

UIKit e Alertas

Conhecendo outros componentes

X-Code com Swift

Prof. Agesandro Scarpioni

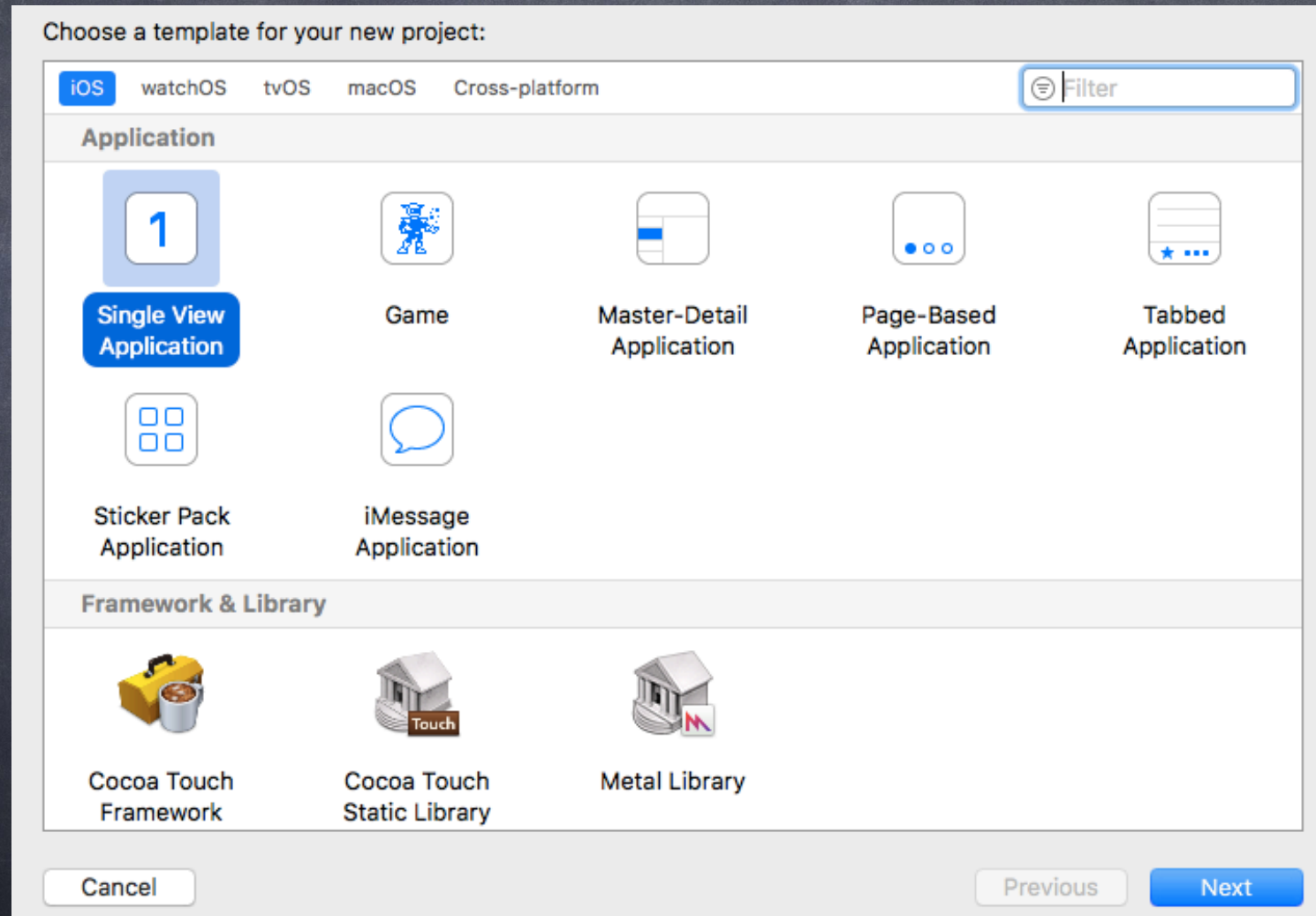
agesandro@fiap.com.br

Outros Componentes – UIKit

- Vamos criar um aplicativo para testarmos vários componentes que serão de grande utilidade no desenvolvimento de seus APP's, além de exibirmos mensagens para os usuários.

Iniciando o Projeto

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



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, desmarque o 2º e 3º check box. Em Devices selecione iPhone, em Language escolha Swift.

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

☐ Use Core Data

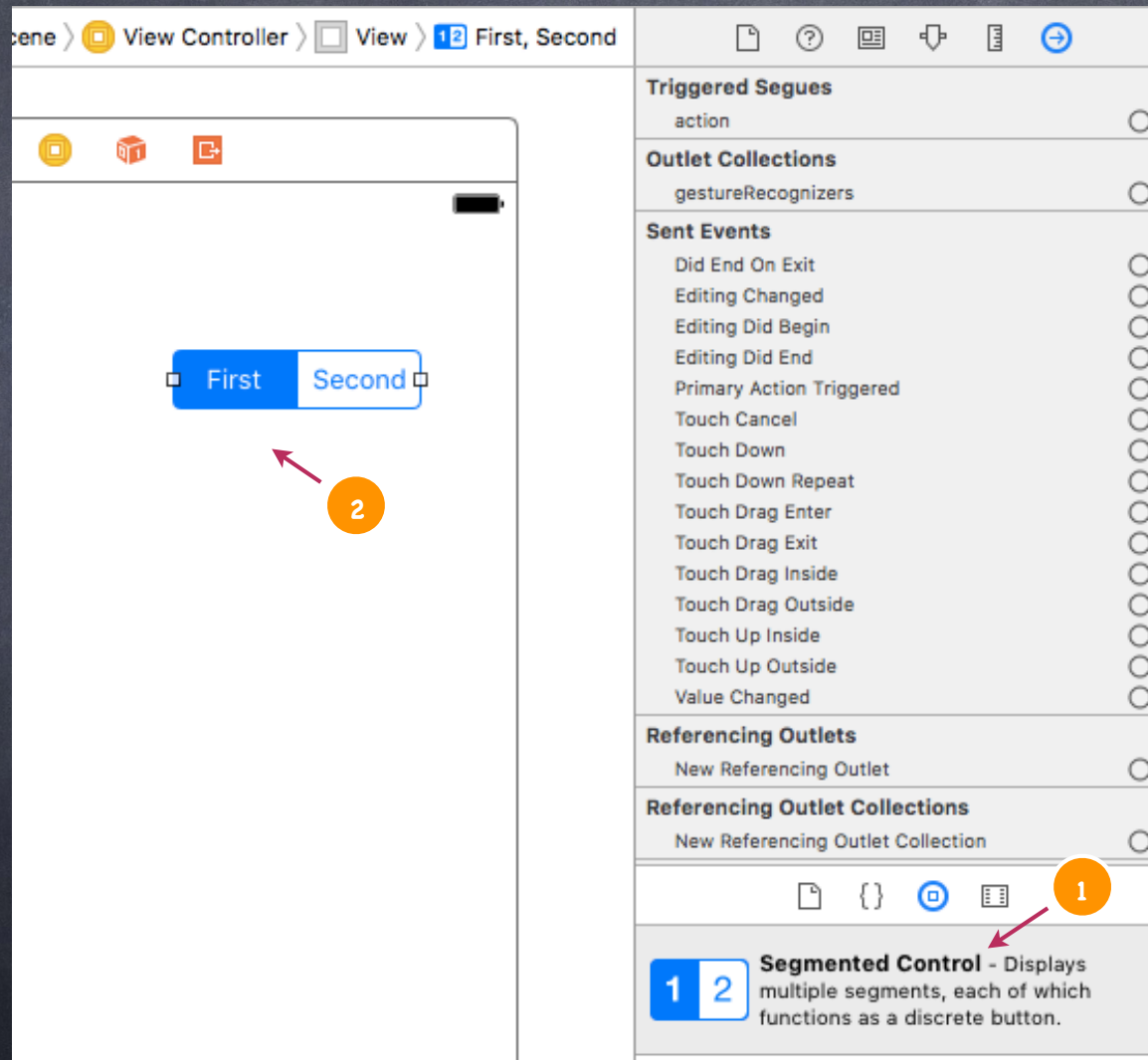
☐ Include Unit Tests

☐ Include UI Tests

ATENÇÃO: O Nome do projeto não pode ser apenas UIKIT pois esse é o nome de uma classe

