Phases

PROJECT
- MeusDados

TARGETS
- MeusDados

Identity

Display Name: My Data
Bundle Identifier: com.scarpioni.MeusDados
Version: 1.0
Build: 1

Signing

No accounts found
Add a developer account to sign your app.
Add Account...

Deployment Info

Deployment Target: 10.2
Devices: iPhone
Main Interface: Main
Device Orientation:
- [x] Portrait
- [ ] Upside Down
- [x] Landscape Left
- [x] Landscape Right

Status Bar Style: Default
- [ ] Hide status bar
- [ ] Requires full screen

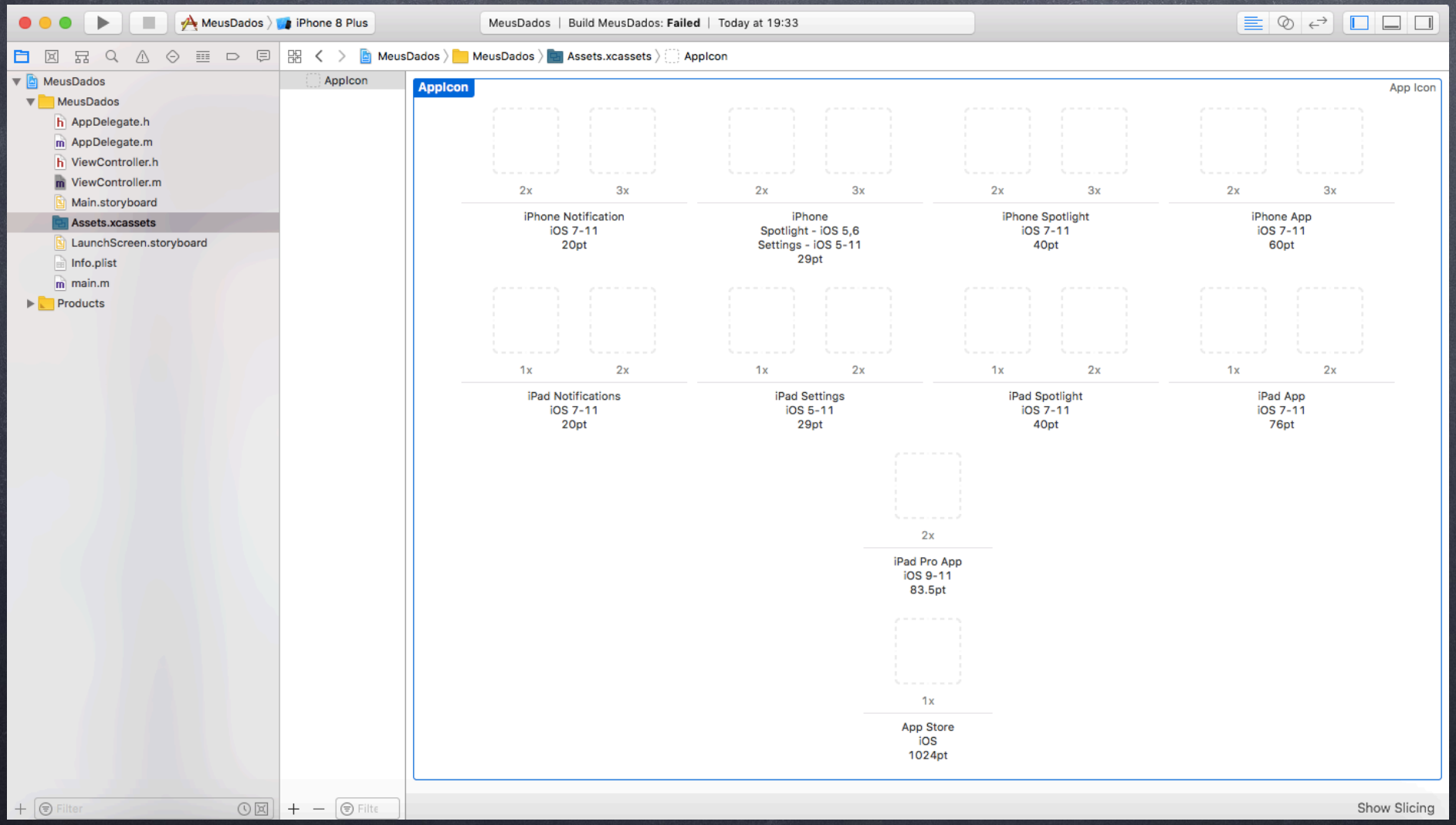
App Icons and Launch Images

App Icons Source: AppIcon

2

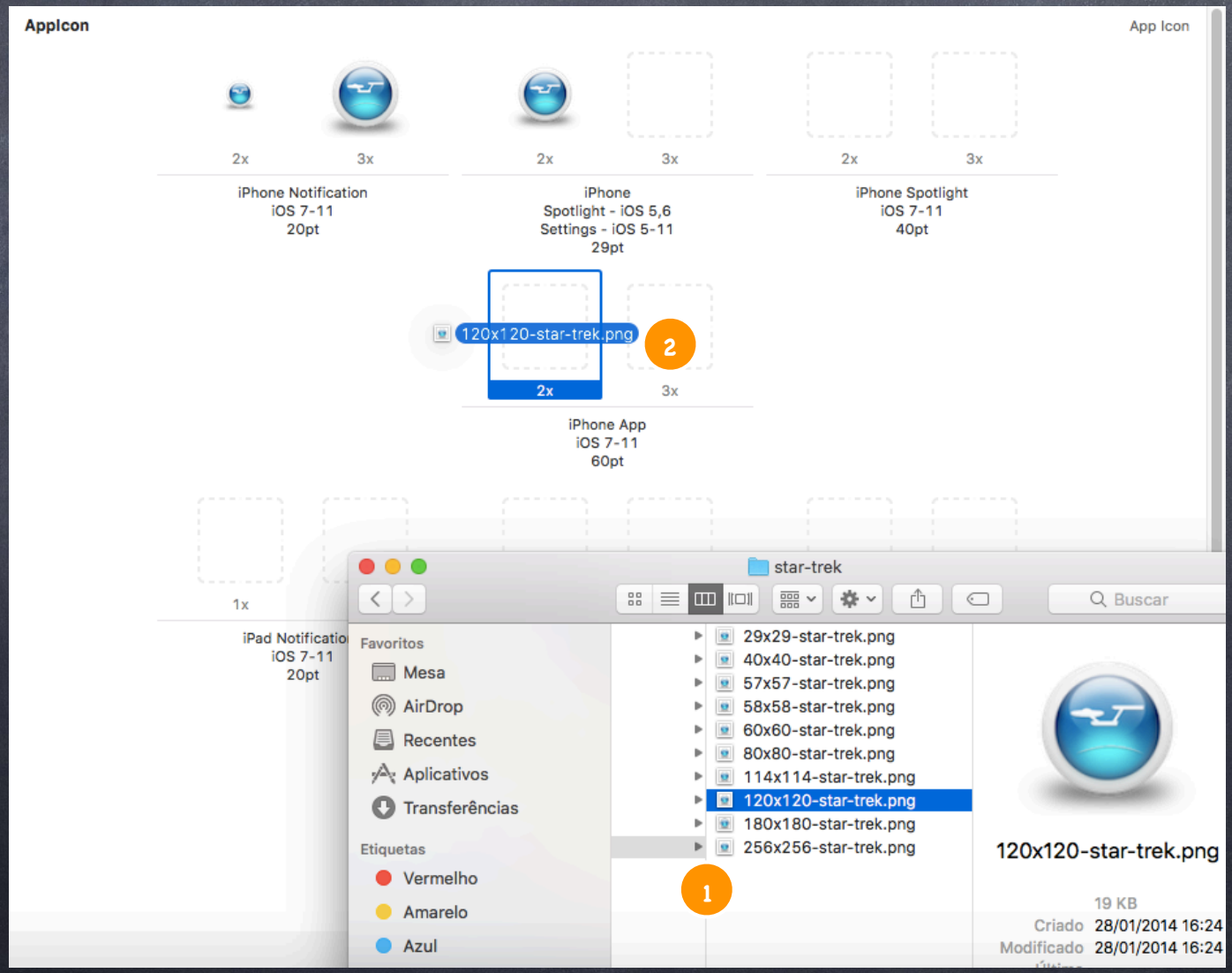
Alterando o Ícone do App

- 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.



