

TableViewController

Parte 2

X-Code - Obj-C

Prof. Agesandro Scarpioni

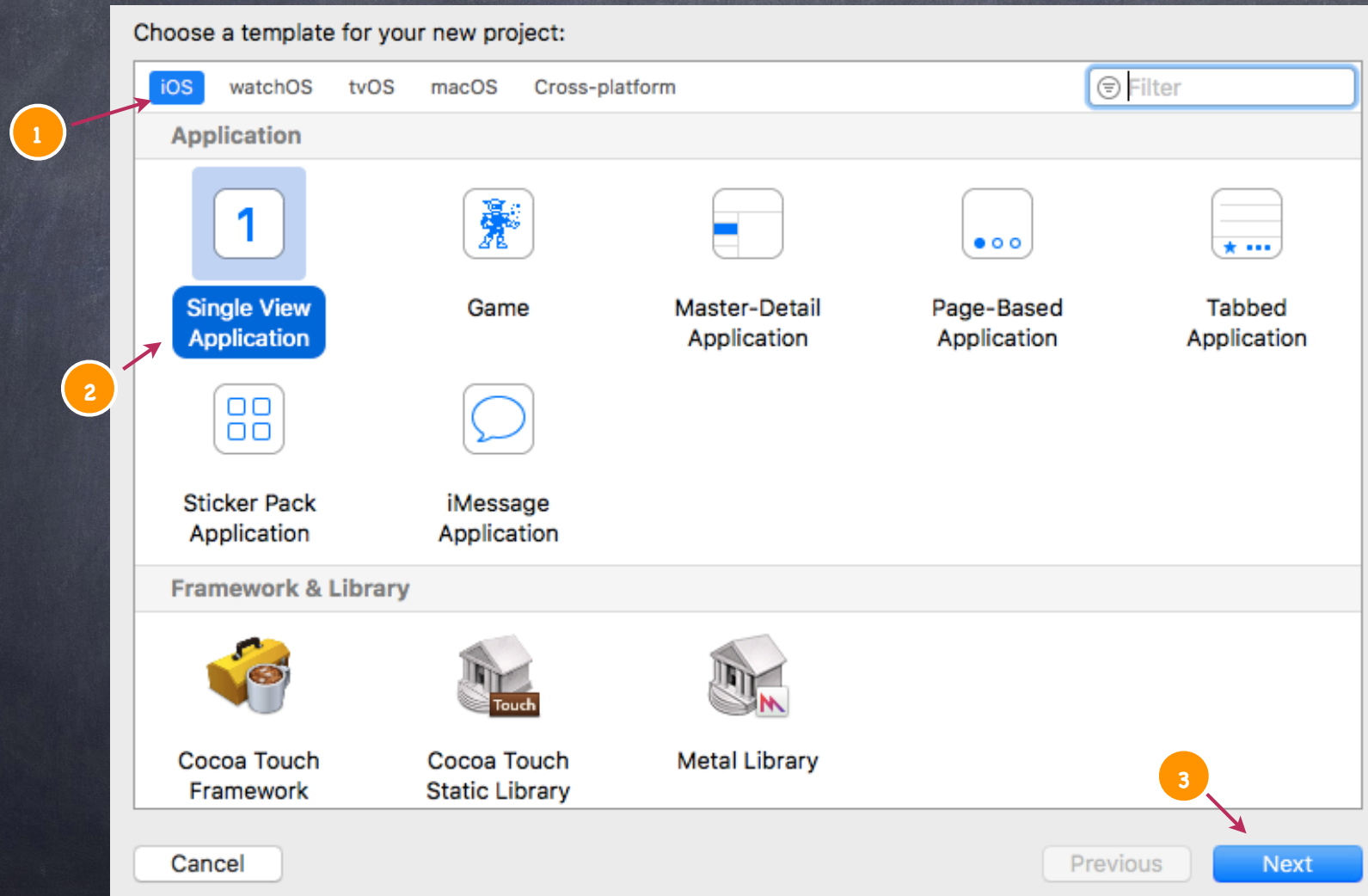
agesandro@fiap.com

TableViewController

- O TableView é um objeto desenvolvido para exibir e manipular listas, neste conjunto de slides será utilizado uma segunda forma para se implementar o TableView.
- Será criado um novo controller, será implementado um método para a exclusão dos itens e ainda uma classe com um método estático para exibir mensagens.

TableViewController

- Vamos criar um projeto novo do tipo IOS application(1), Single View Application(2) clique em Next(3).



TableViewController

- Nomeie o projeto como: "Exemplo2_TableView_OBJC" (4), escolha a linguagem Objective-C(5) e o device iPhone(6).

Choose options for your new project:

4 → Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

5 → Language:

6 → Devices:

