

TableViewController

Parte 2

X-Code - Swift

Prof. Agesandro Scarpioni

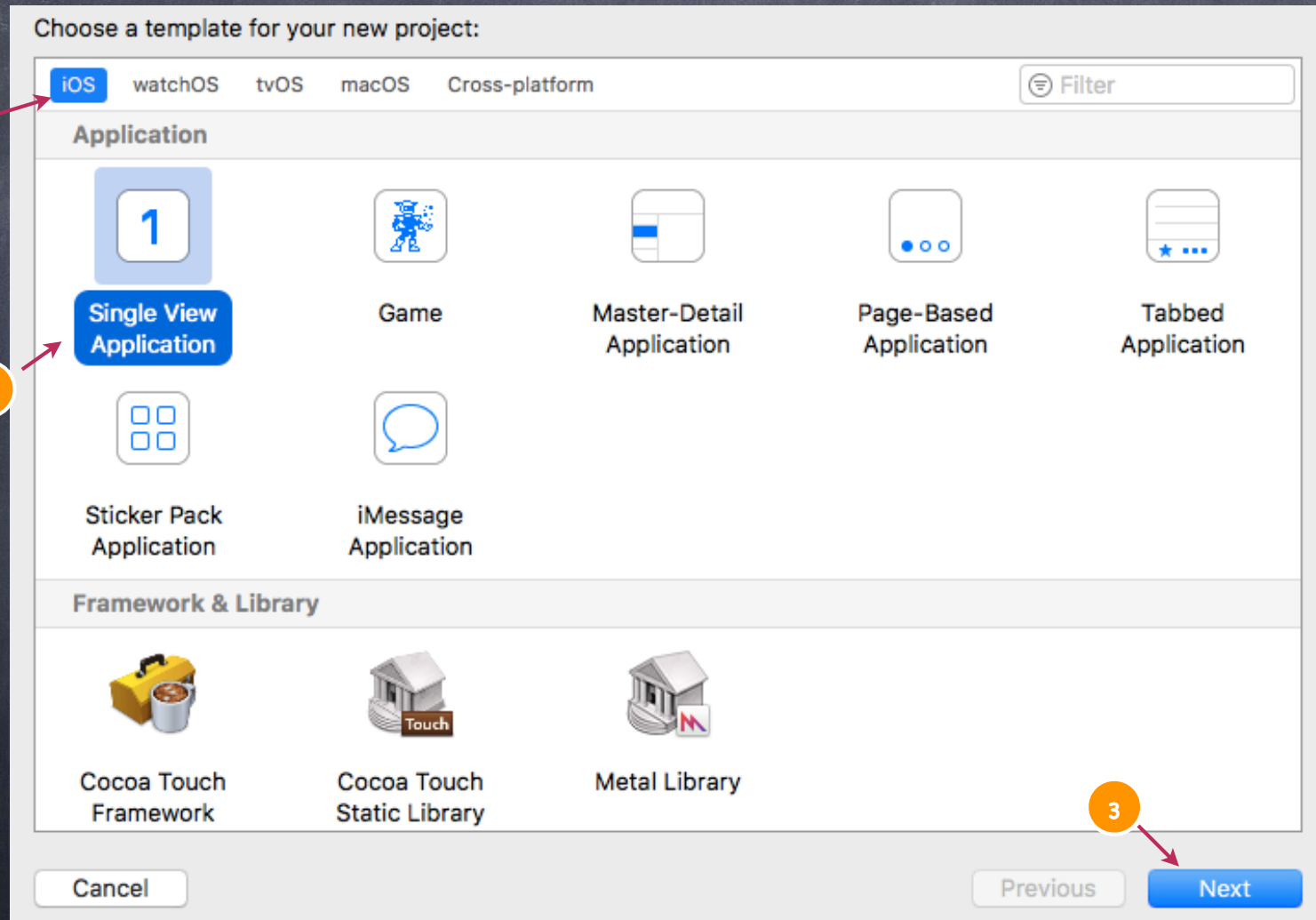
agesandro@fiap.com.br

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_Swift" (4), escolha a linguagem Swift(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:

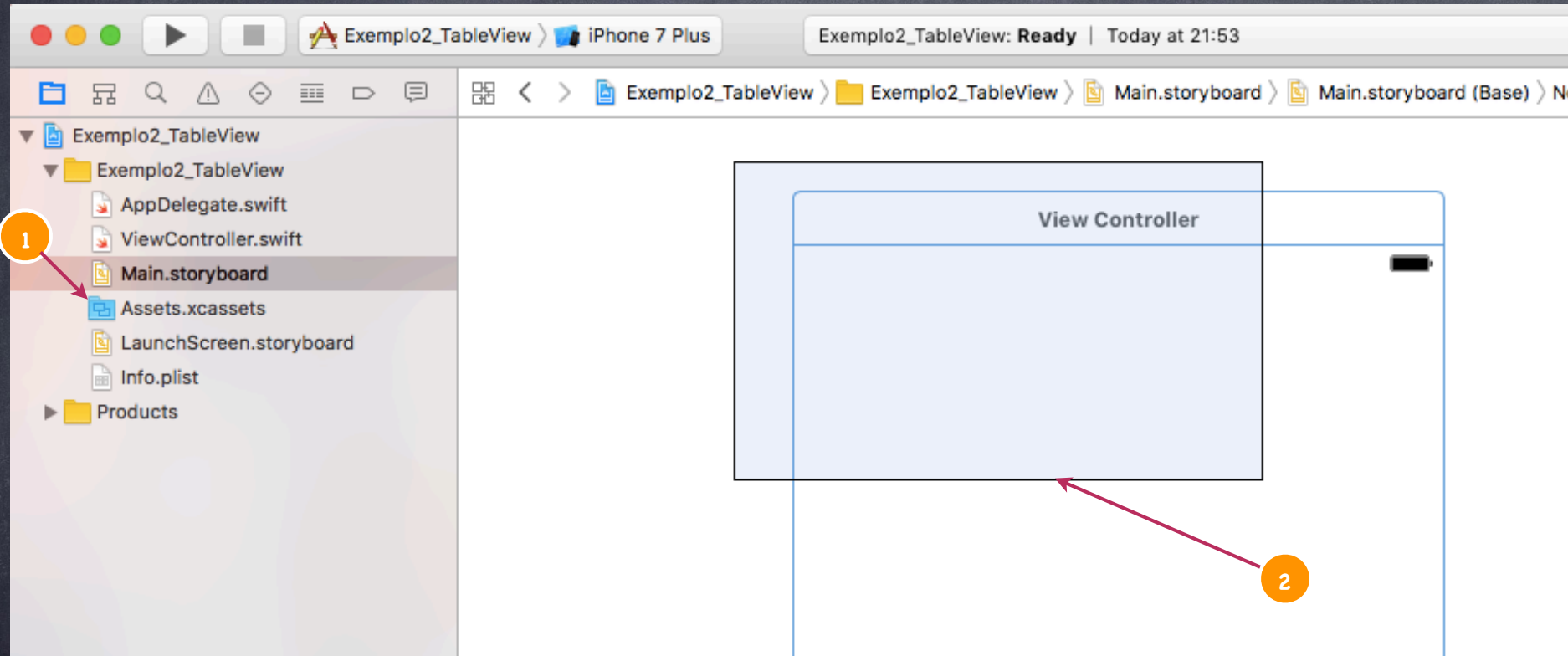
☐ Use Core Data

☐ Include Unit Tests

☐ Include UI Tests

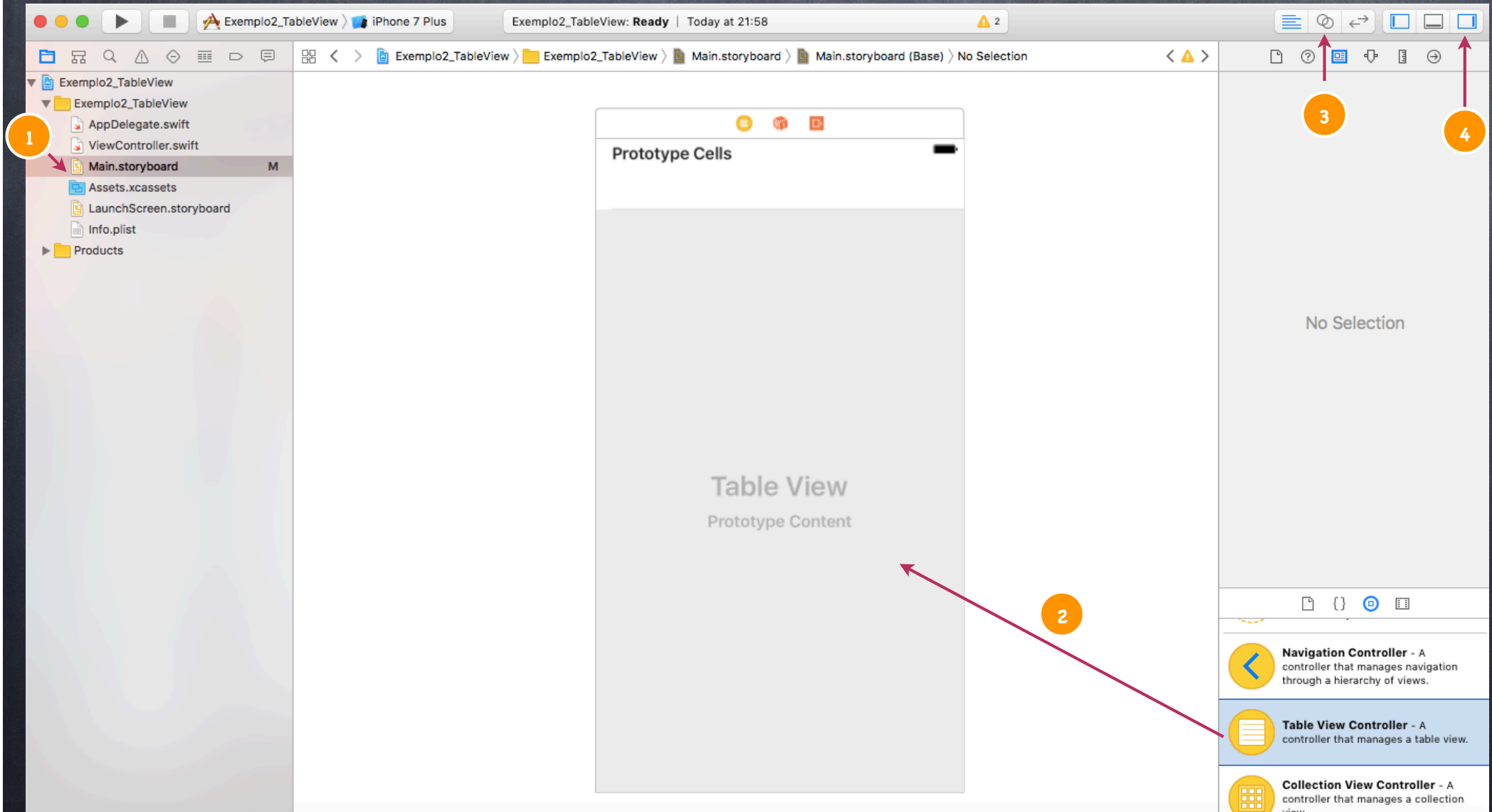
TableViewController

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



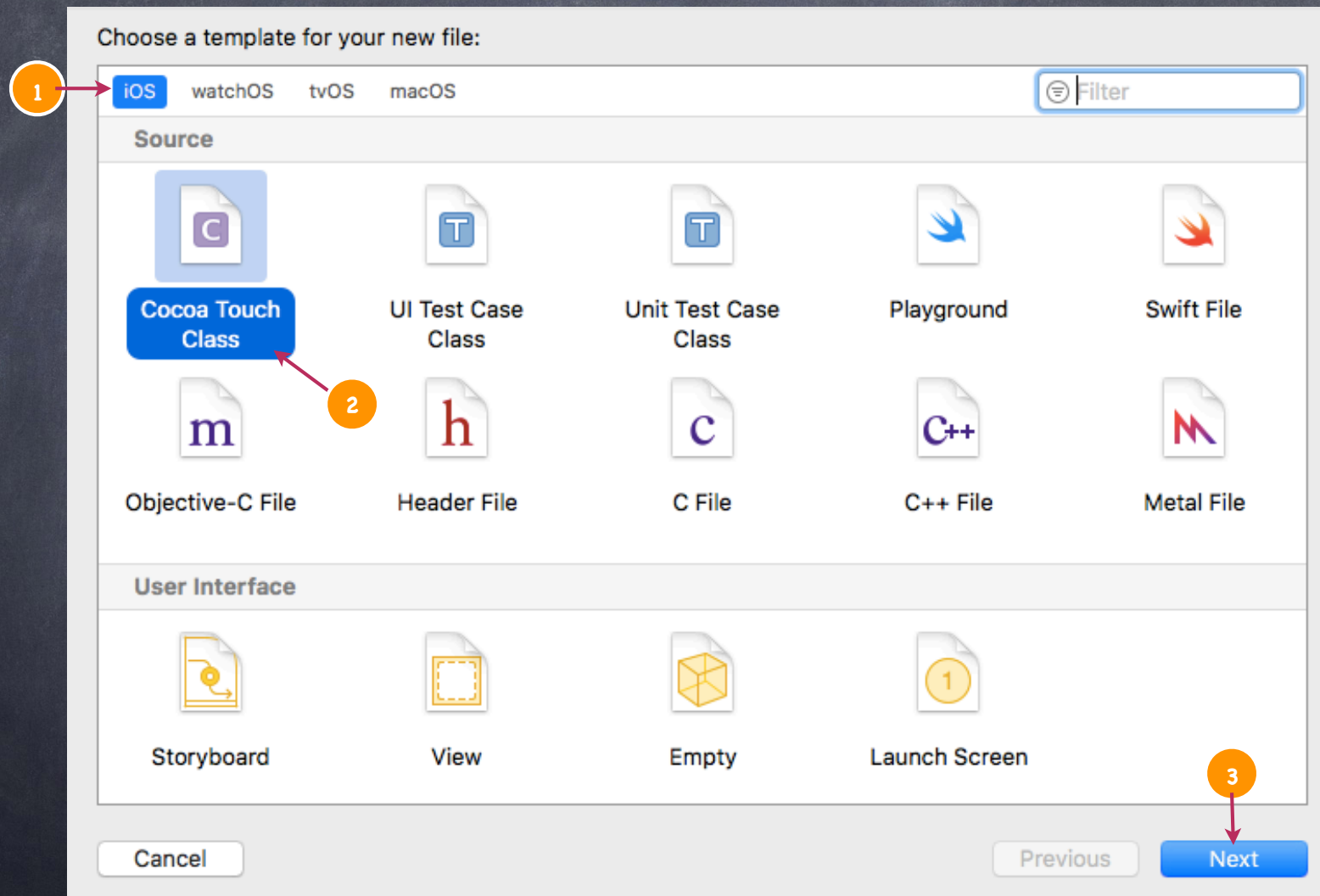
TableViewController

- 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 ViewController.swift.



TableViewController

- Clique em File--> New --> File ou Command+N e inclua uma classe do tipo Cocoa Touch Class(2), clique em Next (3).



TableViewController

- A classe deve possuir o nome AnimalTableViewController (1), subclasse de UITableViewController(2), language Swift(3) e clique em Next(4).

Choose options for your new file:

Class:

Subclass of:

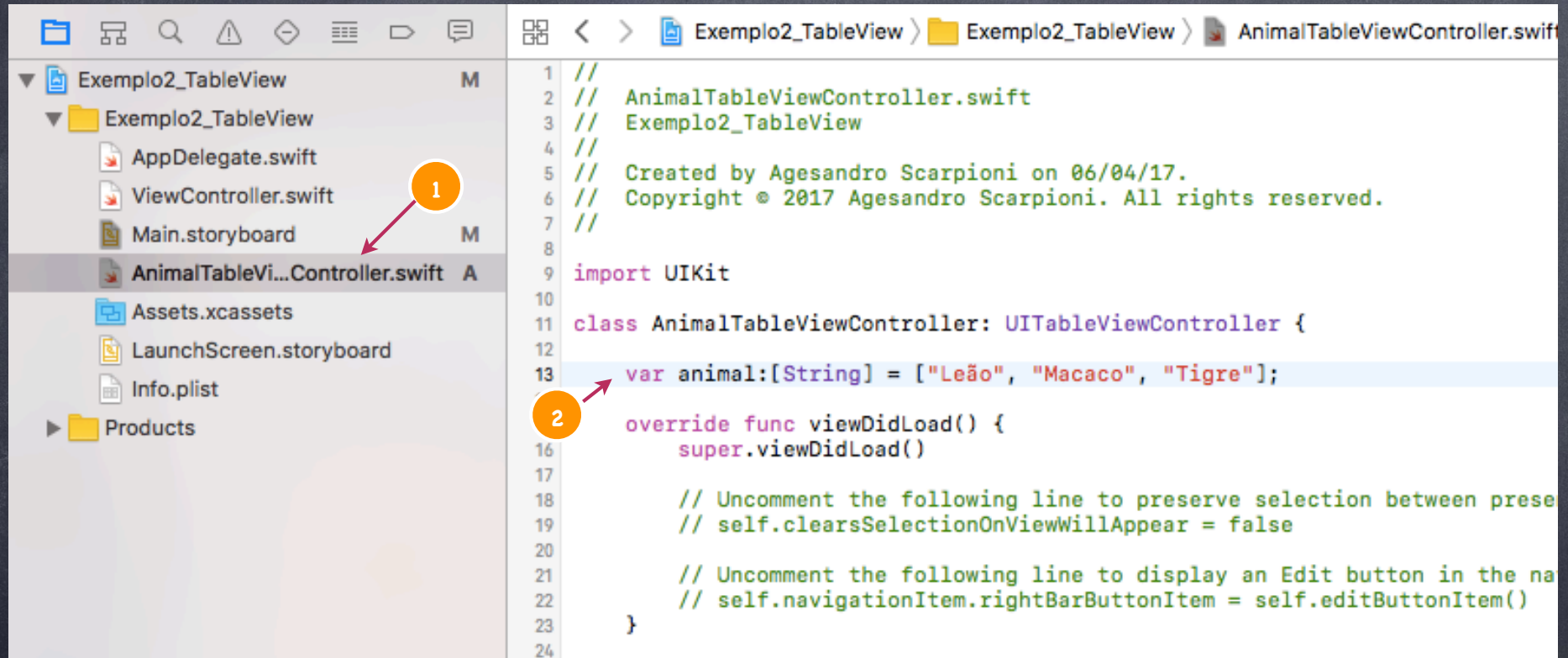
☐ Also create XIB file

Language:

Cancel Previous Next

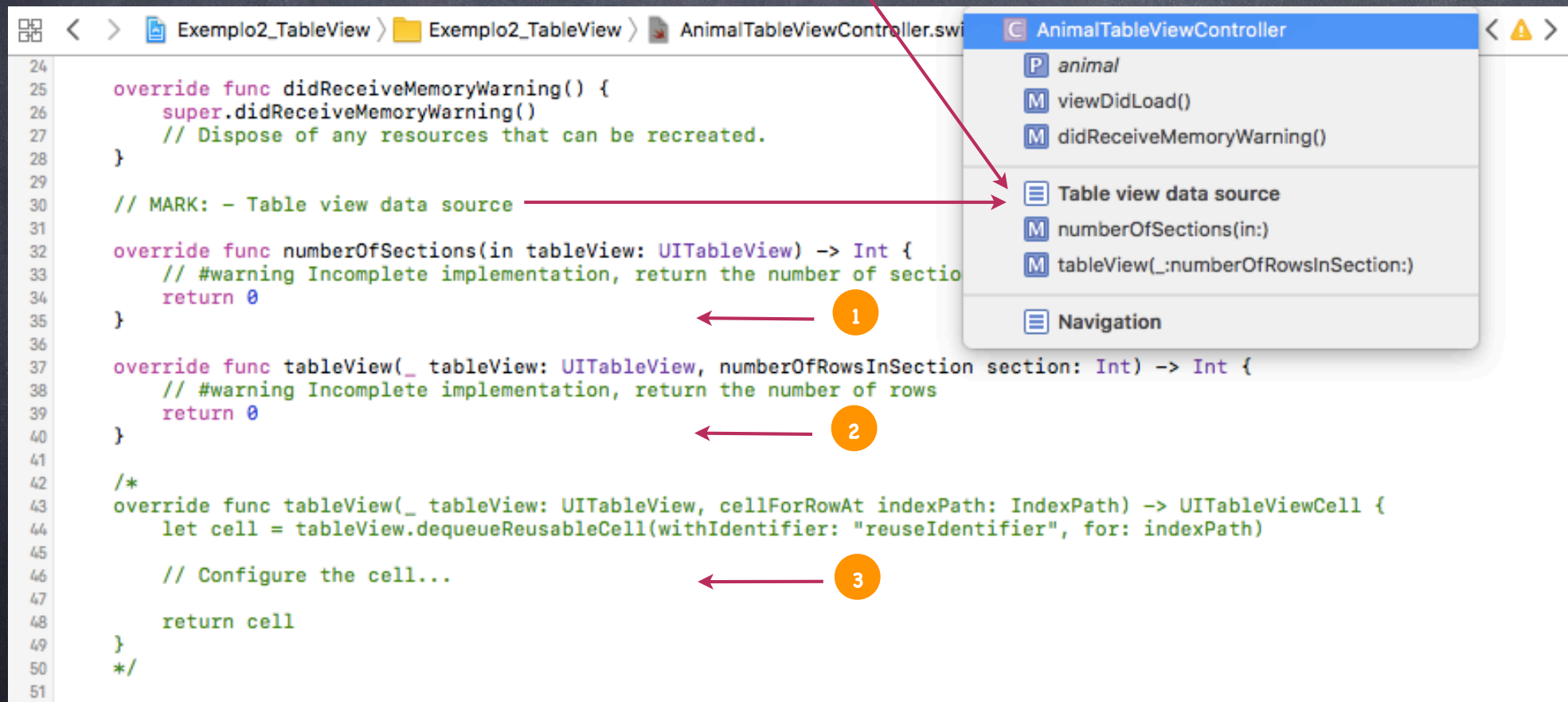
TableViewController

- Na classe AnimalTableViewController.swift (1) crie um array chamado animal (2).