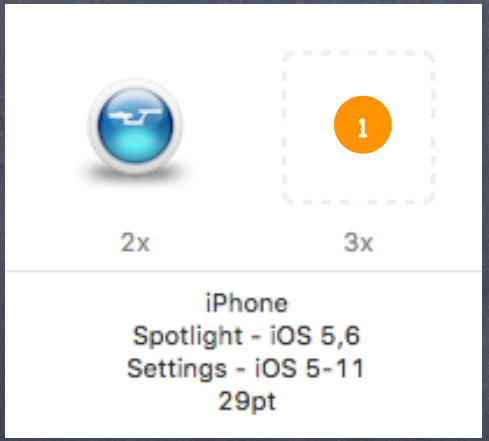
Alterando o Ícone do App

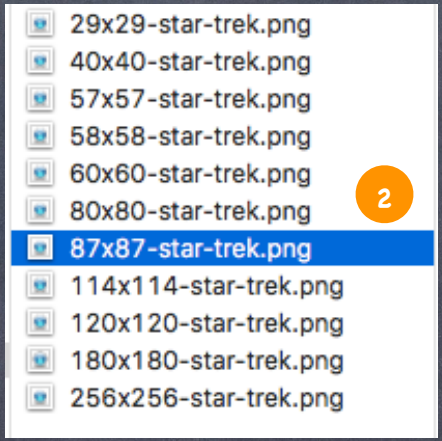
- Abra uma das pastas disponibilizadas pelo professor com o finder e arraste a imagem apropriada para o espaço exclusivo da imagem escolhida, exemplo: A imagem 120x120 entra no 2x - 60pt, passos (1) e (2).

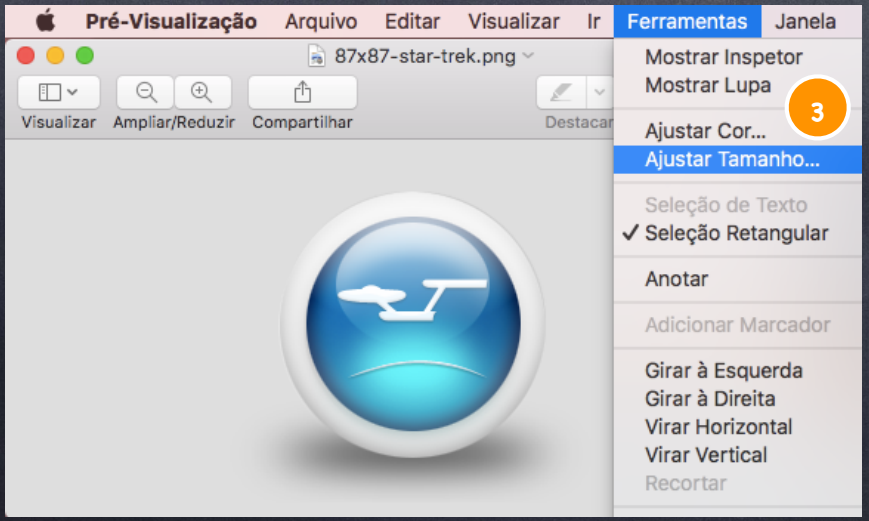


Ajustando um ícone

- Note que para a imagem 3x 29pt, precisamos de uma imagem 87x87 (1) que no momento não temos ainda, uma possibilidade é duplicar a imagem 260x260 e renomear para 87x87 (2), dê um duplo clique na imagem 87x87 dessa forma irá abrir a pré-visualização, clique no menu ferramentas -> Ajustar tamanho (3), troque o tipo de imagem para pixels (4), altere 260x260 para 87x87 (5), clique em Ok e feche a Pré-Visualização a imagem é salva automaticamente, pronto você já tem um ícone 87x87.