☐ Use Core Data

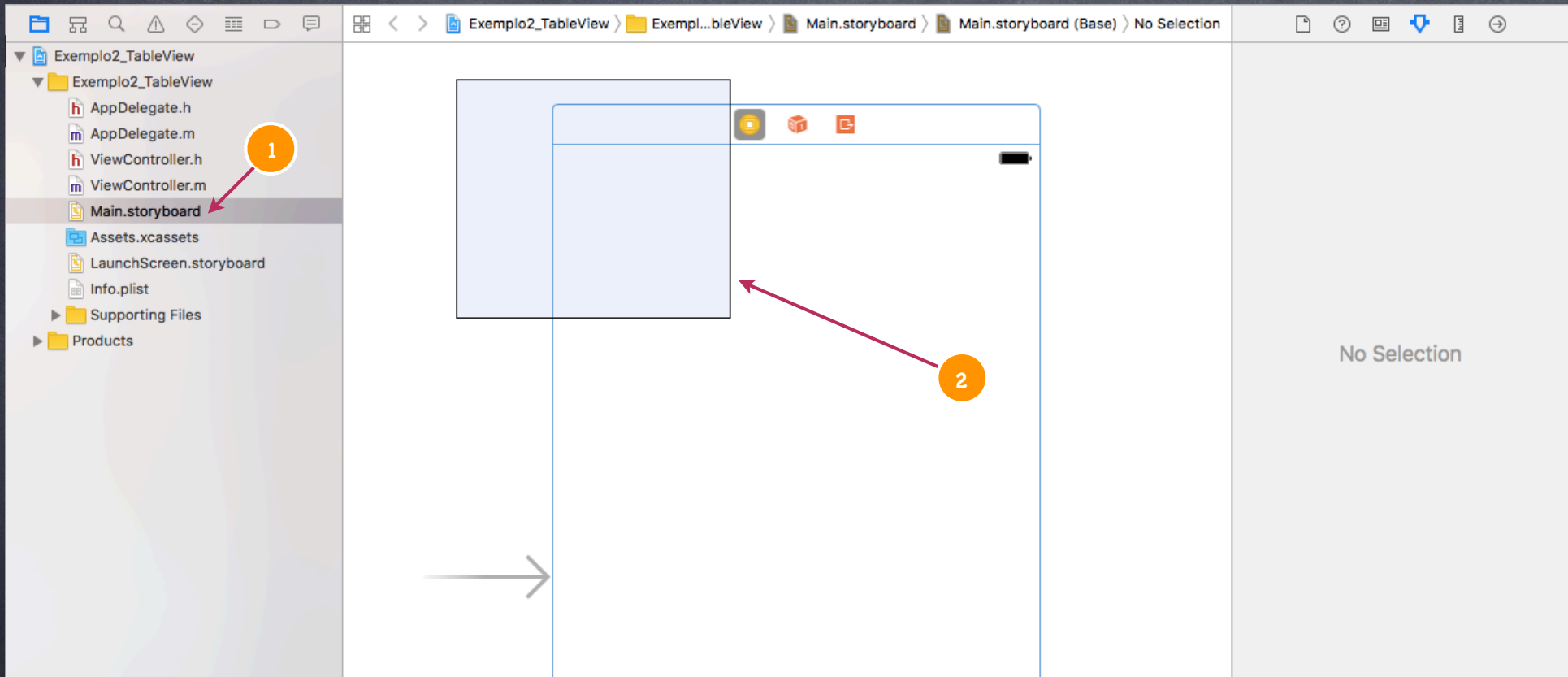
☐ Include Unit Tests

☐ Include UI Tests

TableViewController

FIAP

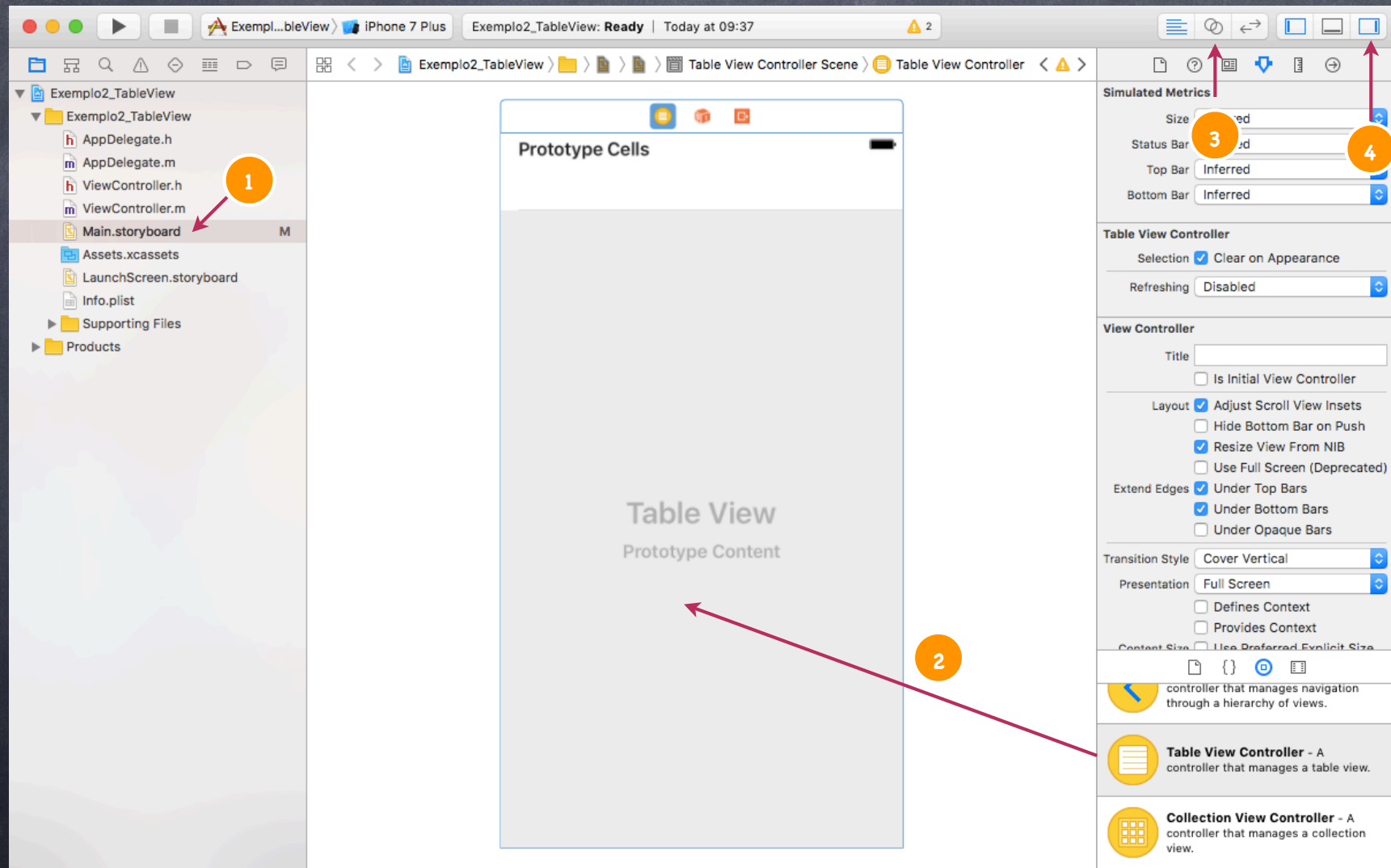
- No MainStoryboard(1) selecione a View (2) e a apague.