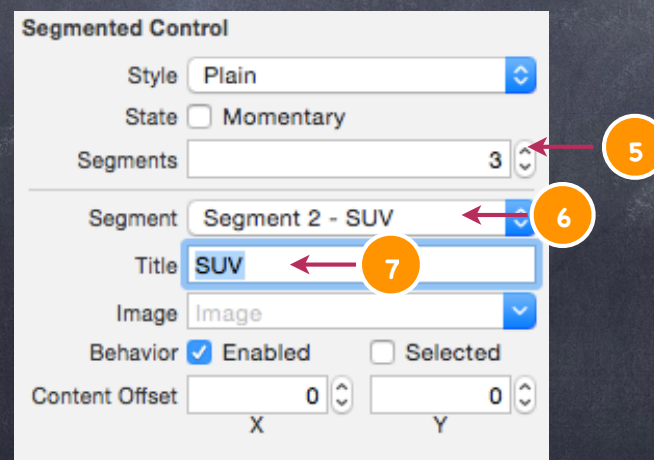
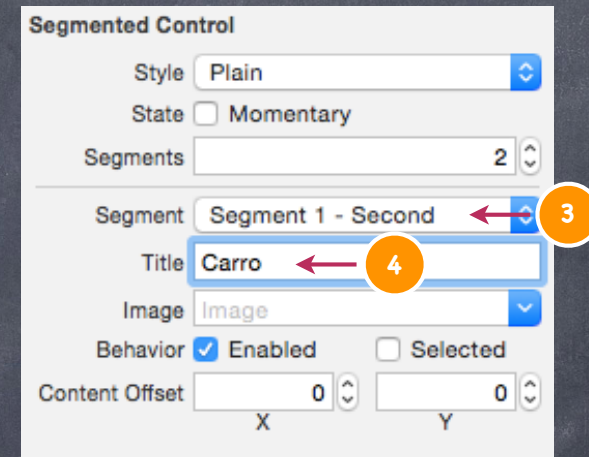
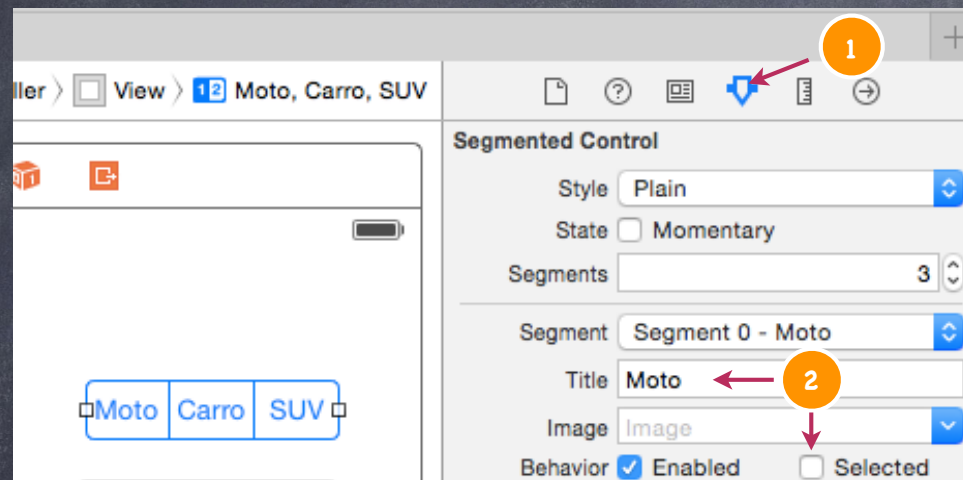
Segmented Control

- Coloque em sua tela um objeto do tipo Segmented Control.



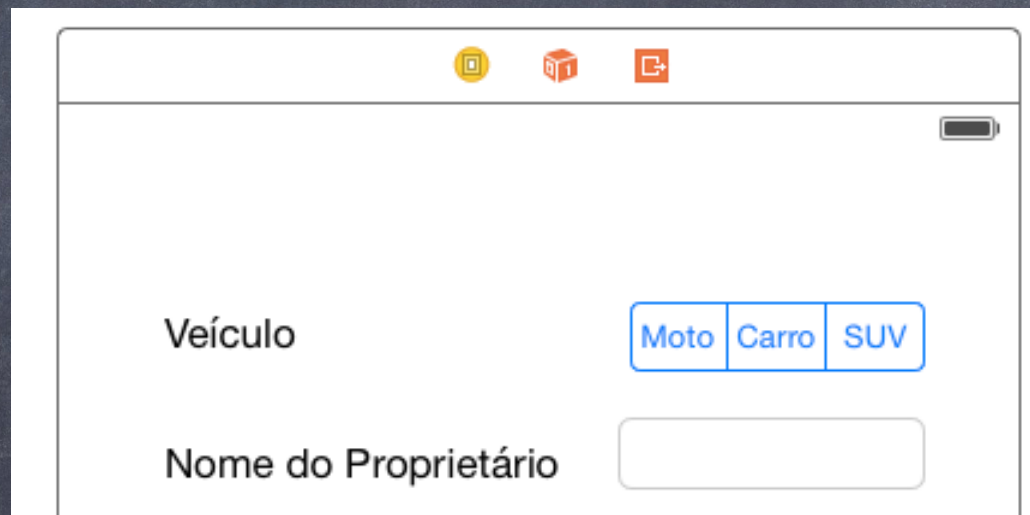
Segmented Control

- 1 Clique no Attributes Inspector (1) e altere os dados conforme a imagem abaixo:



Label's e Text

- Inclua em sua tela dois Label's (Veículo e Nome do Proprietário) e um Text conforme a figura abaixo:



The image shows a UI mockup of a form. At the top, there is a header bar with three icons: a yellow circle with a square, a red cube, and a red square with a plus sign. Below the header, there is a battery icon in the top right corner. The main content area contains two labels and a text input field. The first label is 'Veículo' and is followed by a button group with three buttons: 'Moto', 'Carro', and 'SUV'. The second label is 'Nome do Proprietário' and is followed by a text input field.

Stepper e Slider

- Inclua 4 Labels (Ano Modelo, KM mensal e dois label's) em frente de cada um, inclua também um Stepper para o Ano Modelo e um Slider para o KM mensal.

The image displays a mobile application interface on the left and its SwiftUI code on the right.

Mobile App Interface:

- Veículo:** Three buttons labeled "Moto", "Carro", and "SUV".
- Nome do Proprietário:** A text input field.
- Ano Modelo:** A label followed by a stepper control with minus and plus buttons.
- KM mensal:** A label followed by a slider control.