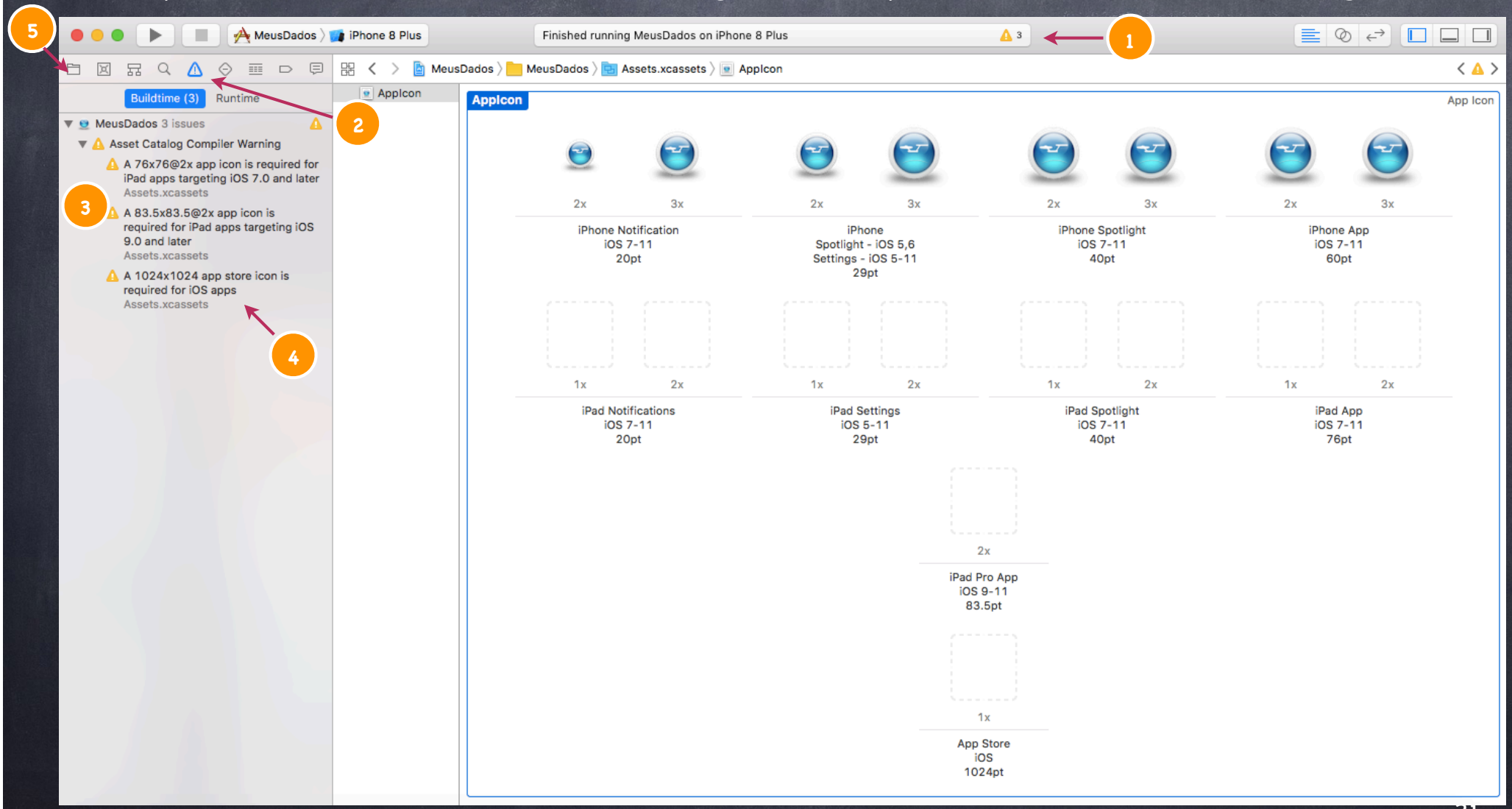






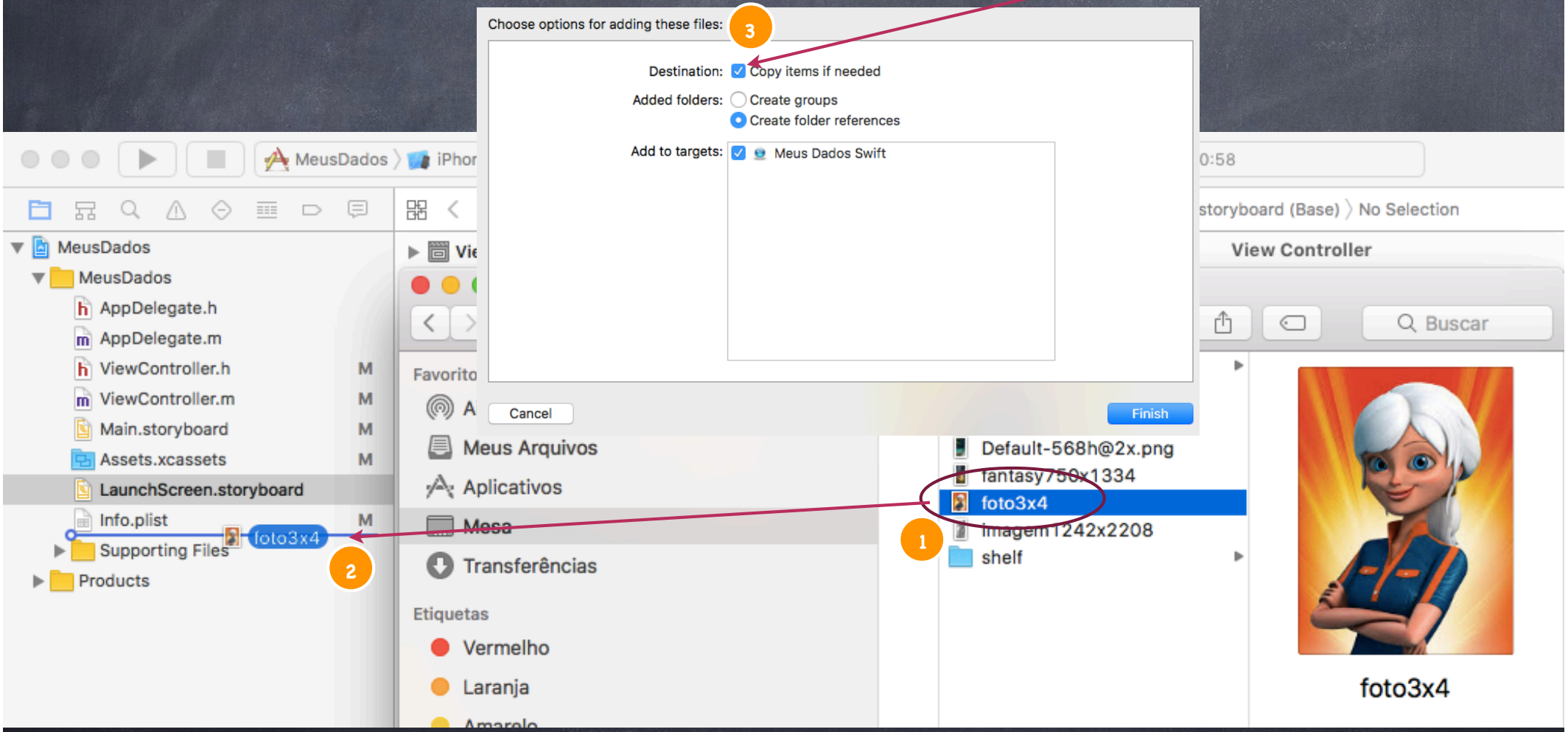
Warning's

- Se você não errou nenhum tamanho irá aparecer apenas 3 Warning's (1), clique em Show the Issue navigator (2), aqui você verá que os avisos aparecem pela falta dos ícones de iPad (3) e o ícone da Apple Store (4), isso não impede de seu App funcionar, você poderá cuidar disso mais tarde, agora volte para Show the Project navigator (5).