TableViewController

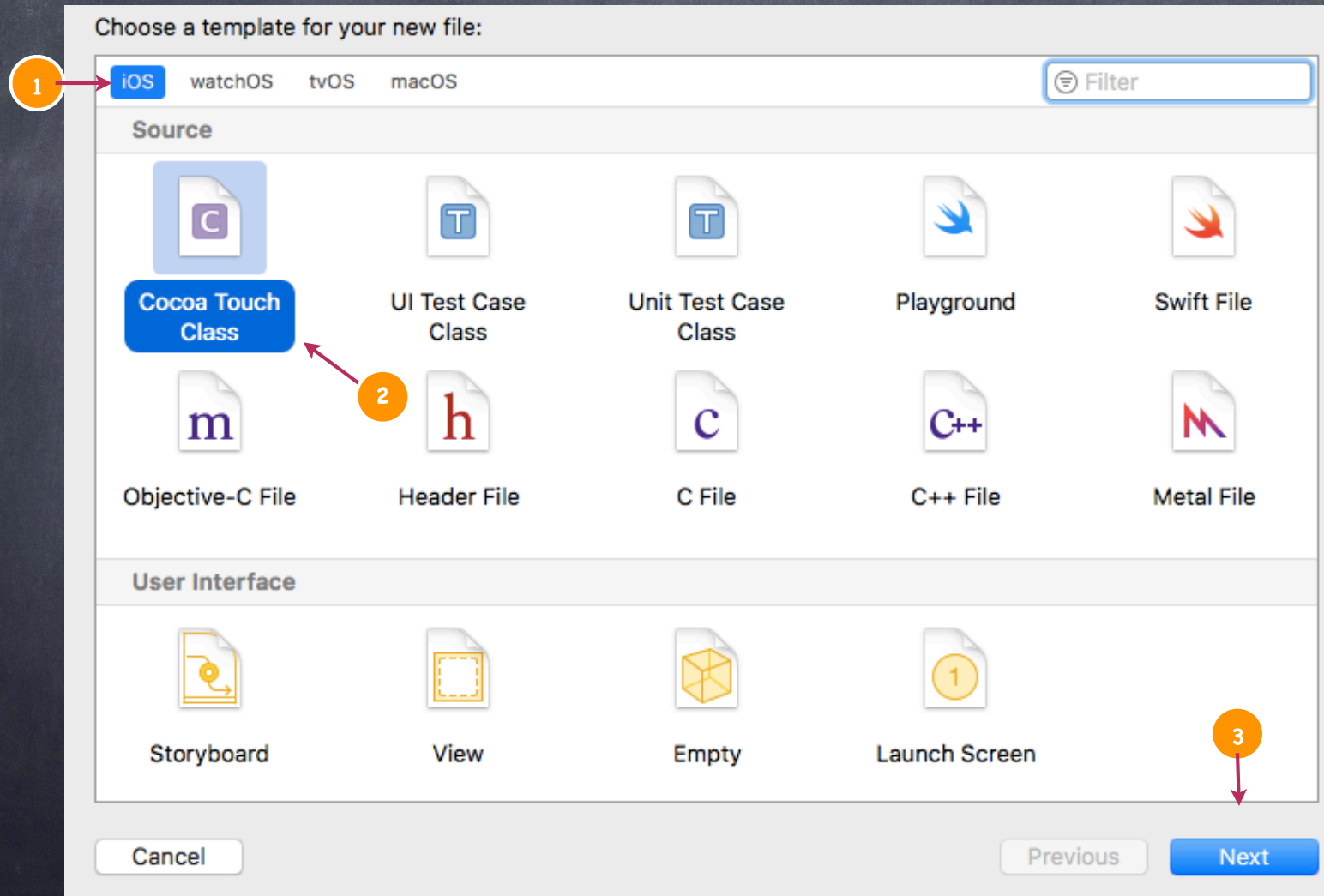
FIAP

- Insira um TableViewController(2) no lugar de onde foi excluída a ViewController, depois clique nos botões 3 e 4 dessa forma ficarão abertas simultaneamente as telas de storyboard e arquivo.h.



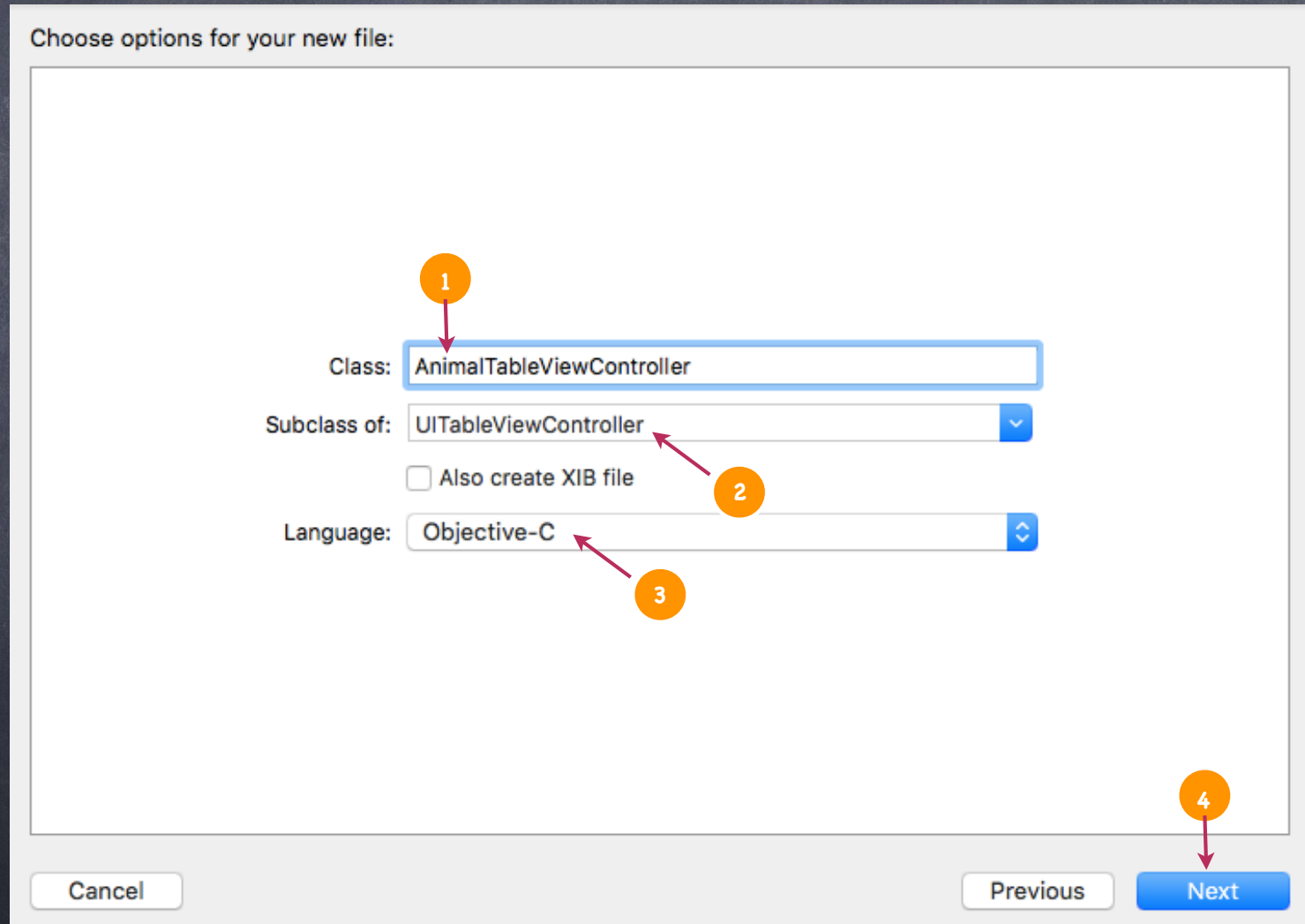
TableViewController

- Clque em File--> New --> File ou Command+N e inclua uma classe em Objective-C, do tipo iOS -> Cocoa Touch Class -> clique em Next.