TableViewController

- Note que abaixo do ViewDidLoad, 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 // MARK: serve para organizar o código em seções como é mostrado na figura menor.

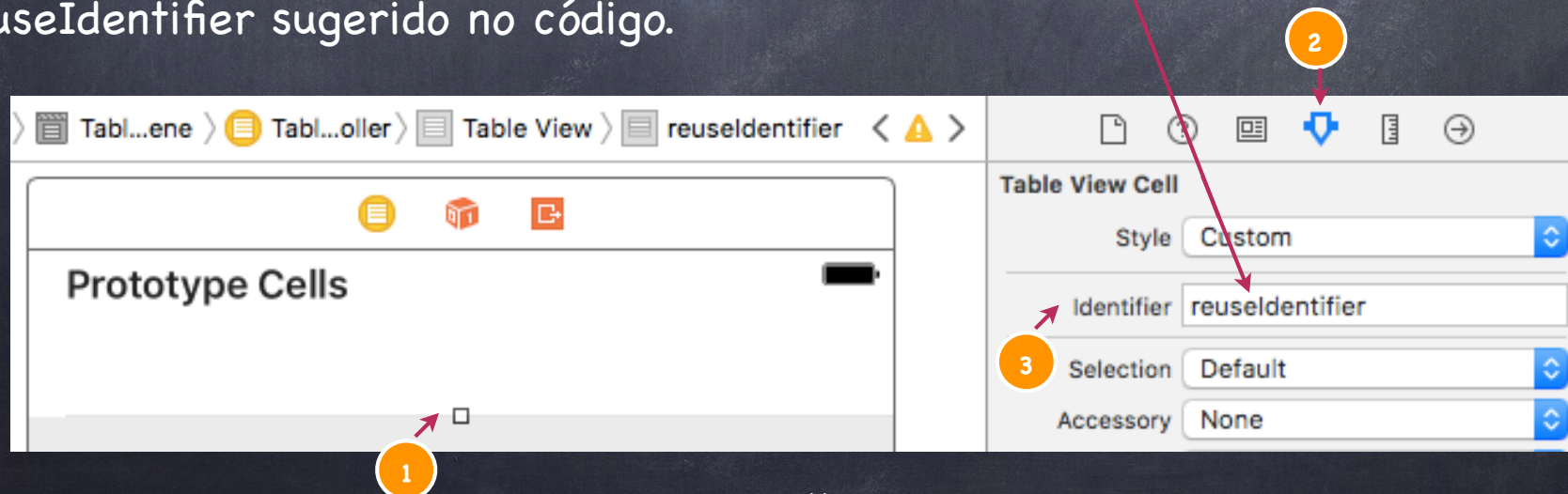


TableViewController

- Veja que foi implementado os 3 métodos para data source.

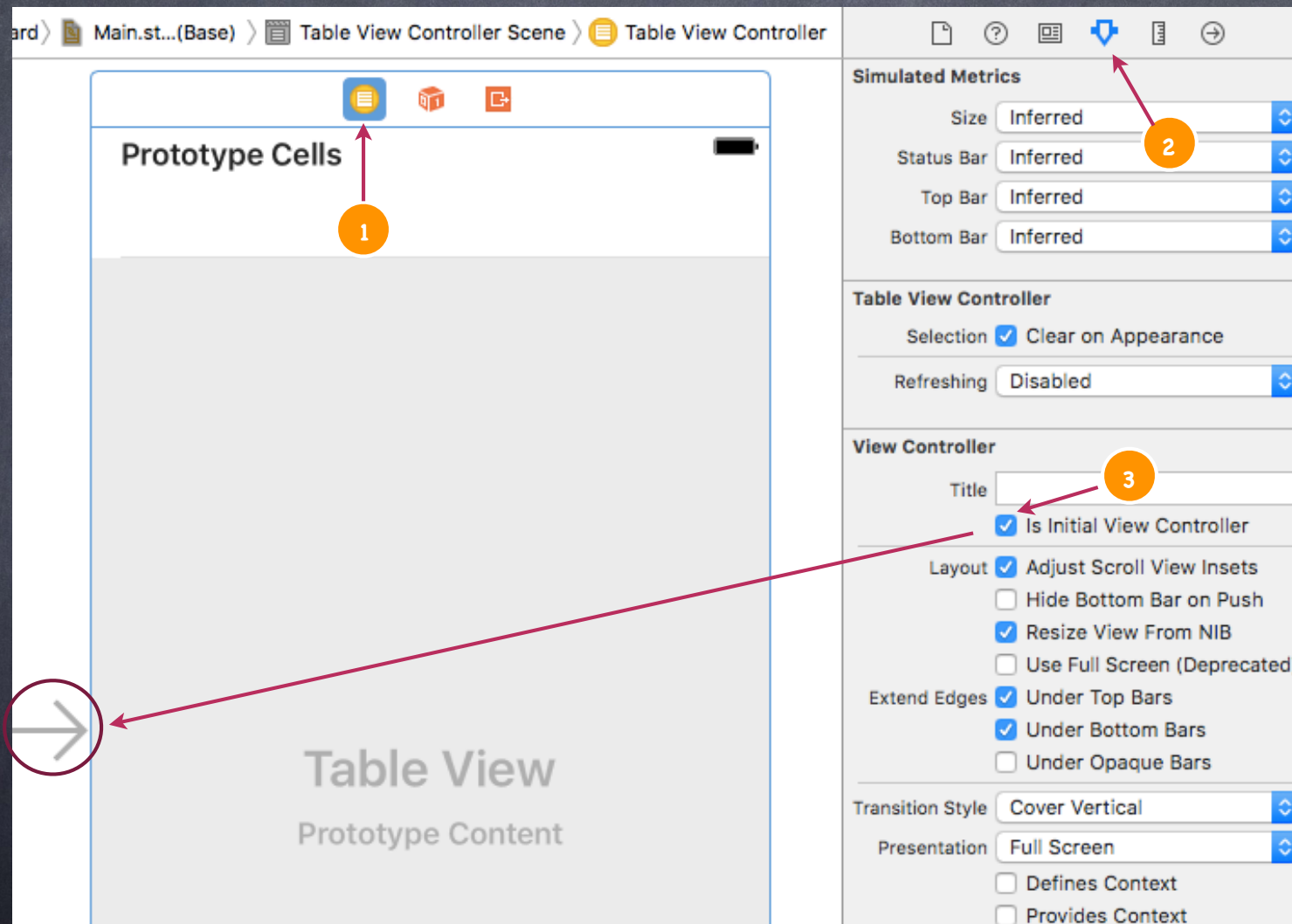
```
29
30 // MARK: - Table view data source
31
32 override func numberOfSections(in tableView: UITableView) -> Int {
33     // #warning Incomplete implementation, return the number of sections
34     return 1
35 }
36
37 override func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
38     // #warning Incomplete implementation, return the number of rows
39     return animal.count
40 }
41
42
43 override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
44     let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
45
46     // Configure the cell...
47     cell.textLabel?.text = animal[indexPath.row]
48     return cell
49 }
50
```