SwiftUI Code:

```
import SwiftUI

struct Form: View {
    @State private var vehicle: String = "Carro"
    @State private var name: String = ""
    @State private var year: Int = 2020
    @State private var km: Int = 0

    var body: some View {
        FormView()
    }
}

struct FormView: View {
    @State private var vehicle: String = "Carro"
    @State private var name: String = ""
    @State private var year: Int = 2020
    @State private var km: Int = 0

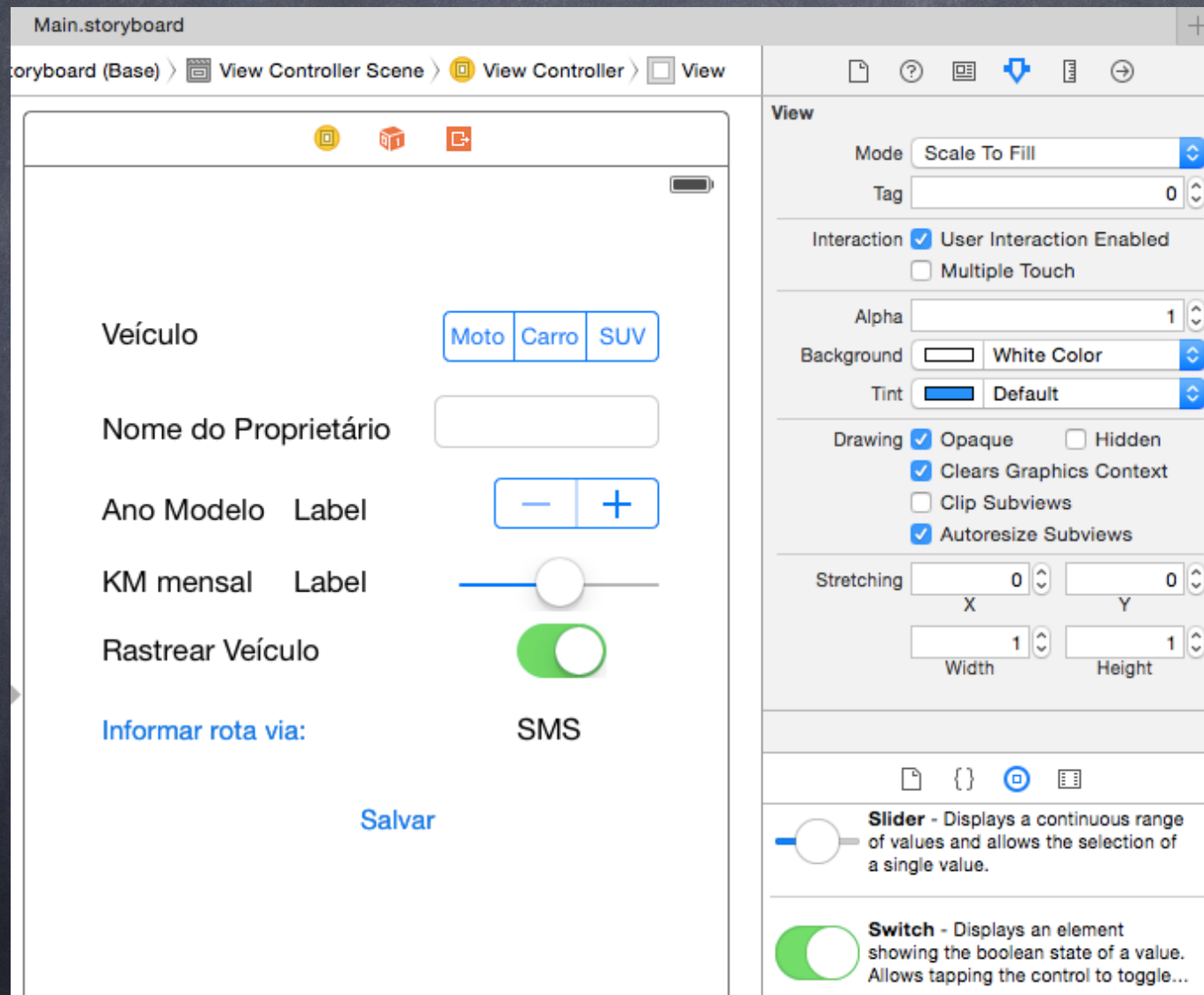
    var body: some View {
        VStack {
            Picker(selection: $vehicle, label: "Veículo", items: ["Moto", "Carro", "SUV"]) {
                Text("Moto")
                Text("Carro")
                Text("SUV")
            }
            TextField("Nome do Proprietário", text: $name)
            Stepper("Ano Modelo", value: $year, in: 2015...2025)
            Slider(value: $km, in: 0...100000) {
                Label("KM mensal", text: .display)
            }
        }
    }
}
```

Code Comments:

- Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.
- Slider** - Displays a continuous range of values and allows the selection of a single value.
- Switch** - Displays an element showing the boolean state of a value. Allows tapping the control to toggle...

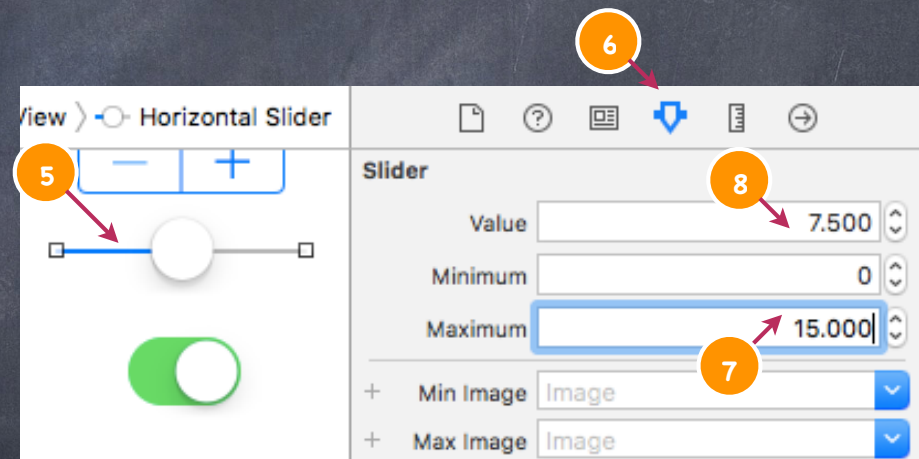
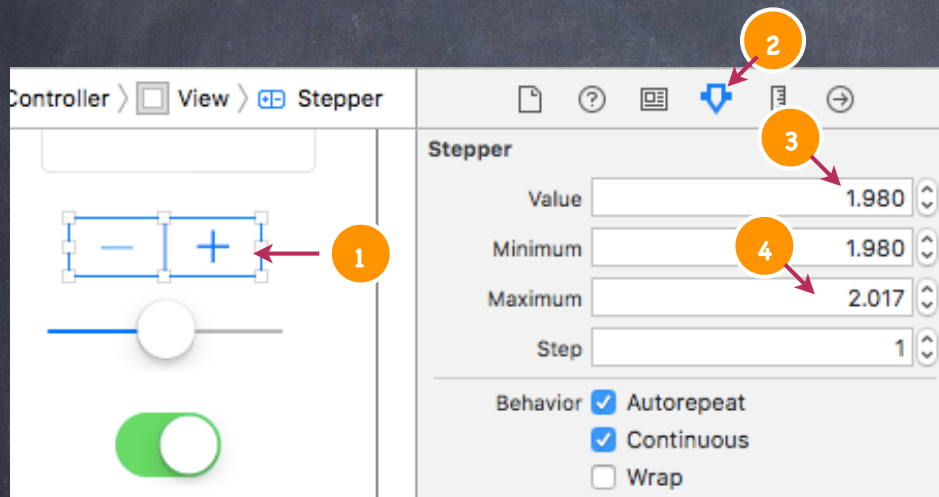
Switch

- Conforme a figura insira um Label Rastrear Veículo e um Switch, depois insira um botão com o texto: "Informar a rota via:" e em frente um outro Label escrito SMS, ao final insira um último botão com o texto Salvar.