TableViewController

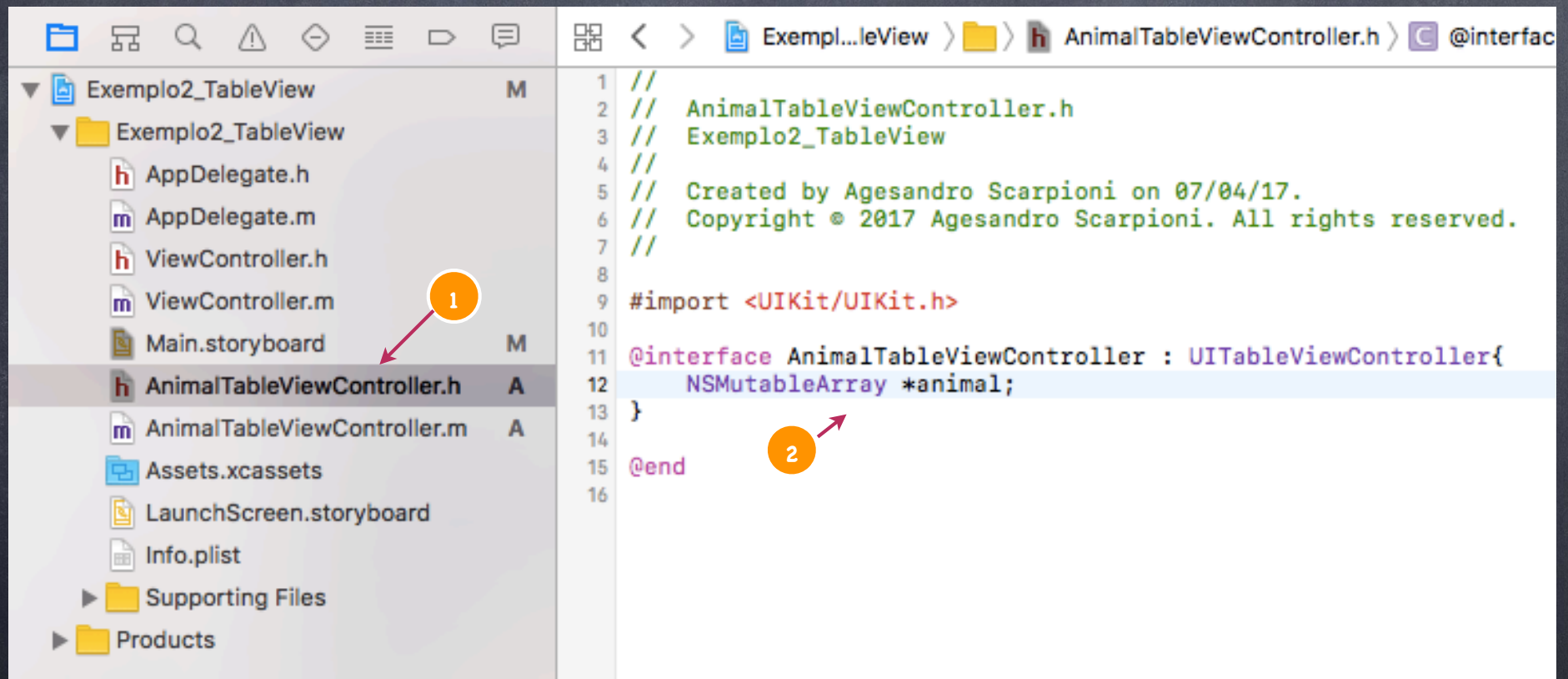
- A classe deve possuir o nome AnimalTableViewController (1), subclasse de UITableViewController(2), language Objective-C e clique em Next.



TableViewController

FIAP

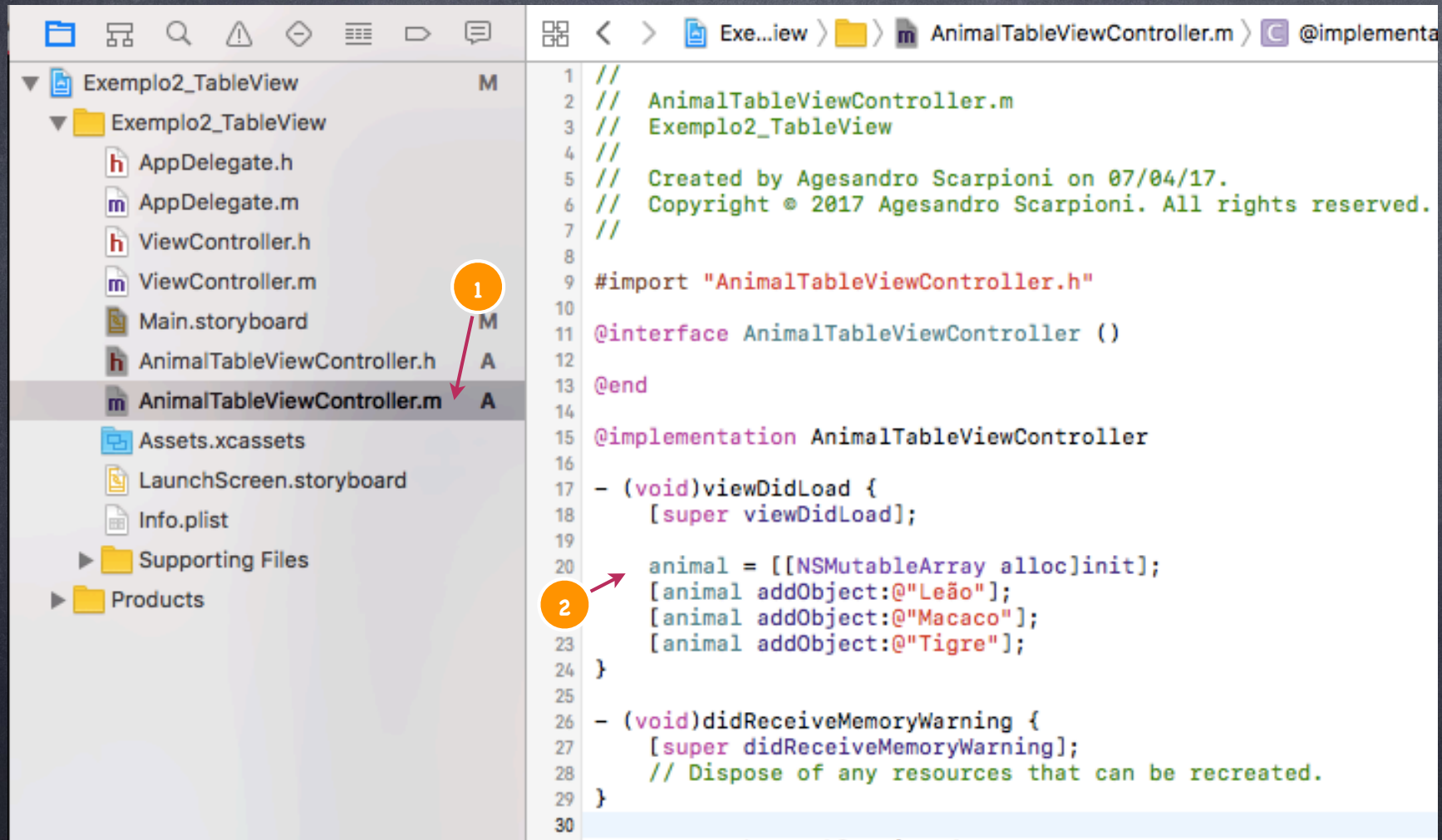
- Na classe AnimalTableViewController.h (1) abra as chaves e crie um array chamado animal (2).



