

UIKit e Alertas

Conhecendo outros componentes

X-Code com ObjC

Prof. Agesandro Scarpioni

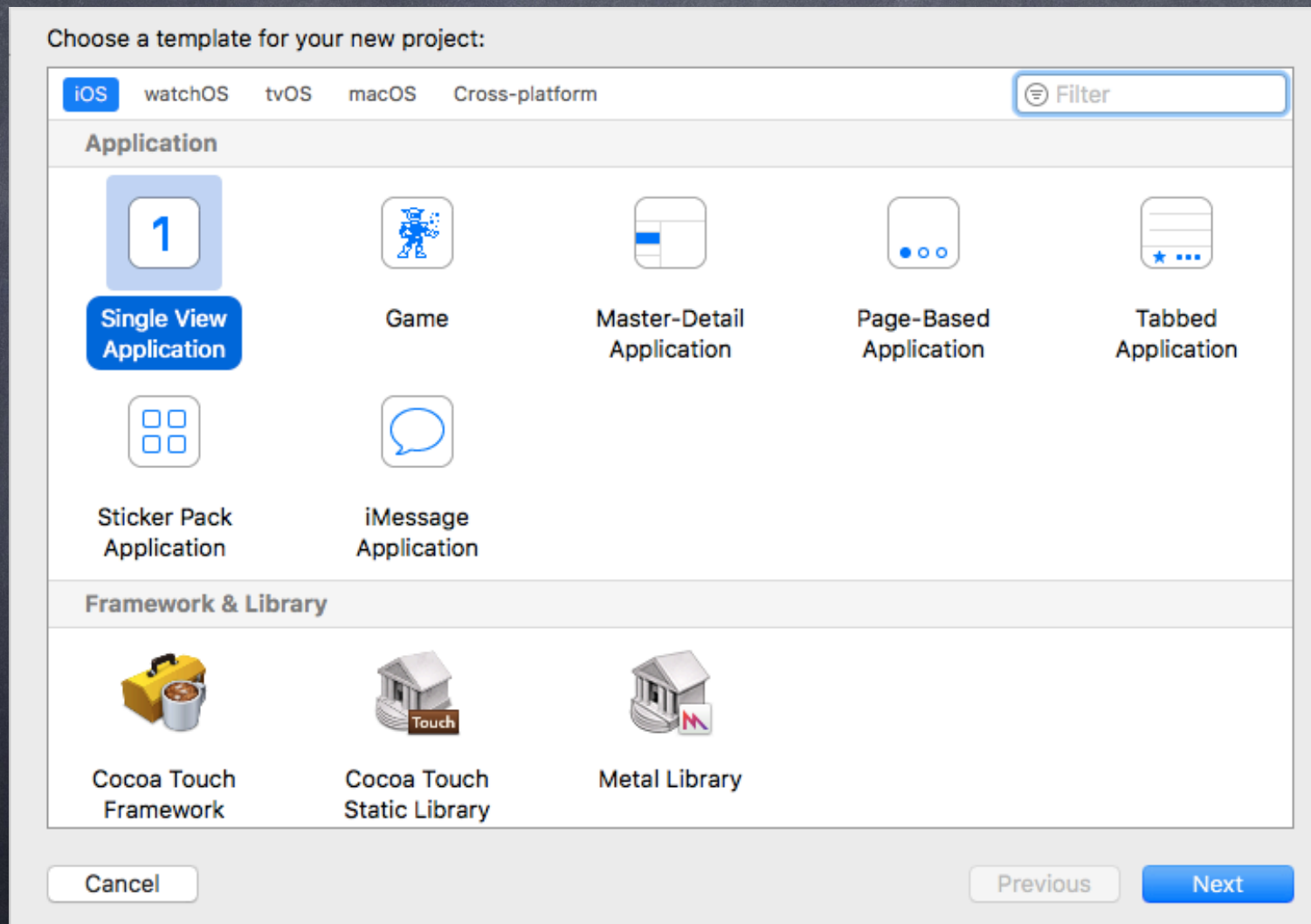
agesandro@fiap.com.br

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 e na linguagem escolha Objective-C.