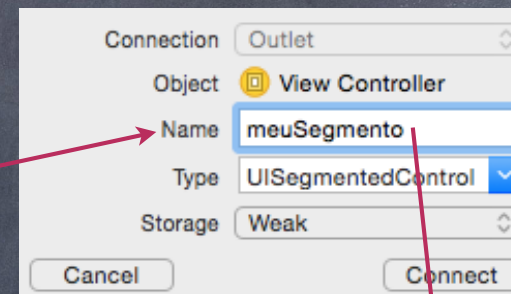
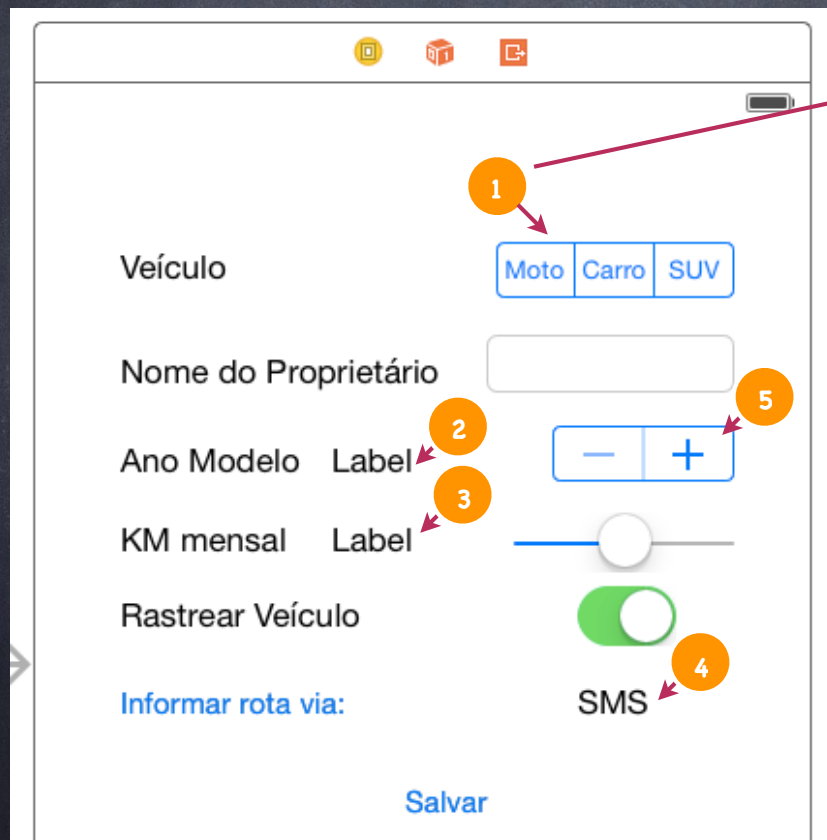
Stepper e Slider

- Selecione o Stepper e altere as propriedades Value Minimum e Value Maximum, depois selecione o Slider e altere as mesmas propriedades conforme a imagem abaixo:



Outlet's no viewController.swift

- No arquivo .swift crie os Outlet's dos componentes numerados, veja na imagem os seus respectivos nomes.



```

1 //
2 // ViewController.swift
3 // UIKit Swift
4 //
5 // Created by agesandro scarpioni on 14/02/16.
6 // Copyright © 2016 Agesandro Scarpioni. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var meuSegmento: UISegmentedControl!
14
15     @IBOutlet weak var lblAno: UILabel!
16
17     @IBOutlet weak var lblKm: UILabel!
18
19     @IBOutlet weak var lblRota: UILabel!
20
21     @IBOutlet weak var meuStepper: UIStepper!
22
23 }
  
```

Red arrows from callouts 1 through 5 point to the corresponding outlet declarations in the code:

- 1** points to `meuSegmento: UISegmentedControl!`
- 2** points to `lblAno: UILabel!`
- 3** points to `lblKm: UILabel!`
- 4** points to `lblRota: UILabel!`
- 5** points to `meuStepper: UIStepper!`

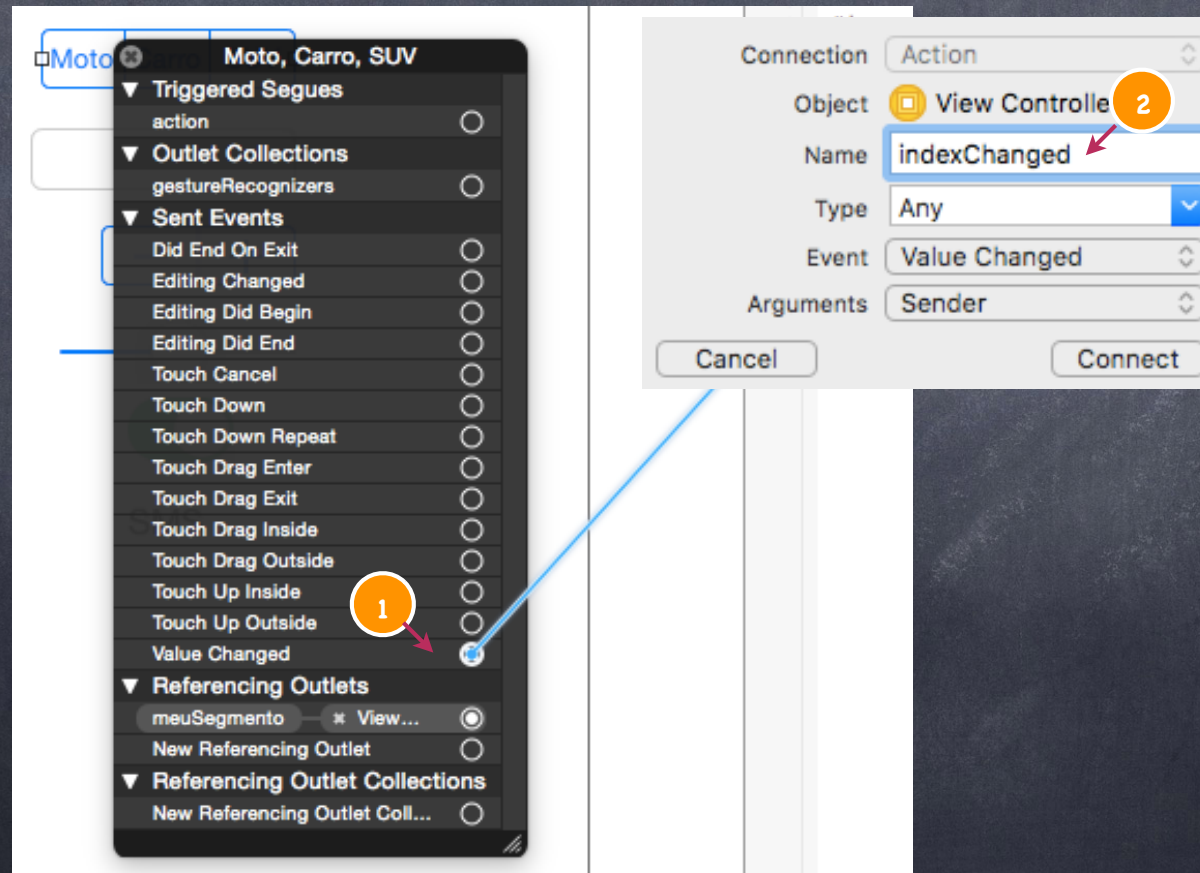
Arquivo .swift

- No método viewDidLoad escreva as linhas abaixo, ao executarmos o programa as informações nos label's irão aparecer.

```
23  
24     override func viewDidLoad() {  
25         super.viewDidLoad()  
26         lblAno.text = "1980"  
27         lblKm.text = "7500"  
28     }  
29
```


IBAction do Segmented Control

- Clique com o botão direito sobre o Segmented Control escolha Value Changed(1), arraste para dentro da área da class ViewController.swift, em Name (2) digite indexChanged.