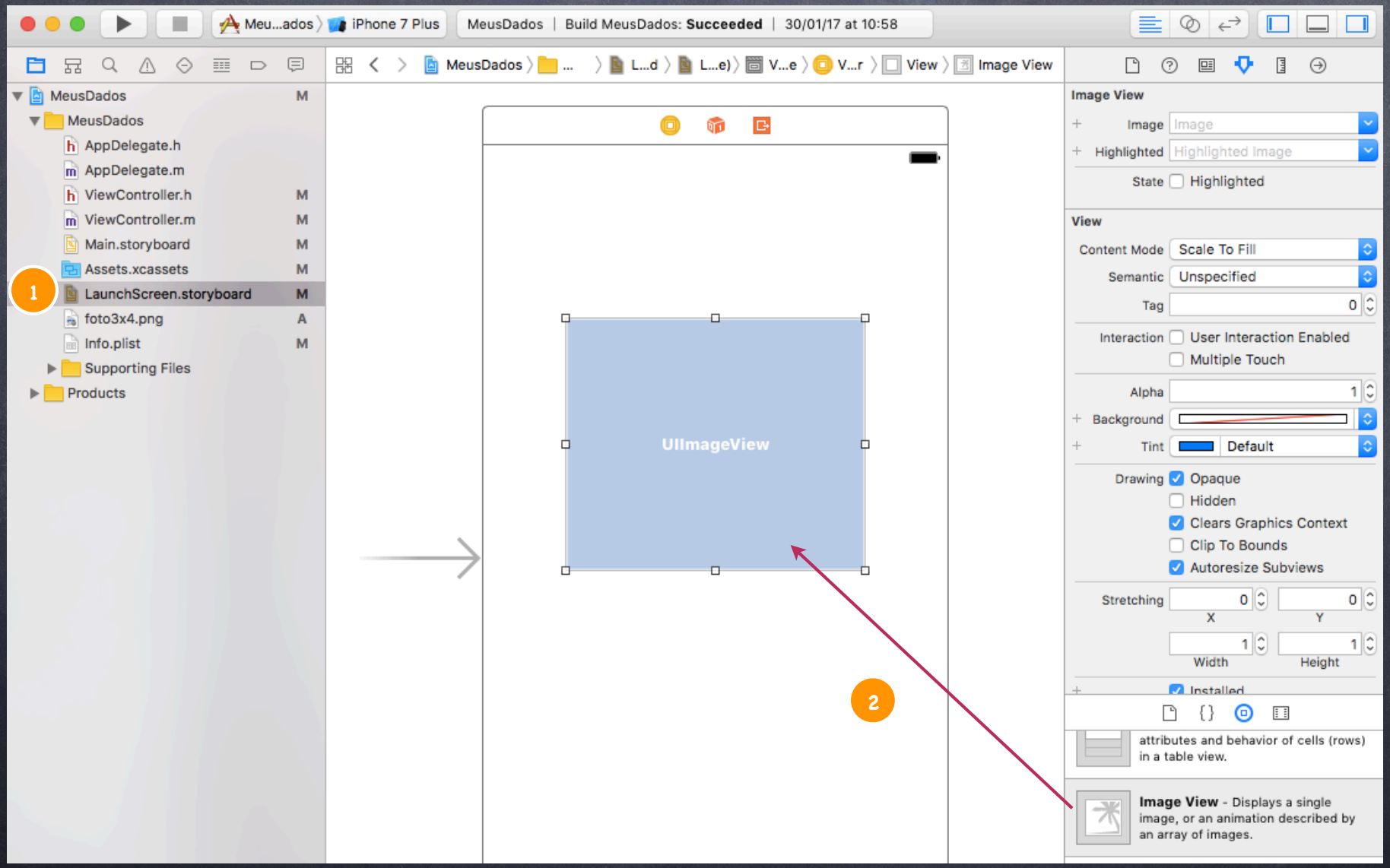
Splash Screen - Forma 1

- Abra a pasta de imagem disponibilizada e arraste a imagem para o projeto, você pode se preferir criar uma pasta de imagens e coloca-las todas em uma única pasta, faça uma cópia da imagem para o projeto clicando aqui.



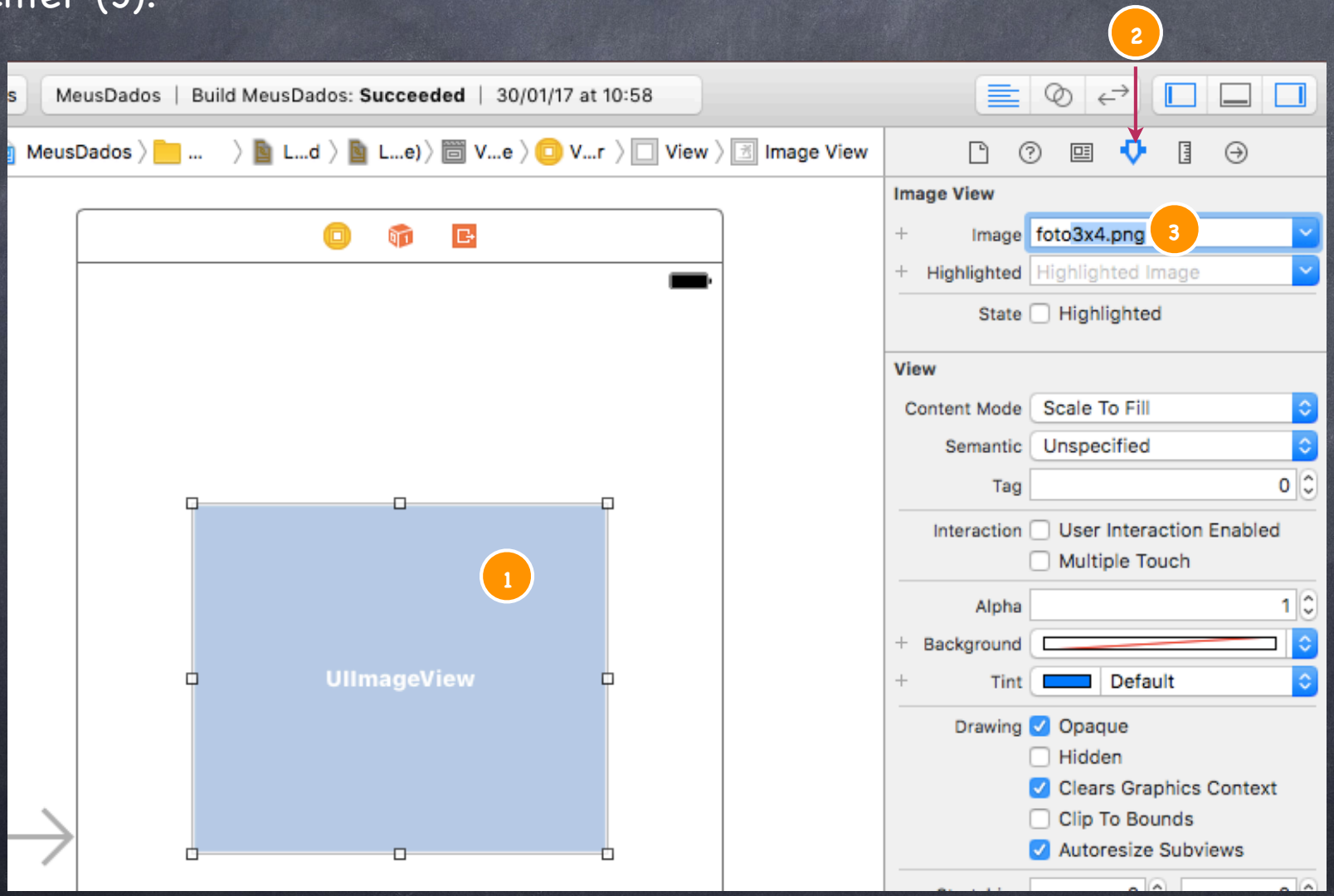
Splash Screen - Forma 1

- Clique no LaunchScreen.storyboard (1), arraste um Image View para a View (2)