- Selecione a célula(1), clique em attributes(2), em Identifier(3) digite o nome reuseIdentifier sugerido no código.



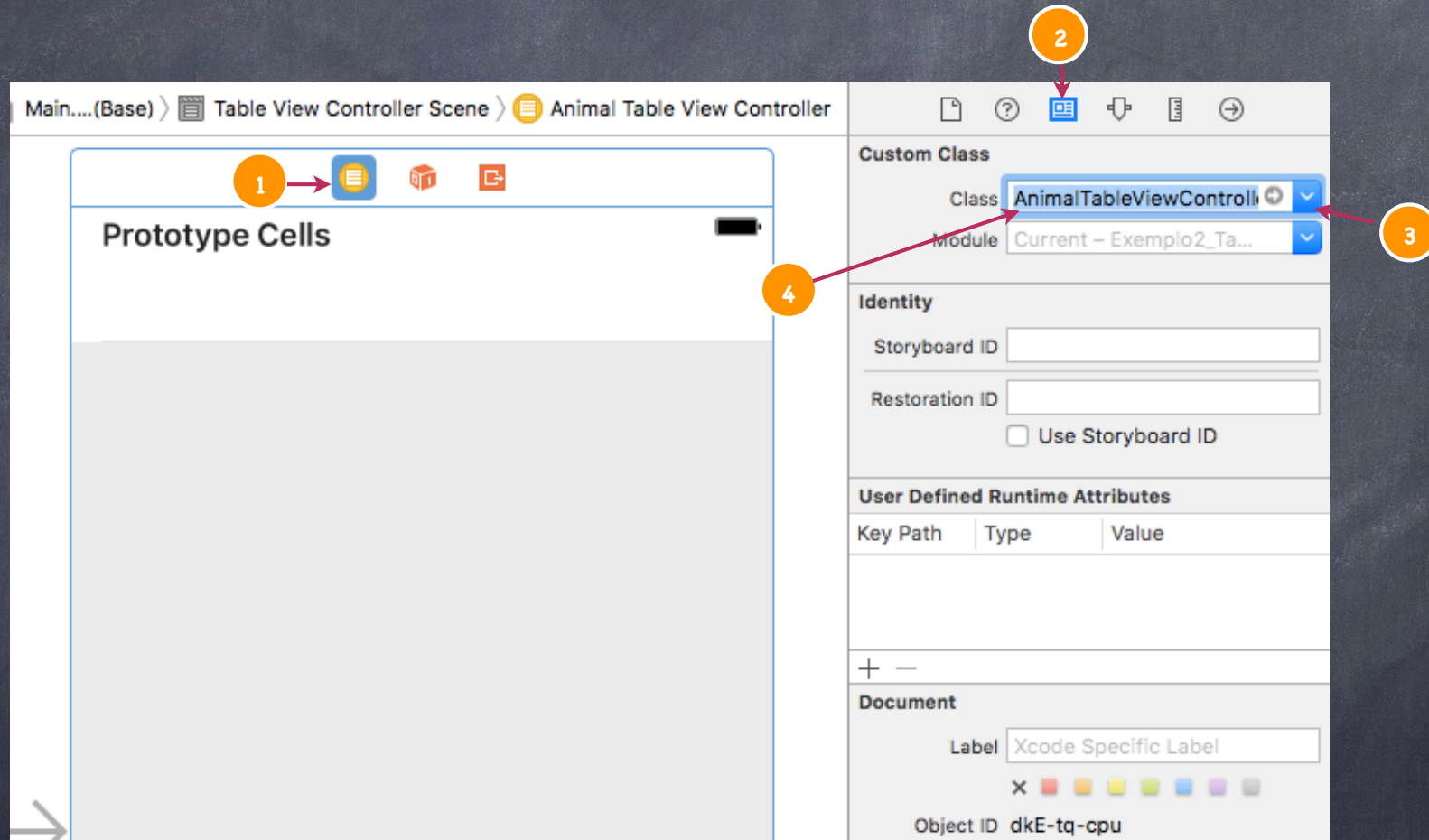
TableViewController

- 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).