Segmented Control

- Para recuperar qual botão foi clicado usamos a propriedade `selectedSegmentIndex`, podemos fazer com comando `if` ou `switch`, optei por fazer com `switch`, observe que na terceira opção (case 2) decidi mostrar o texto do `SegmentedControl`, para recuperarmos o tal texto usamos o método `titleForSegmentAtIndex` e informamos o `selectedSegmentIndex`.

```
30
31 @IBAction func indexChanged(_ sender: Any) {
32     switch (meuSegmento.selectedSegmentIndex){
33     case 0:
34         print("Você escolheu o primeiro botão")
35         break
36     case 1:
37         print("Você escolheu o segundo botão")
38         break
39     case 2:
40         print("Você escolheu o \(meuSegmento.titleForSegment(at: meuSegmento.selectedSegmentIndex)!)" )
41         break
42     default:
43         break
44     }
45 }
46
```


Executando o App

- Command + R para executar, veja as informações que já aparecem nos label's (1) viaDidLoad e os resultados do NSLog no Console (2) clicando nos botões Carro, Moto e SUV.

The screenshot shows a user interface for a vehicle tracking application. At the top, there are three buttons labeled 'Moto', 'Carro', and 'SUV'. The 'SUV' button is highlighted in blue. Below these buttons, there is a text field for 'Nome do Proprietário'. Underneath that is a numeric input for 'Ano Modelo' with the value '1980' and two buttons, '-' and '+', for adjusting the value. Below the year input is a slider for 'Km mensal' with the value '7500'. Further down is a toggle switch for 'Rastrear Veículo', which is currently turned on (green). At the bottom left, there is a blue link that says 'Informar rota via'. At the bottom right, there is a blue button that says 'Salvar'. A red arrow points from the 'SUV' button to a console window on the right.

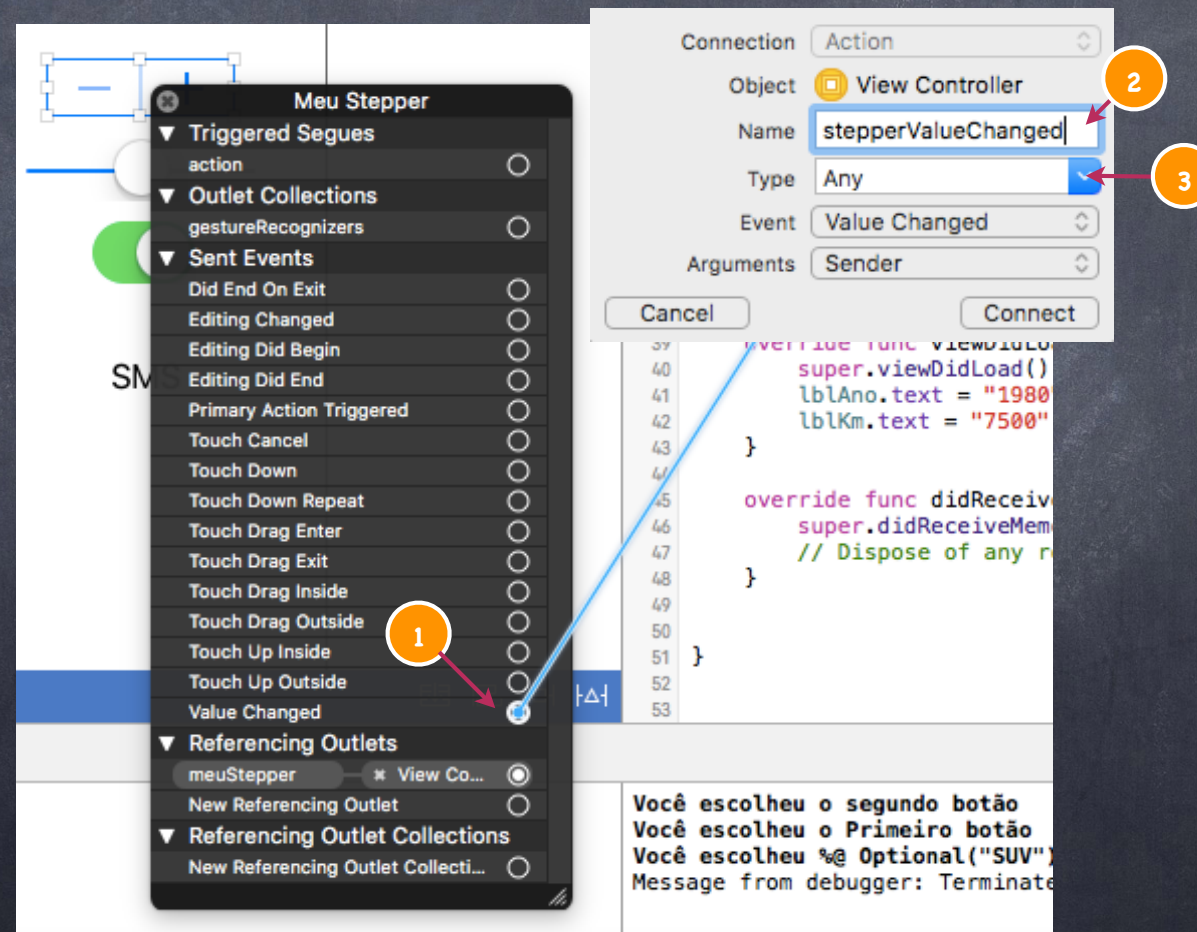
2

```
Você escolheu o primeiro botão  
Você escolheu o segundo botão  
Você escolheu o SUV
```


IBAction no Arquivo .swift

do Stepper

- Clique com o botão direito sobre o Stepper, escolha Value Changed(1), em Name (2) digite stepperValueChanged em Type escolha Any (3).



Implementação do Stepper

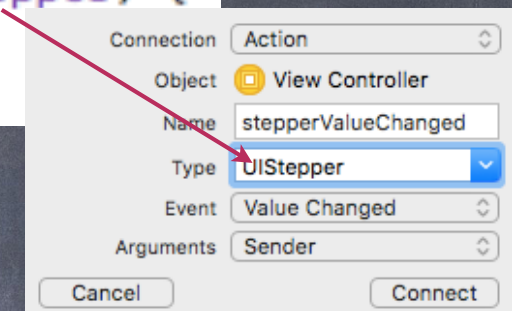
- A implementação do Stepper é muito simples, basta passar o value do stepper para o lblAno, porém, como o lblAno é texto e o value do Stepper é um Double, temos que utilizar o Int para que não apareça casas decimais, como ainda é um número precisamos converter para String utilizando "\\".
- Lembre-se que meuStepper é o nome do Outlet que demos para o Stepper e lblAno foi o nome do Outlet que demos para o Label, stepperValueChanged é o nome que demos para o IBAction do Stepper.

```
38  
39 @IBAction func stepperValueChanged(sender: AnyObject) {  
40     lblAno.text = "\\(Int(meuStepper.value))"  
41 }  
42
```


Implementação do Stepper

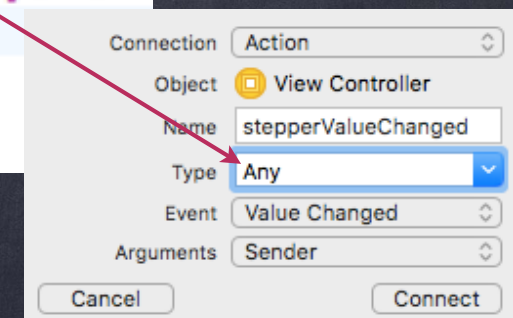
- Uma outra possibilidade é não criar o Outlet com o nome de meuStepper, e ao criarmos o Action ao invés de selecionarmos no Type o tipo Any selecionamos UIStepper e a assinatura ficaria da seguinte forma:

```
47  
48 @IBAction func stepperValueChanged(_ sender: UIStepper) {  
49     lblAno.text = "\(Int(sender.value))"  
50 }
```