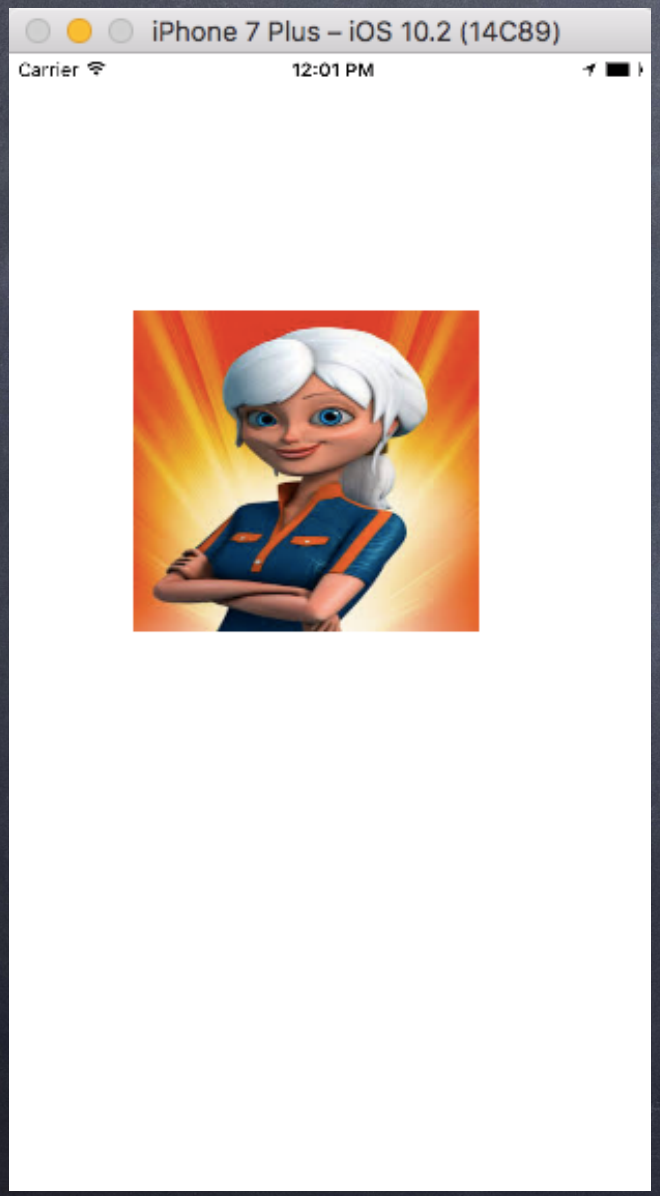
Splash Screen - Forma 1

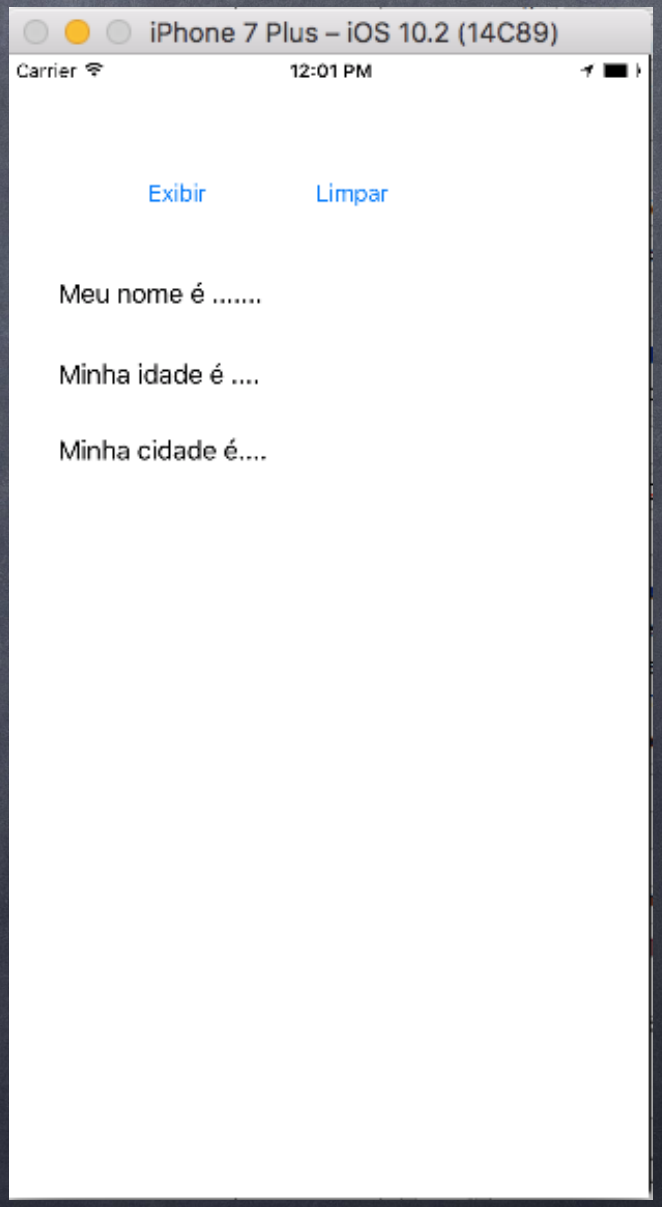
- Selecione a imagem (1), em Attributes Inspector (2), digite o nome da imagem e dê um enter (3).



Splash Screen - Forma 1

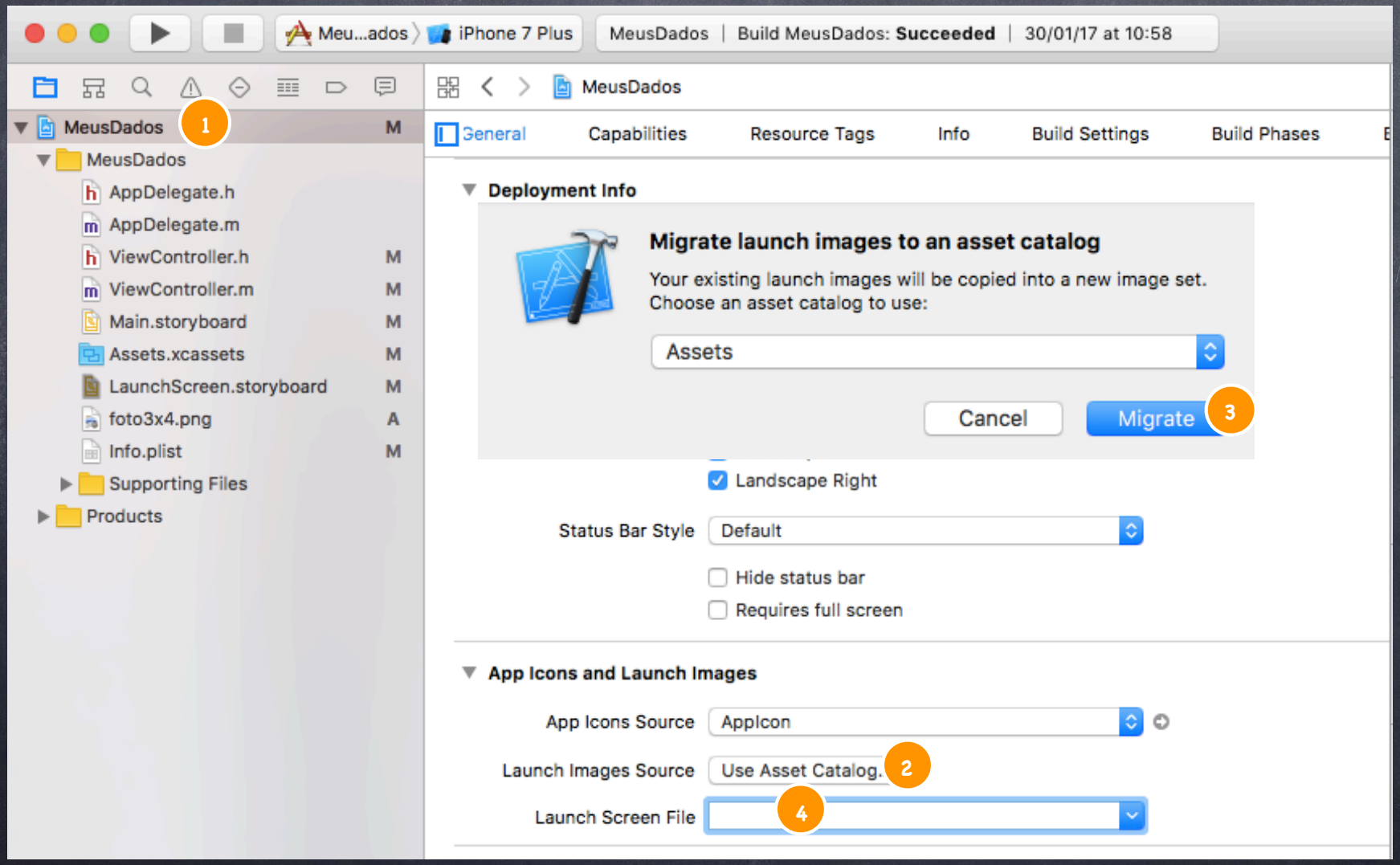
- Execute (Cmd + R).