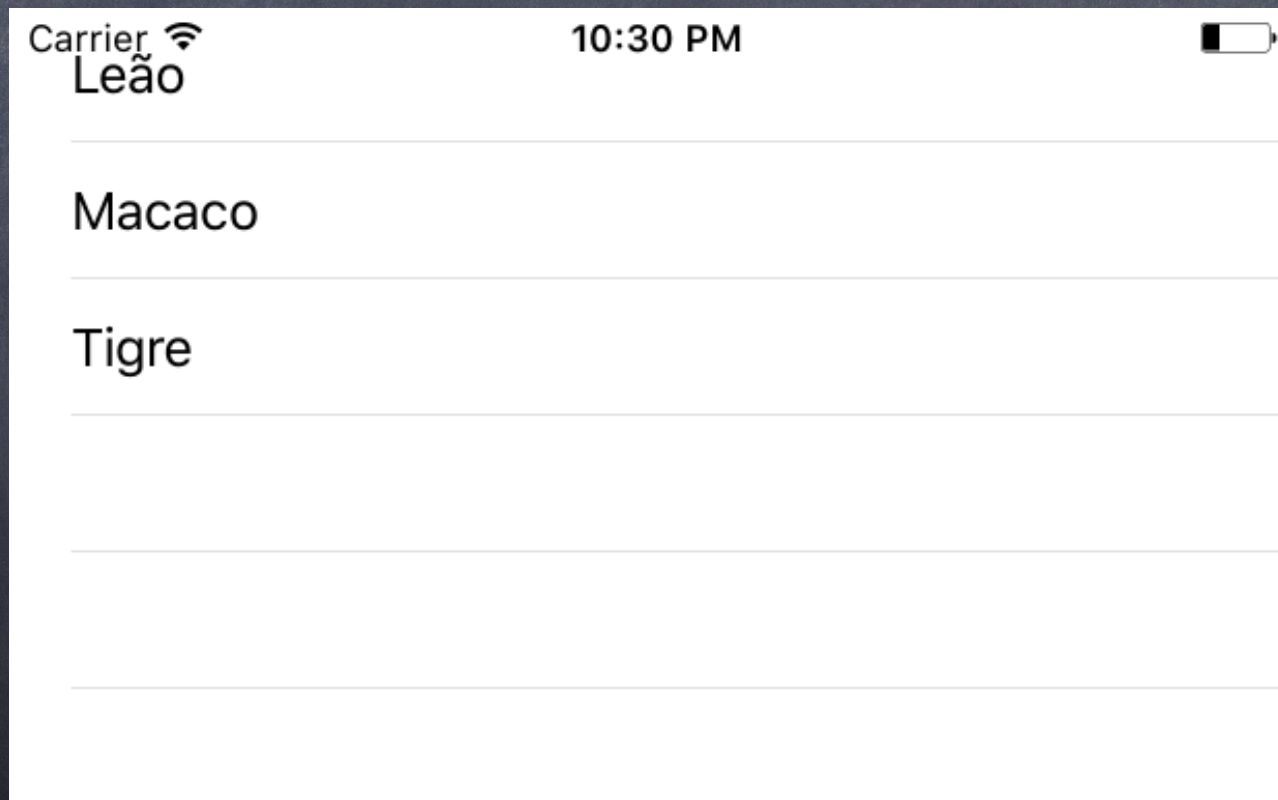
TableViewController

- 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

- 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

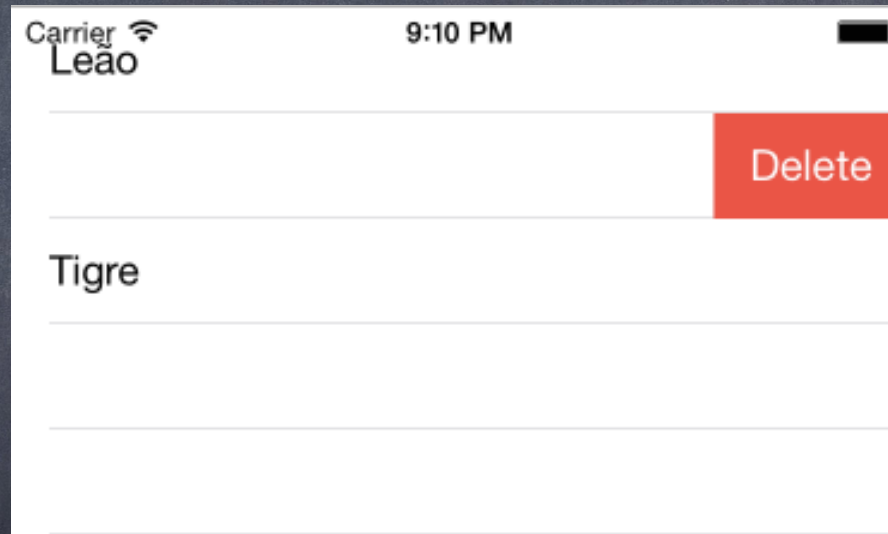
- Procure o método na classe AnimalTableViewController.swift chamado `commitEditingStyle`, desmarque os comentários para fazermos a exclusão do conteúdo da TableView. Veja a implementação na segunda imagem, a única coisa que você fará é remover o item do array(1).

```
59
60  /*
61  // Override to support editing the table view.
62  override func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
63      if editingStyle == .delete {
64          // Delete the row from the data source
65          tableView.deleteRows(at: [indexPath], with: .fade)
66      } else if editingStyle == .insert {
67          // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view
68      }
69  }
70  */
71
```

```
60
61  // Override to support editing the table view.
62  override func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
63      if editingStyle == .delete {
64          // Delete the row from the data source
65          animal.remove(at: indexPath.row) ← 1
66          tableView.deleteRows(at: [indexPath], with: .fade)
67      } else if editingStyle == .insert {
68          // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view
69      }
70  }
71
```