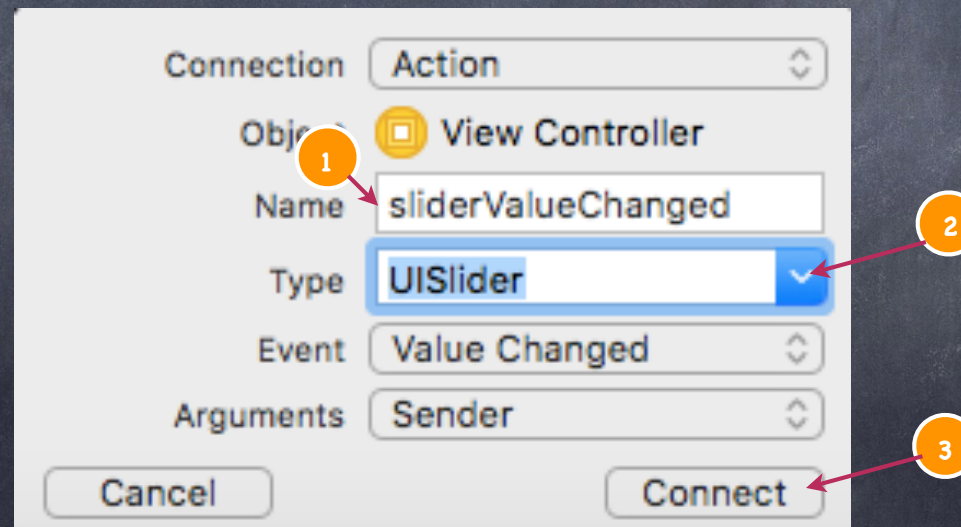
- Compare essas duas imagens, abaixo usamos meuStepper.value, na imagem acima utilizamos sender.value, faremos da segunda forma com o objeto Slider.

```
46  
47 @IBAction func stepperValueChanged(_ sender: Any) {  
48     lblAno.text = "\(Int(meuStepper.value))"  
49  
50 }
```



IBAction do Slider

- 1 Clique com o botão direito sobre o Slider, escolha Value Changed leve até o arquivo.swift, em Name (1) digite sliderValueChanged, troque o type (2) de Any para UISlider.
- 2 Quando trocamos o tipo, não precisamos criar um Outlet do componente para enviarmos o value, veja a programação no próximo slide e compare com a programação no slide anterior onde o tipo era Any.



Implementação do Slider

- A implementação do Slider é simples como o Stepper, basta ir no arquivo.swift e passar o value do Slider para o lblKm, veja que como mudamos o type, ao invés de utilizarmos o nome de um Outlet para o slider usamos sender.value.

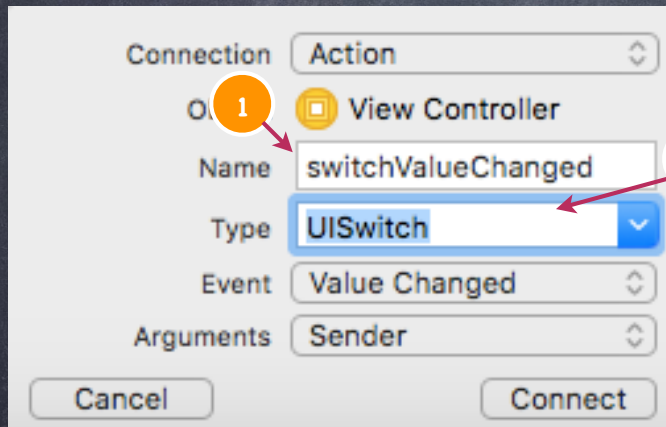
```
52  
53 @IBAction func sliderValueChanged(_ sender: UISlider) {  
54     lblKm.text = "\(Int(sender.value))"  
55 }  
56
```

- Lembre-se que o intervalo do slider foi alterado para o máximo 15000, já o Current foi ajustado para 7500, isso ocorreu para posicionar o botão do slider no centro.

The screenshot shows the Xcode Interface Builder for a UISlider. The 'Slider' section is expanded, showing three input fields: 'Value' (7.500), 'Minimum' (0), and 'Maximum' (15.000). Red arrows point to the 'Value' and 'Maximum' fields, which are labeled with orange circles containing the numbers 1 and 2 respectively.

IBAction do Switch no arquivo.swift

- Clique com o botão direito sobre o Switch, escolha Value Changed leve até o arquivo.swift, em Name (1) digite switchValueChanged, troque o type (2) de Any para UISwitch. Depois vá até o arquivo.m (3) e implemente, execute o programa e veja o console (4), transforme o sender para inteiro (5) e veja o resultado (6).



```
57  
58 @IBAction func switchValueChanged(_ sender: UISwitch) {  
59     print(sender.isOn)  
60 }  
61
```

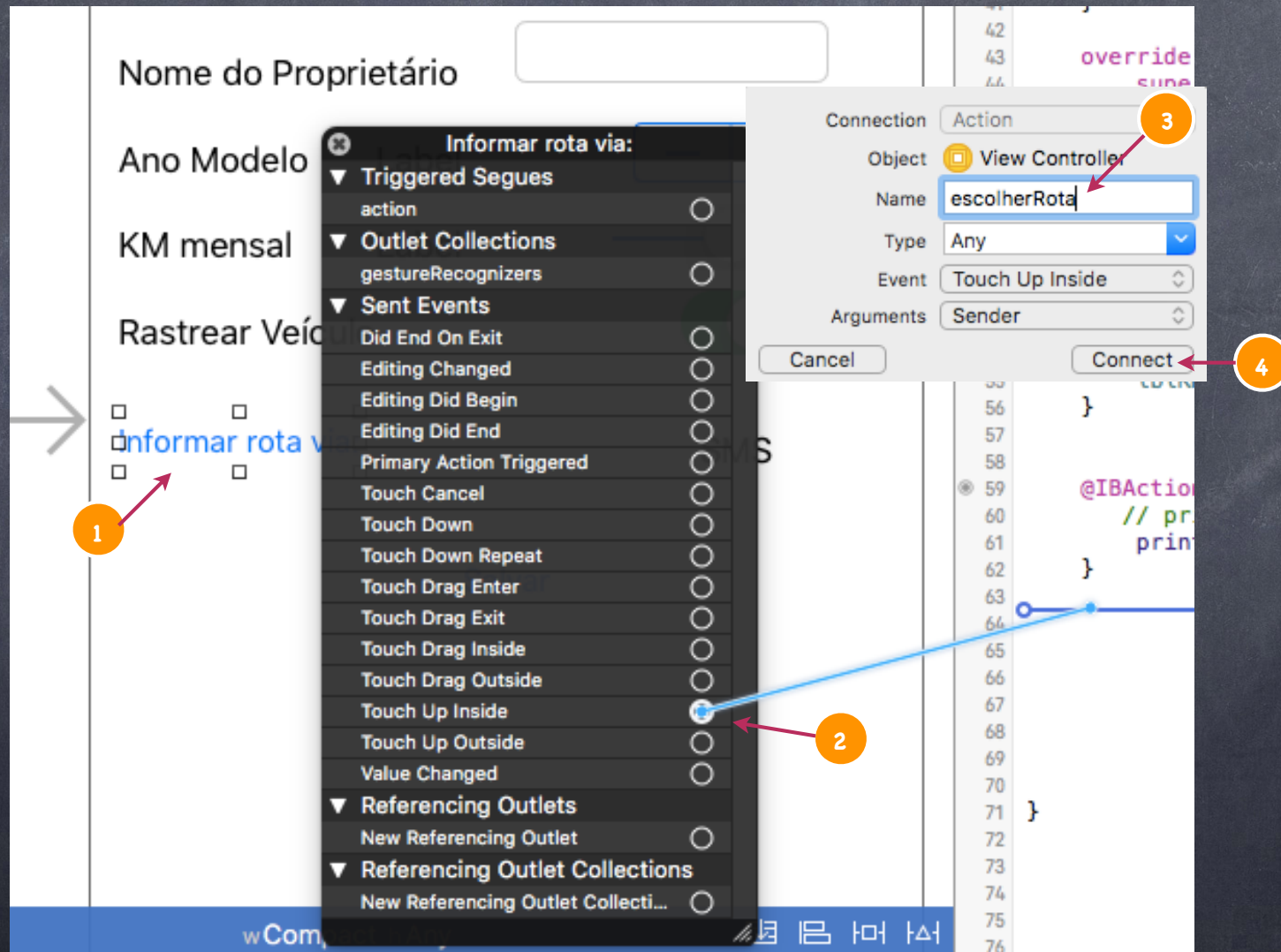
```
false  
true  
  
All Output
```

```
57  
58 @IBAction func switchValueChanged(_ sender: UISwitch) {  
59     //print(sender.isOn)  
60     print(Int(NSNumber(value:sender.isOn)))  
61 }  
62
```

```
0  
1  
  
All Output
```