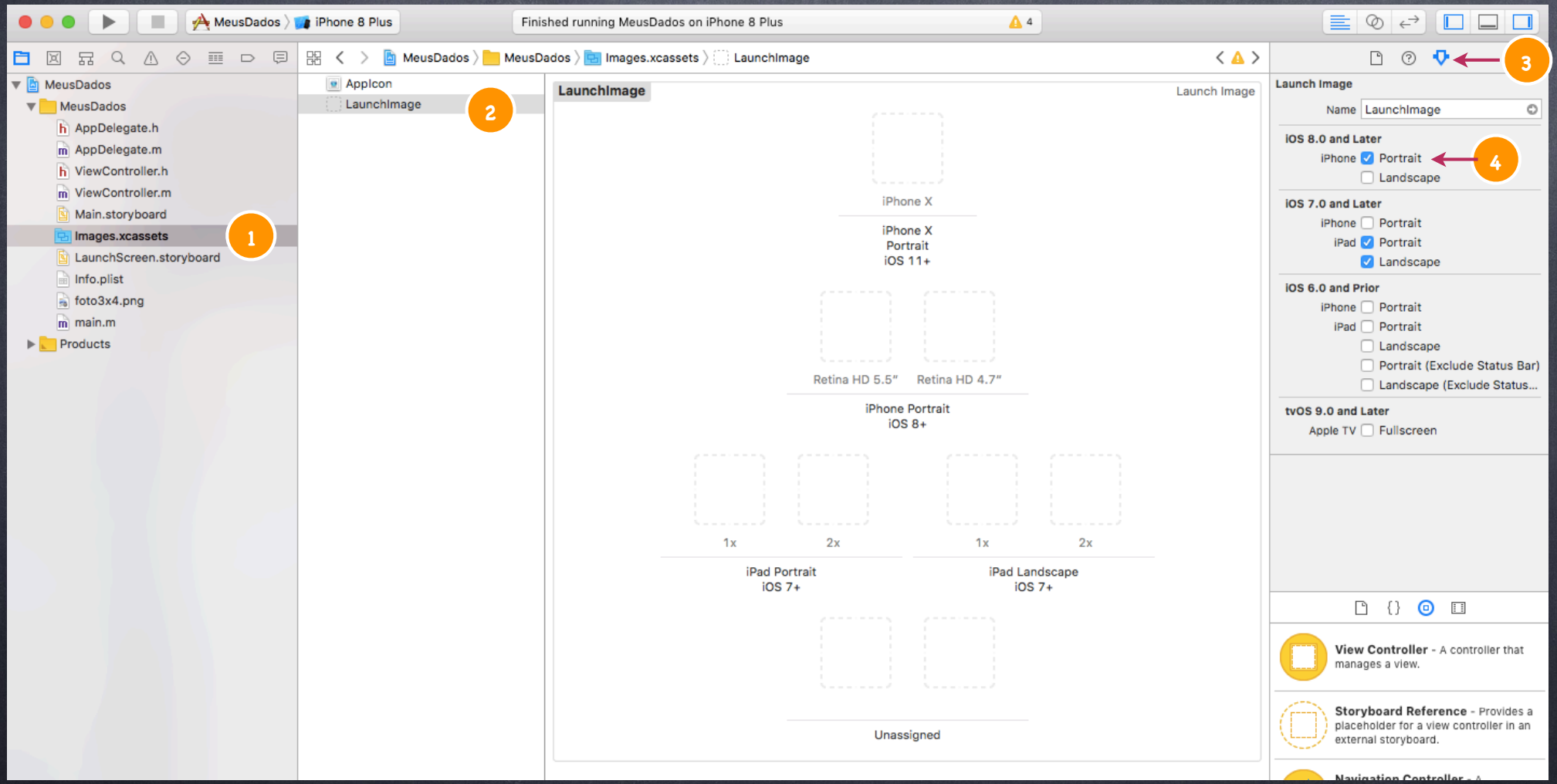
Splash Screen - Forma 2

- Clique no Projeto (1) depois clique em Use Asset Catalog (2), ao aparecer a caixa de diálogo escolha Migrate (3), para não ser utilizado o LauncScreen.storyboard com a imagem anterior apague o item (4), não esqueça de apertar a tecla return/enter.



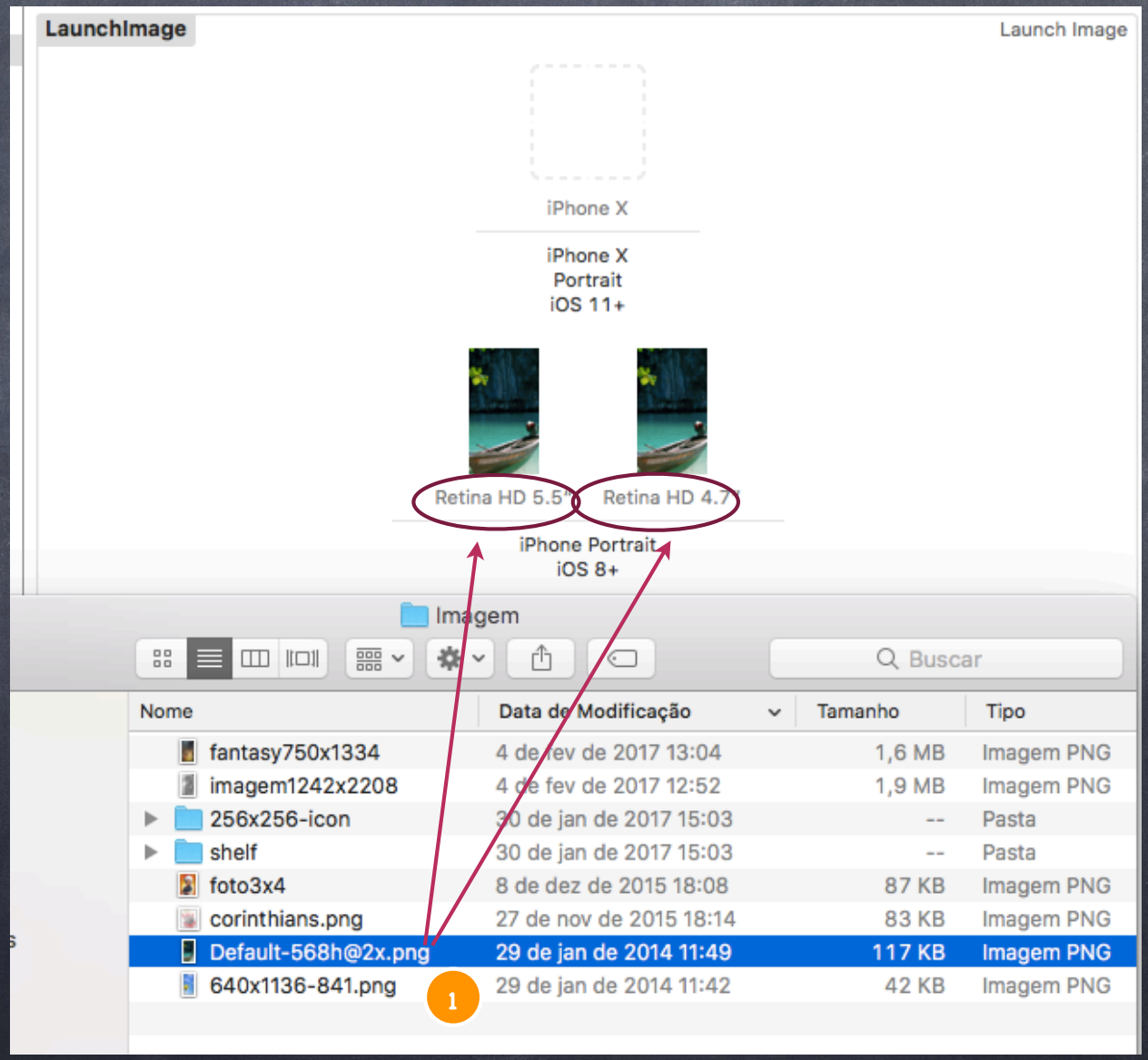
Splash Screen - Forma 2

- Clique em Assets.xcassets (1), clique em LaunchImage (2), clique em Attributes Inspector (3), depois clique no CheckBox iOS 8.0 and Later iPhone Portrait (4).