TableViewController

FIAP

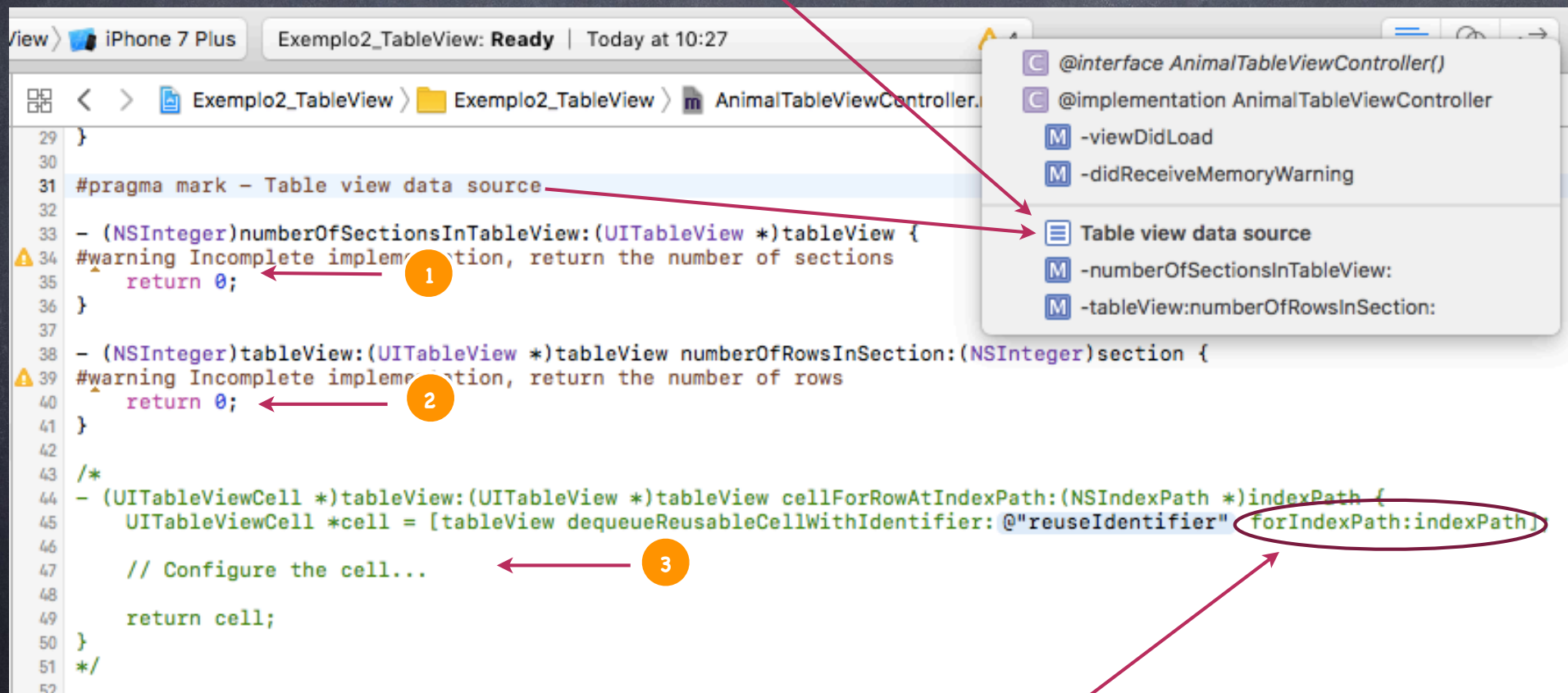
- Na classe AnimalTableViewController.m (1) implemente o array (2).



TableViewController

FIAP

- Note que abaixo do ViewDidLoad, ainda no arquivo .m, todos os métodos que foram digitados no conjunto de slides anterior, já estão prontos para serem implementados aqui, veja (1, 2 e 3). A diretiva #pragma mark serve para organizar o código em seções como é mostrado na figura menor.



TableViewController

FIAP

- Veja que foi implementado os 3 métodos para data source, no método `cellForRowAtIndexPath` no primeiro exemplo foi feito diferente, pois, foi criado um nome para `CellIdentifier` e um nome para a `UITableViewCell`, aqui como já esta quase tudo pronto foi usado o padrão sugerido `*cell` e `@“reuseIdentifier”` (duplo clique aqui).

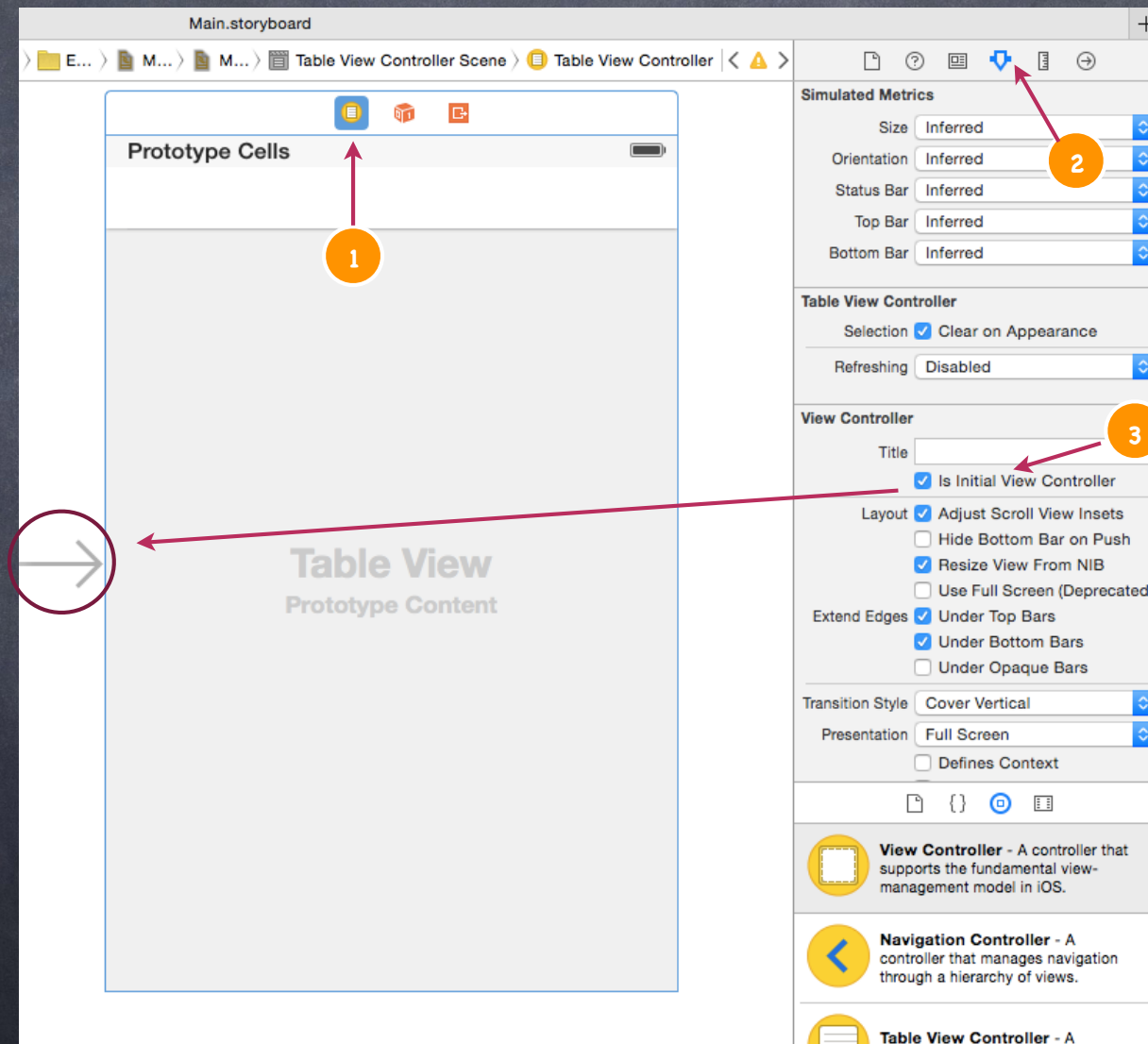
```
38 #pragma mark - Table view data source
39
40 - (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
41     return 1;
42 }
43
44 - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
45     return [animal count];
46 }
47
48
49 - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
50     UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"reuseIdentifier"];
51
52     // Configure the cell...
53     if (cell == nil){
54         cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:@"reuseIdentifier"];
55     }
56     cell.textLabel.text = [animal objectAtIndex:indexPath.row];
57     return cell;
58 }
59 }
```