IBAction do botão

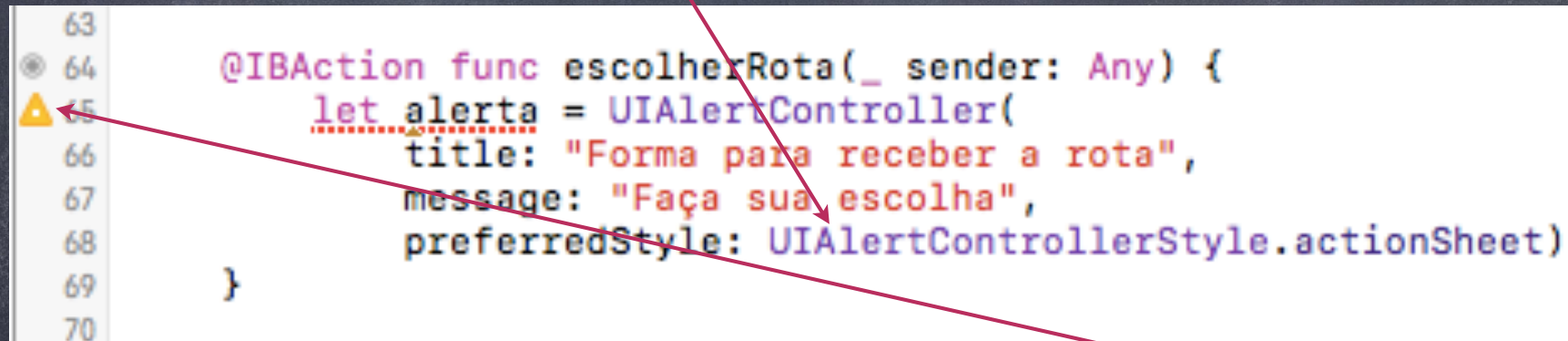
"informar rota via" no arquivo.swift



Implementação

- Inicie a implementação da IBAction EscolherRota, nas linhas abaixo vamos criar uma constante chamada alerta. Para informar o .actionSheet você deverá dar 2 cliques na palavra UIAlertControllerStyle e depois digitar o .actionSheet, ou apenas digite .actionSheet sem o UIAlertControllerStyle

```
63
64 @IBAction func escolherRota(_ sender: Any) {
65     let alerta = UIAlertController(
66         title: "Forma para receber a rota",
67         message: "Faça sua escolha",
68         preferredStyle: UIAlertControllerStyle.actionSheet)
69 }
70
```



- Não se preocupe com o Warning que irá aparecer nesse ponto, ao continuar o código no próximo slide esse aviso irá desaparecer

Implementação (Continuação)

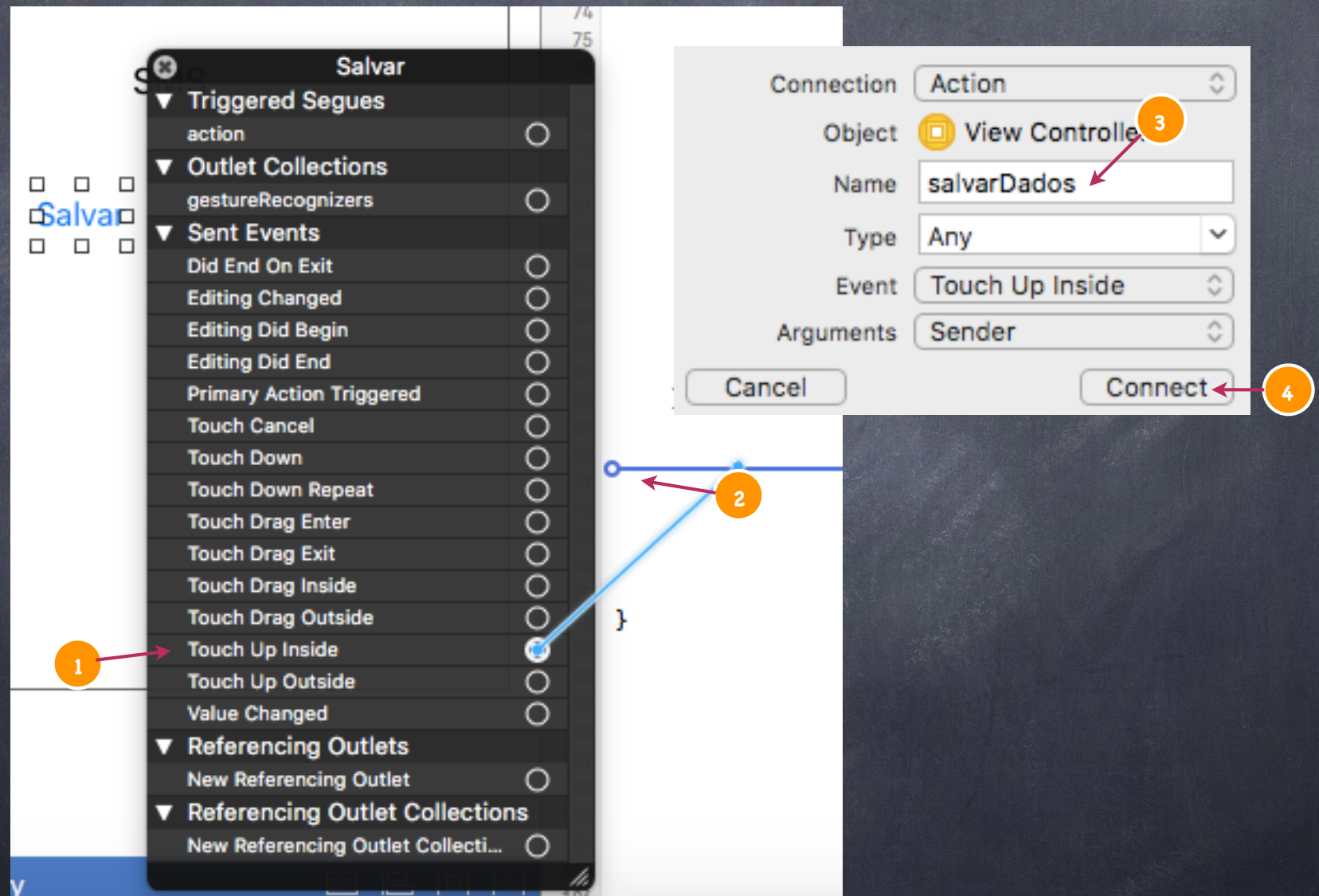
```
63
64  @IBAction func escolherRota(_ sender: Any) {
65      let alerta = UIAlertController(
66          title: "Forma para receber a rota",
67          message: "Faça sua escolha",
68          preferredStyle: UIAlertControllerStyle.actionSheet)
69
70      alerta.addAction(UIAlertAction(
71          title: "SMS",
72          style: UIAlertActionStyle.default,
73          handler: { action in
74              self.lblRota.text = action.title}))
75
76      alerta.addAction(UIAlertAction(
77          title: "E-mail",
78          style: UIAlertActionStyle.default,
79          handler: { action in
80              self.lblRota.text = action.title}))
81
82      alerta.addAction(UIAlertAction(
83          title: "Cancelar",
84          style: UIAlertActionStyle.cancel,
85          handler: nil))
86
87      present(alerta, animated: true, completion: nil)
88
89  }
```