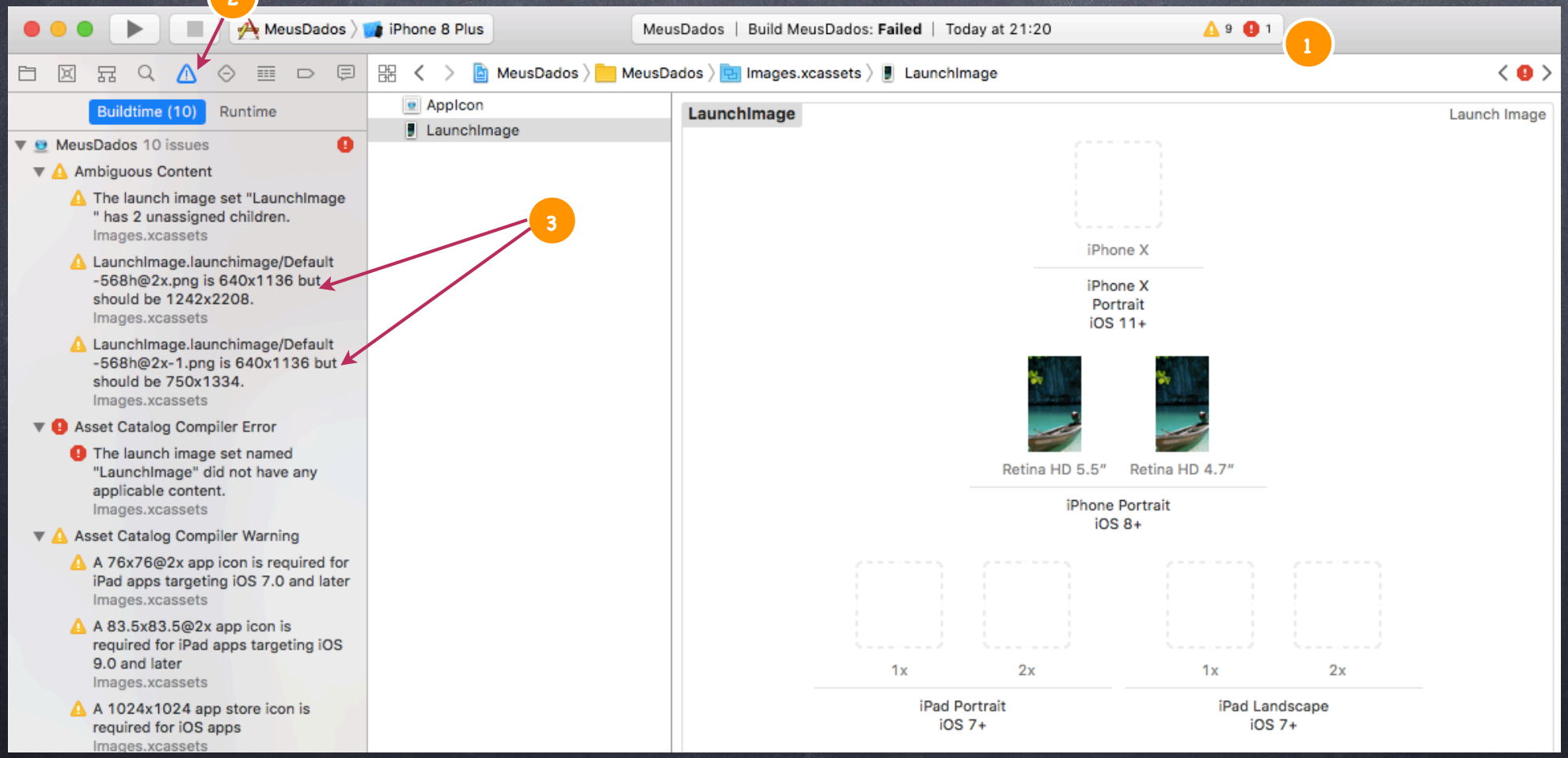
Splash Screen - Forma 2

- Arraste a imagem disponibilizada chamada Default-568h@2x.png para cada imagem do iPhone Portrait (1).