- Veja aqui o exemplo do programa anterior que não foi utilizado o nome padrão e foi fornecido um nome novo, pois foi tudo implementado tudo do zero.

```
45
46 -(UITableViewCell *) tableView:(UITableView *)tableVie
47 //essa função é chamada para cada item de nosso nu
48 //criamos a célula para a linha da tabela.
49 static NSString *idCelula = @"Celula";
50 UITableViewCell *celula = [tableView dequeueReusableCellWithIdentifier:idCelula];
51 if (celula==nil){
52     //faz cache da célula para evitar criar muitos
53     celula = [[UITableViewCell alloc] initWithStyle:
54 }
55 //celula.textLabel.text = [NSString stringWithForm
56 celula.textLabel.text = [animal objectAtIndex:inde
57 celula.imageView.image = [UIImage imageNamed:@"cor
58 return celula;
59 }
```


TableViewController

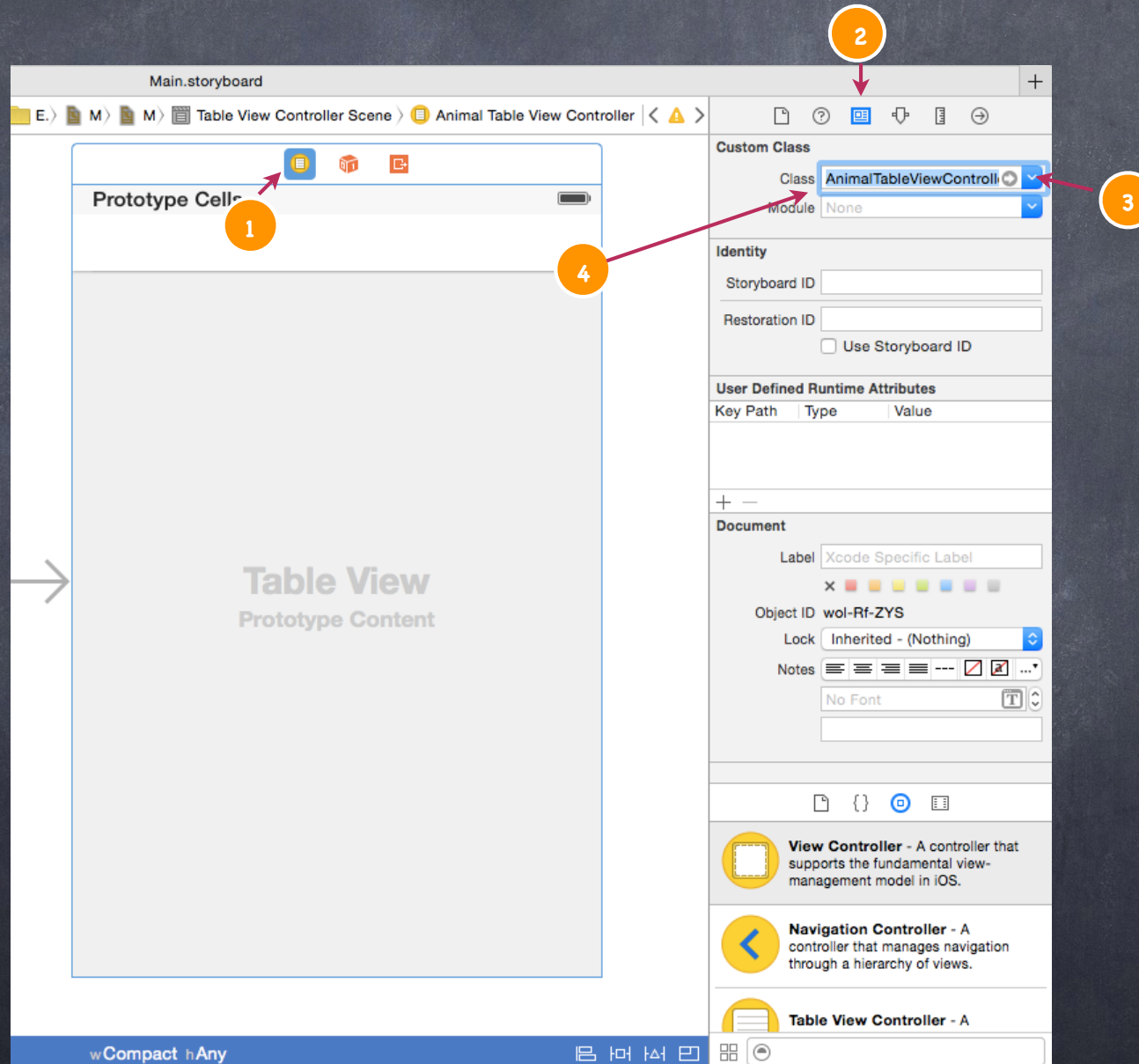
- Indique que a nova ViewController é a View inicial, lembre-se que foi apagada a ViewControlel antiga, para isso siga os passos. (1,2 e 3).



TableViewController

FIAP

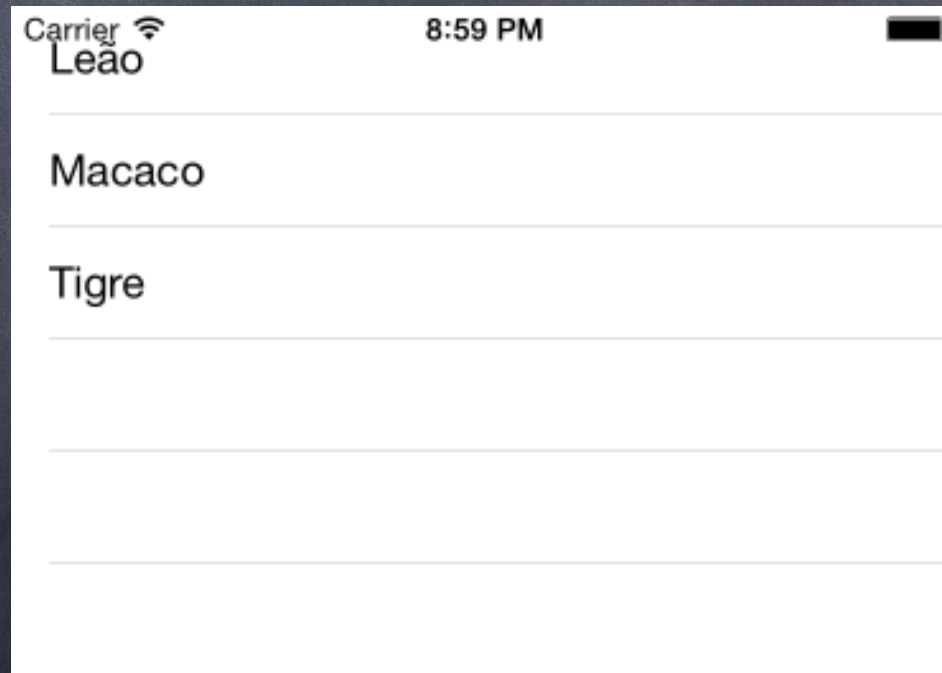
- Indique a classe "Dona" da View, para isso, com a view selecionada(1), em identity inspector (2), abra o combo (3) e escolha a classe AnimalTableViewController (4).