Execute - Command + R

The image shows a screenshot of an iPhone app interface. The top status bar displays 'Carrier', signal strength, '11:30 AM', and battery level. The app's main form includes a 'Veículo' section with buttons for 'Moto', 'Carro', and 'SUV'. Below this is a 'Nome do Proprietário' field. The 'Ano Modelo' is set to '1980' with minus and plus buttons. 'Km mensal' is set to '7500' with a slider. A 'Rastrear Veículo' toggle switch is turned on. There is a 'Informar rota via' section with a 'Label' field. A 'Salvar' button is at the bottom of the form. An action sheet is displayed at the bottom of the screen, titled 'Forma para receber a rota' with the subtitle 'Faça sua escolha'. It contains three options: 'SMS', 'E-mail', and 'Cancelar'.

Obs: Veja que esse tipo de mensagem (.actionSheet) é exibida na parte inferior da tela do iPhone.

IBAction do botão "Salvar" no arquivo.swift

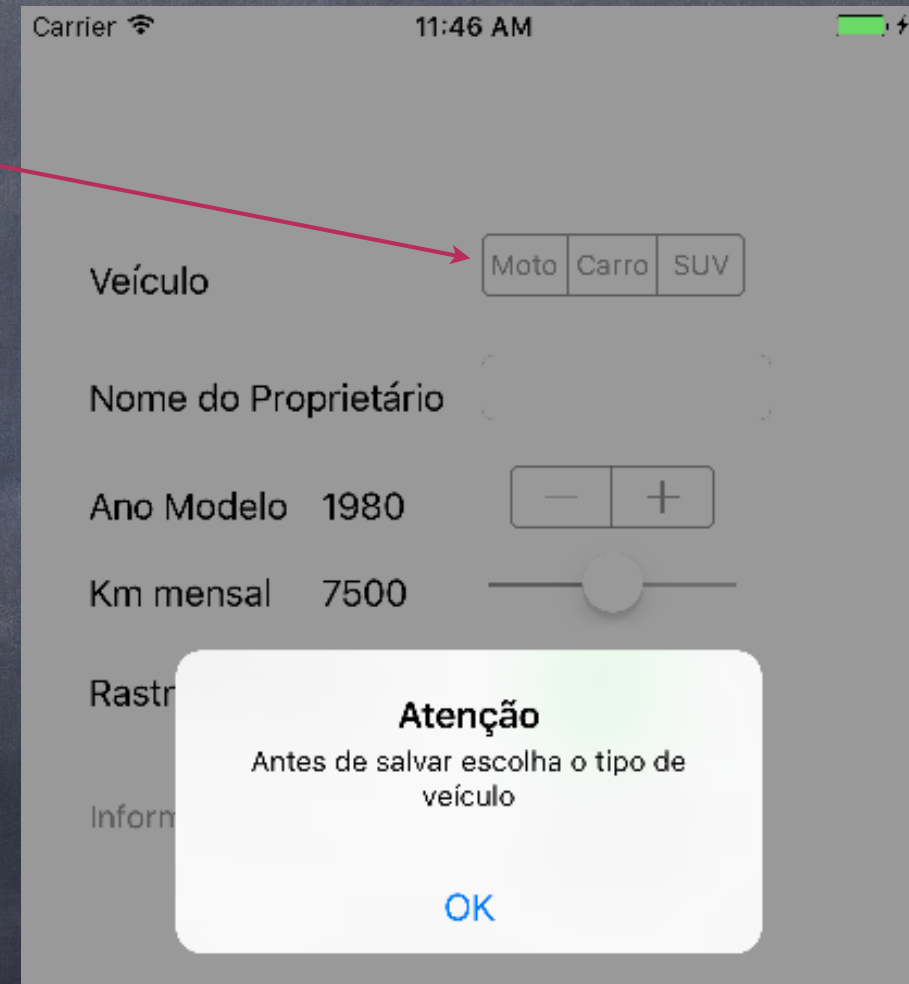
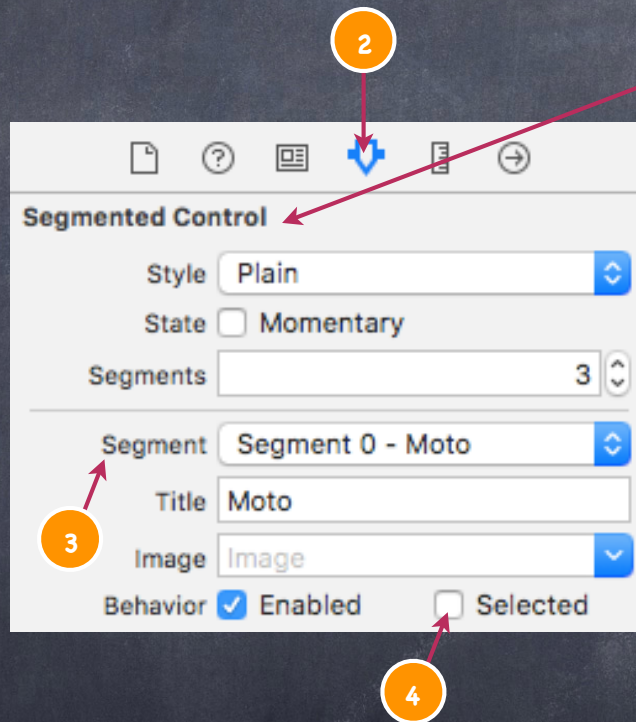


Implementação do Botão Salvar

```
90
91  @IBAction func salvarDados(_ sender: Any) {
92      var msg: String
93      var tipo: String
94
95      if meuSegmento.selectedSegmentIndex >= 0{
96          tipo = meuSegmento.titleForSegment(at: meuSegmento.selectedSegmentIndex)!
97          msg = "Veículo \(tipo) do ano \(lblAno.text!) salvo com sucesso"
98      }else{
99          tipo = ""
100          msg = "Antes de salvar escolha o tipo de veículo"
101      }
102
103      let alerta = UIAlertController(
104          title: "Atenção",
105          message: msg,
106          preferredStyle: UIAlertControllerStyle.alert)
107
108      alerta.addAction(UIAlertAction(
109          title: "OK",
110          style: UIAlertActionStyle.default,
111          handler: nil))
112
113      present(alerta, animated: true, completion: nil)
114
115  }
116
117
```


Command + R

- Caso você não tenha observado na página 6, o Segmented Control deve ficar sem tipos de veículo selecionado, selecione o Segmented Control (1), vá em Attributes Inspector(2), selecione o segment 0 (3) desmarque a opção Selected(4) para que nosso IF funcione perfeitamente.



Obs: Veja que esse tipo de mensagem (.alert) é exibida na centro da tela do iPhone.