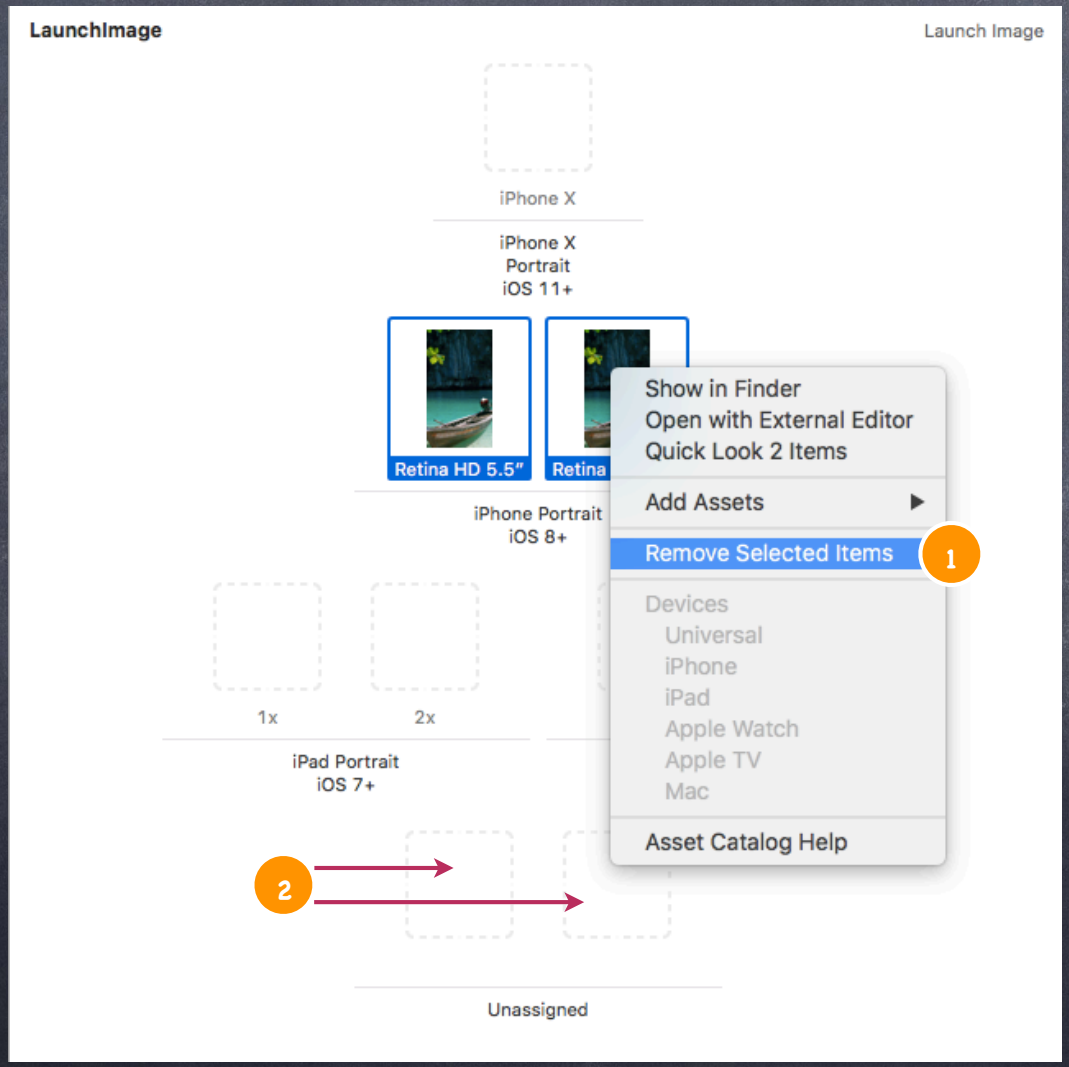
Splash Screen - Forma 2

- Execute seu programa - Command + R, note que irá aparecer erro de compilação (1), isso ocorre porque as imagens utilizadas não estão no padrão adequado (2), para Iphone 6 Plus, 6s Plus e 7 Plus que é 1242x2208 e 750x1334 para os Iphones 6 e 6s em diante (3)



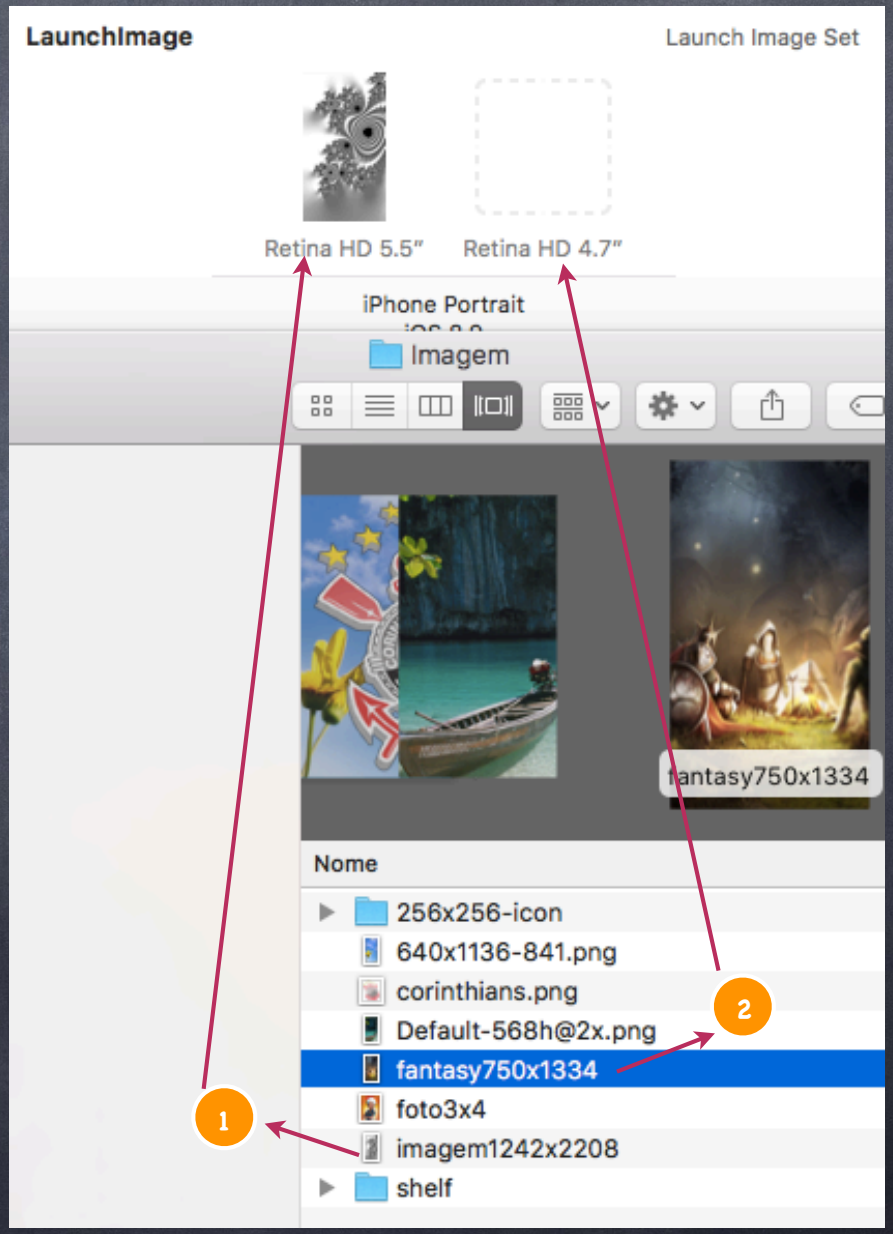
Splash Screen - Forma 2

- Para resolver esse problema selecione as duas imagens, clique com o botão direito e escolha Remove Selected Items (1), aproveite para remover os dois templates logo abaixo. (2)



Splash Screen - Forma 2

- Arraste a imagem1242x2208. para a retina HD 5.5'' (1) em seguida repita o processo arrastando a imagem Fantasy750x1334 para o espaço Retina HD 4.7'' (2).



Splash Screen - Forma 2

- Execute o programa para iPhone 7Plus e para Iphone 7 e note que aparecerá telas distintas de Spplash Screen

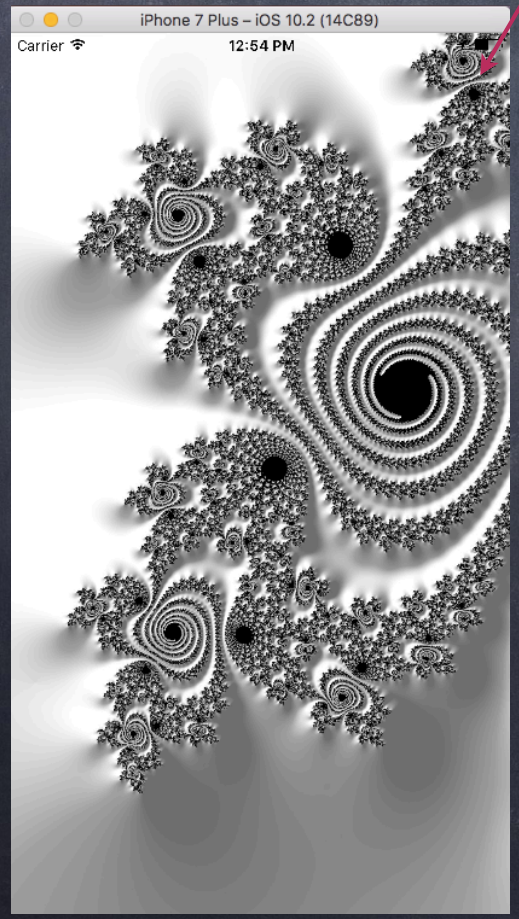


Imagem de splash screen

- 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.

FIAP

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

FIAP

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)

FIAP

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.

FIAP

Veja os Links para Gerar Icons e Telas Splash

- http://makeappicon.com
- http://ticons.fokkezb.nl

Utilizando break points

FIAP

- Ao clicar nessa área é inserido um break point - Azul escuro break point ativo

- Ao clicar novamente em um break point ativo, ele fica azul claro e se torna inativo.

- Para apagar todos os break point's, use esse atalho e com o botão direito na classe de seu interesse, escolha delete break point.

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

FIAP

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

- Novos objetos em novas interfaces.