TableViewController

FIAP

- Clique em Run ou Command + R e veja que seus dados já aparecem, ou seja, criando uma classe controller chamada Animal subclasse de UITableViewController é mais prático, comparado ao conjunto de slides anterior, isso porque dessa forma já virá uma série de métodos prontos para serem implementados.



UITableView delegate

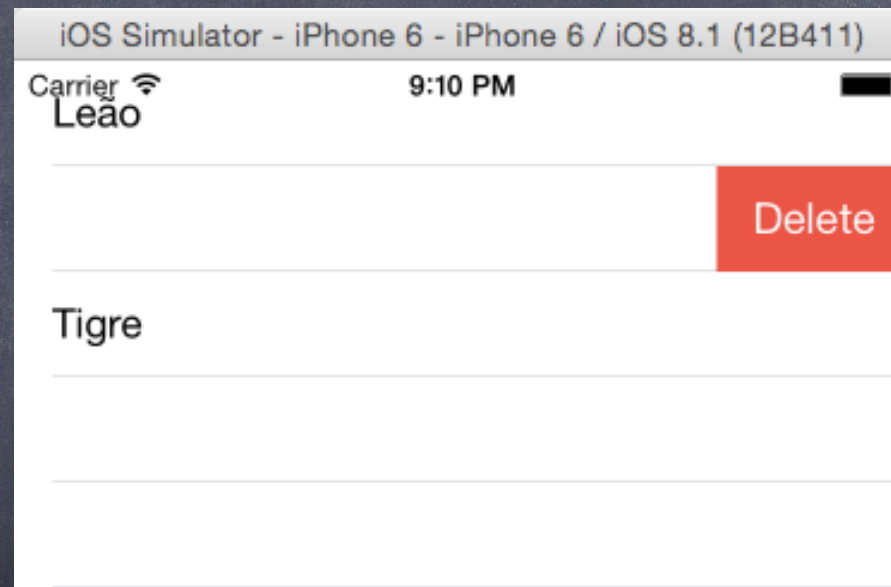
- Procure o método na classe AnimalTableViewController.m chamado commitEditingStyle, desmarque os comentários para fazermos a exclusão do conteúdo da TableView. Veja a implementação na segunda imagem, a única coisa que você fará é remover o item do array(1).

```
80  /*
81  // Override to support editing the table view.
82  - (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:
    (NSIndexPath *)indexPath
83  {
84      if (editingStyle == UITableViewCellEditingStyleDelete) {
85          // Delete the row from the data source
86          [tableView deleteRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationFade];
87      }
88      else if (editingStyle == UITableViewCellEditingStyleInsert) {
89          // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view
90      }
91  }
92  */
```

```
80  // Override to support editing the table view.
81  - (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath
82  {
83      if (editingStyle == UITableViewCellEditingStyleDelete) {
84          // Delete the row from the data source
85          |
86          [ animal removeObjectAtIndex:indexPath.row]; ← 1
87          [tableView deleteRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationFade];
88      }
89      else if (editingStyle == UITableViewCellEditingStyleInsert) {
90          // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view
91      }
92  }
```

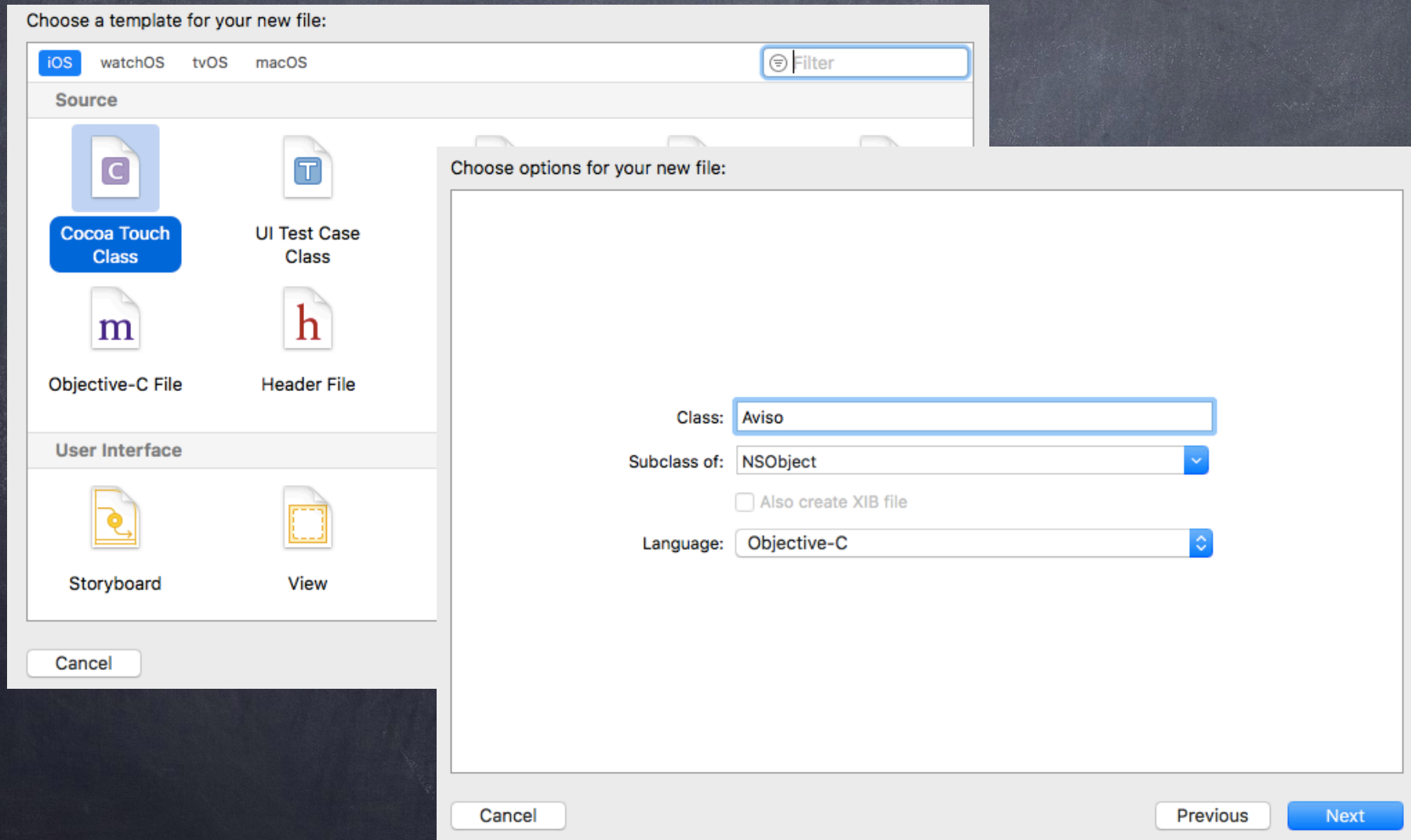

UITableView delegate FIAP

- Clique em Run ou Command + R para executar, ao passar o mouse sobre o item Macaco com o botão pressionado, veja que o botão Delete aparece, confirme para ver a exclusão do item.