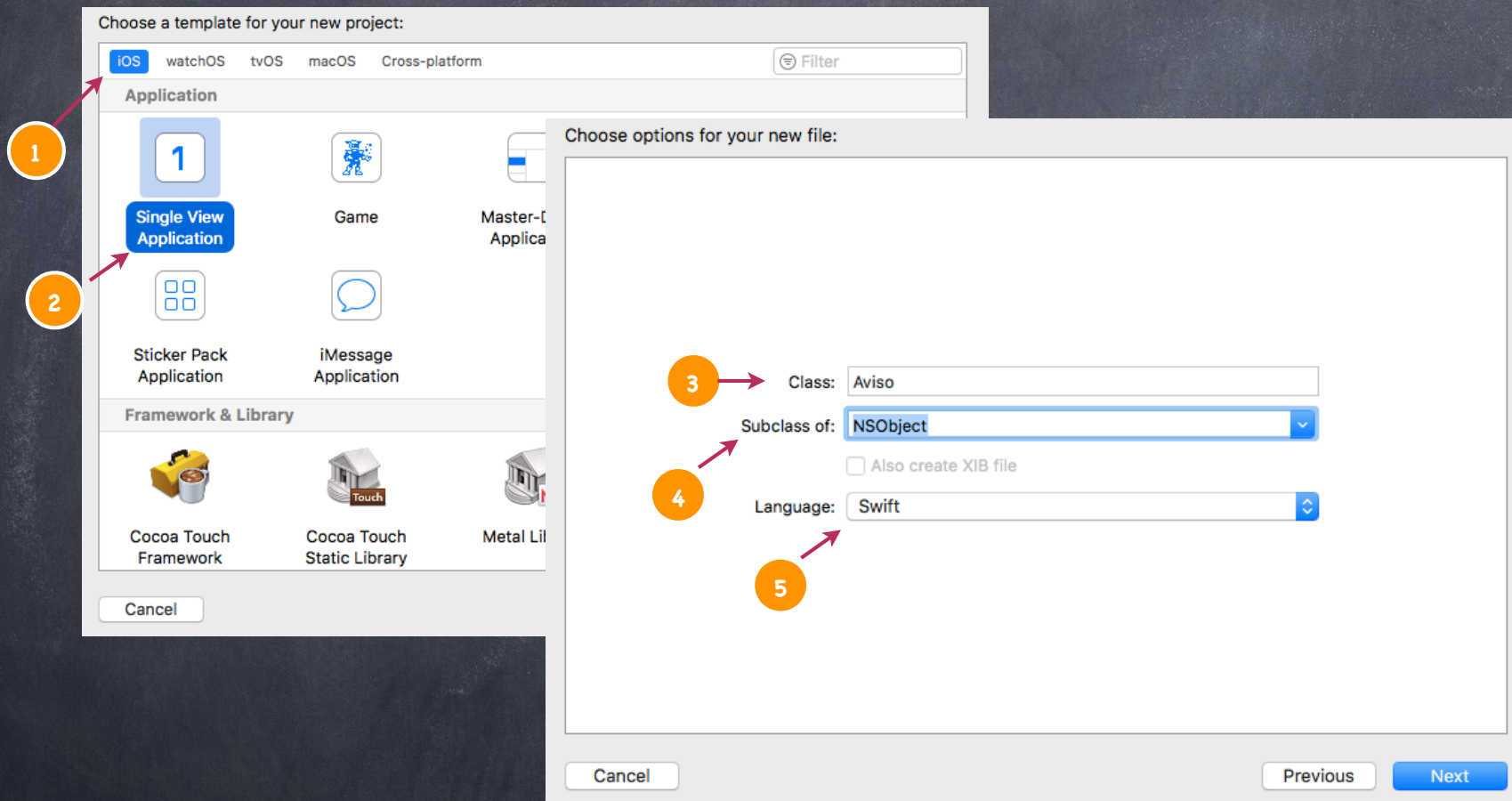

UITableView delegate

- 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, chamada Aviso subclasse de NSObject, linguagem Swift.




Classe Aviso

- Declare um método estático chamado `exibirAviso(1)` no arquivo `Aviso.swift`, para isso use a palavra `static` antes da palavra `func`, esse método deve receber como parâmetro um ponteiro `NSString` chamado `msg(2)` e a `View` (tela) que irá exibir a mensagem.


```
1 //
2 //  Aviso.swift
3 //  Exemplo2_TableView
4 //
5 //  Created by Agesandro Scarpioni on 06/04/17.
6 //  Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 import UIKit
10
11 class Aviso: NSObject {
12     static func exibirAviso(msg:String, sender:AnyObject){
13         let alerta = UIAlertController(
14             title: "Aviso",
15             message: msg,
16             preferredStyle: .alert)
17
18         alerta.addAction(UIAlertAction(
19             title: "OK",
20             style: UIAlertActionStyle.default,
21             handler: nil))
22
23         sender.present(alerta, animated: true, completion: nil)
24     }
25 }
26
27
28
```


UITableView delegate

- Na classe AnimalTableViewController.swift, digite o método didSelectRowAtIndexPath (1), neste método será chamado o método estático exibirAviso que foi criado na classe Aviso. O método didSelectRowAtIndexPath é executado quando uma linha é selecionada no UITableView. No código (1) é apresentado a posição do item na tabela, no código (2) é apresentado o conteúdo do item da tabela.



```
72 override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {  
73     let texto = "Selecionamos o animal \(indexPath.row)"  
74     Aviso.exibirAviso(msg: texto, sender: self)  
75 }
```



```
72 override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {  
73     let texto = "Selecionamos o animal \(animal[indexPath.row])"  
74     Aviso.exibirAviso(msg: texto, sender: self)  
75 }
```


TableView

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