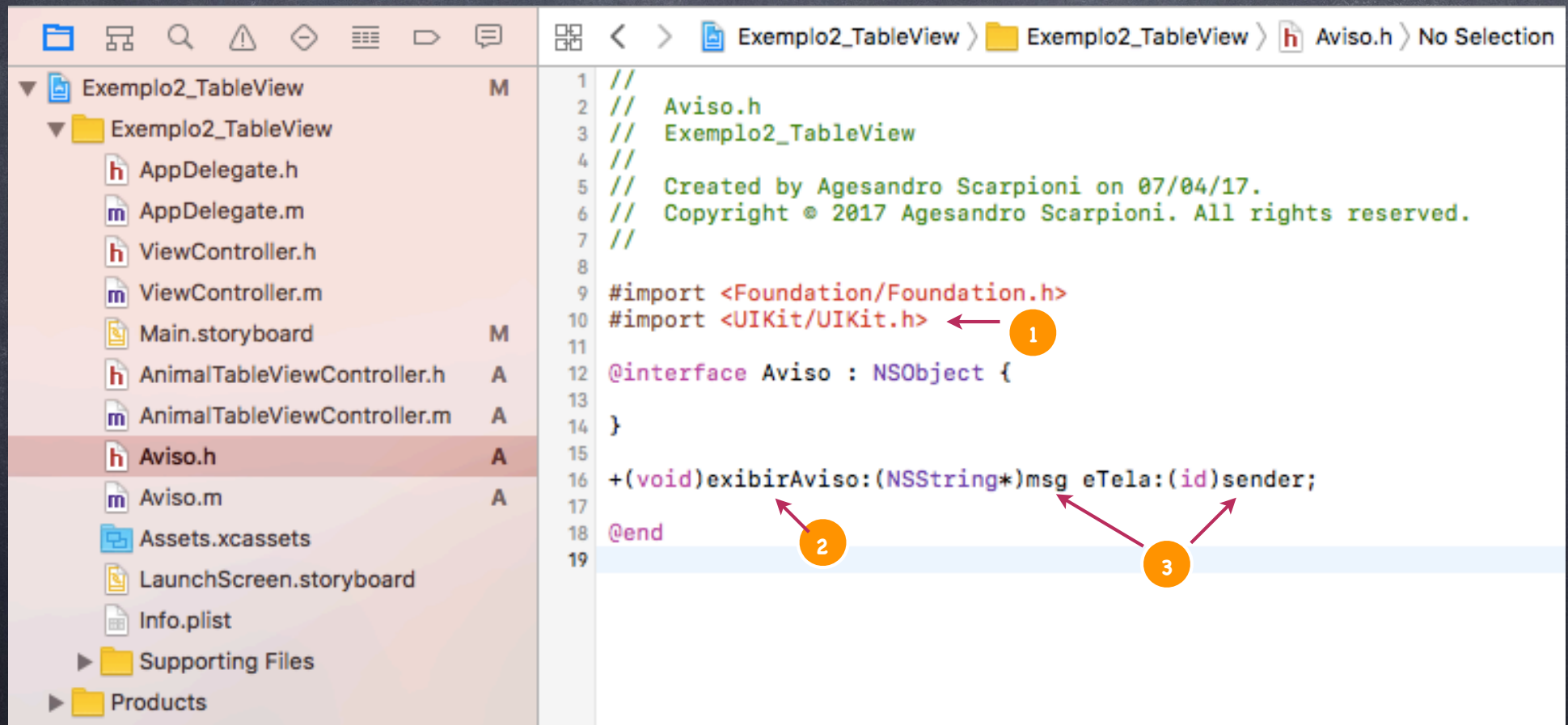
Classe Aviso

- Crie uma nova classe, Command + N, do tipo iOS → Cocoa Touch Class, com o nome Aviso subclasse de NSObject.



Classe Aviso

- Faça o import do UIKit/UIKit.h (1) e declare um método estático(+) no arquivo .h chamado `exibirAviso(2)`, que receba como parâmetro um ponteiro NSString chamado `msg(3)` e a View (tela) que irá exibir a mensagem.



Classe Aviso

- Implemente o método no arquivo .m para exibir as mensagens.

```
1 //
2 //  Aviso.m
3 //  Exemplo2_TableView_OBJC
4 //
5 //  Created by agesandro scarpioni on 22/05/16.
6 //  Copyright © 2016 Agesandro Scarpioni. All rights reserved.
7 //
8 #import "Aviso.h"
9
10 @implementation Aviso
11
12 +(void)exibirAviso:(NSString*)msg eTela:(id)sender{
13
14     UIAlertController *alerta = [UIAlertController
15                                 alertControllerWithTitle:@"Aviso"
16                                 message:msg
17                                 preferredStyle:UIAlertControllerStyleAlert];
18
19     UIAlertAction *ok = [UIAlertAction
20                         actionWithTitle:@"OK"
21                         style:UIAlertActionStyleDefault
22                         handler:^(UIAlertAction * _Nonnull action) {
23                             [alerta dismissViewControllerAnimated:YES completion:nil];
24                         }];
25
26     [alerta addAction:ok];
27
28     [sender presentViewController:alerta animated:YES completion:nil];
29 }
30 @end
31
32
```


UITableView delegate

- Na classe AnimalTableViewController.m, faça o import da sua classe Aviso.h (1) que possui o método estático que foi implementado, digite o método didSelectRowAtIndexPath (2) na classe AnimalTableViewController.m, este método será chamado sempre quando uma linha é selecionada no UITableView.

```
1 //
2 // AnimalTableViewController.m
3 // Exemplo2_TableView
4 //
5 // Created by Agesandro Scarpioni on 07/04/17.
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import "AnimalTableViewController.h"
10 #import "Aviso.h"
11
```

1

2

```
84
85 -(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{
86     NSString *msg = [NSString stringWithFormat:@"Selecionamos o animal %ld", (long)indexPath.row];
87     [Aviso exhibirAviso:msg eTela:self];
88 }
```


TableView

- Command + R, clique em uma célula do TableView, veja o resultado com o método estático "aviso".