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

☐ Use Core Data

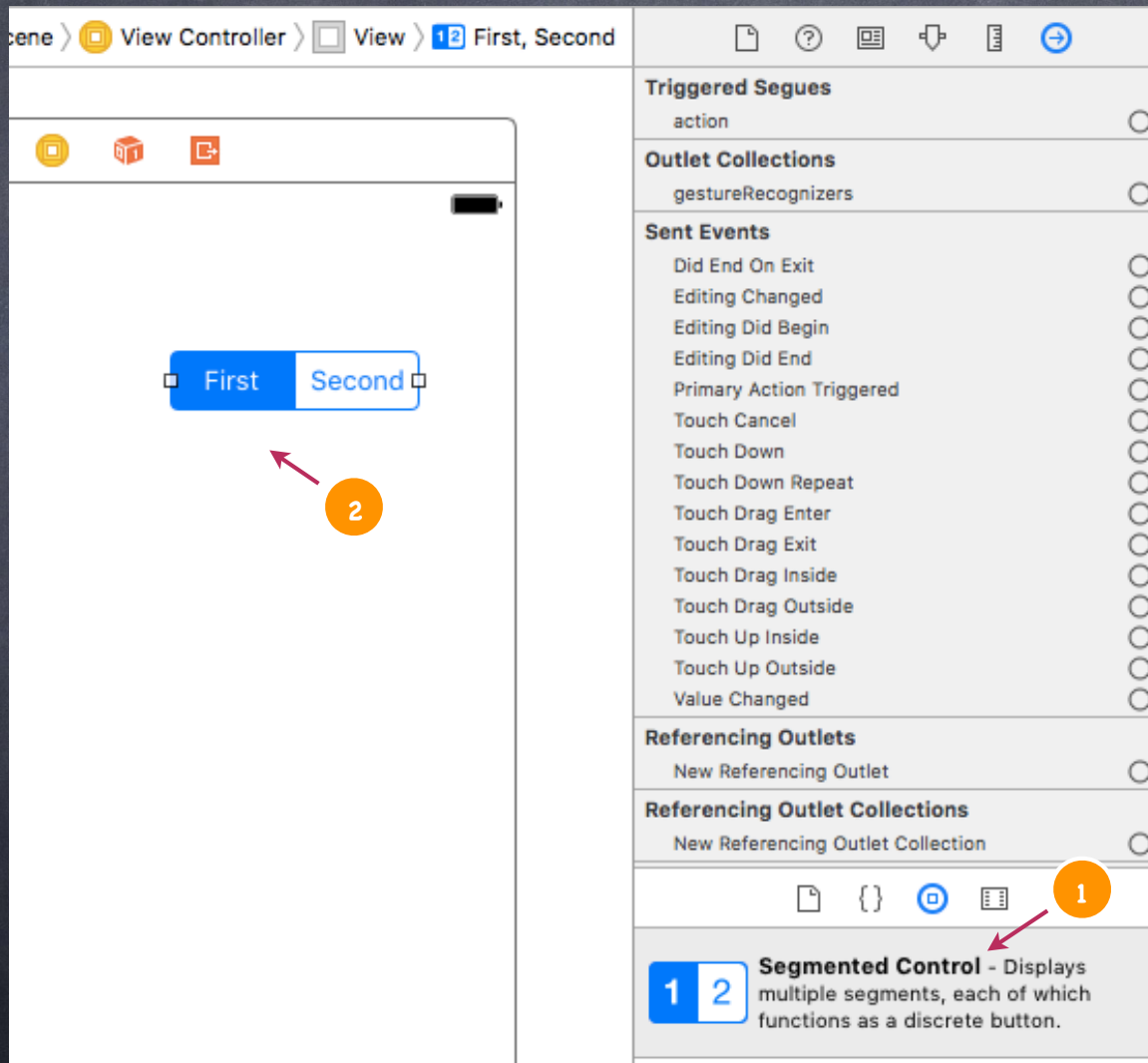
☐ Include Unit Tests

☐ Include UI Tests

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

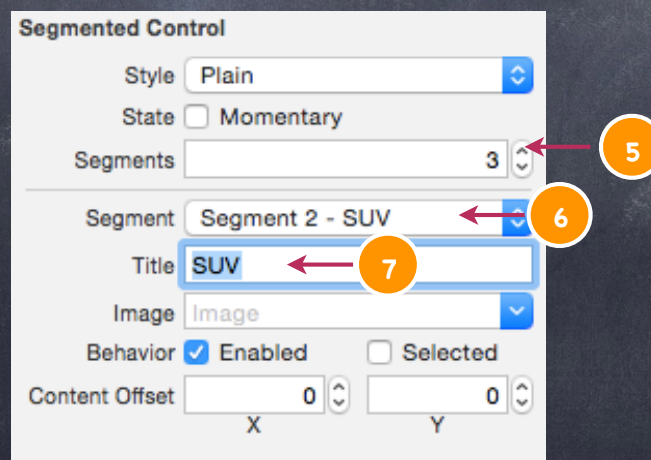
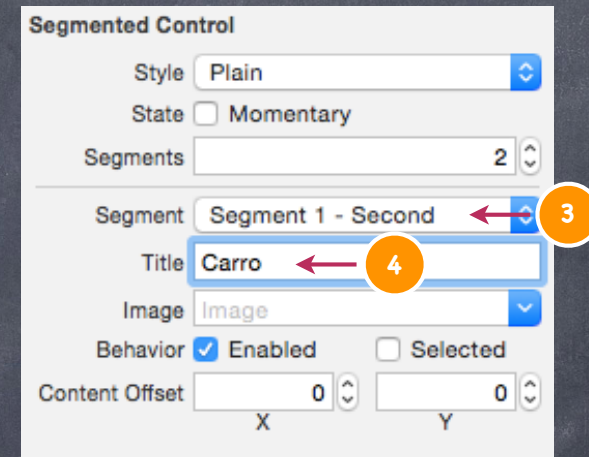
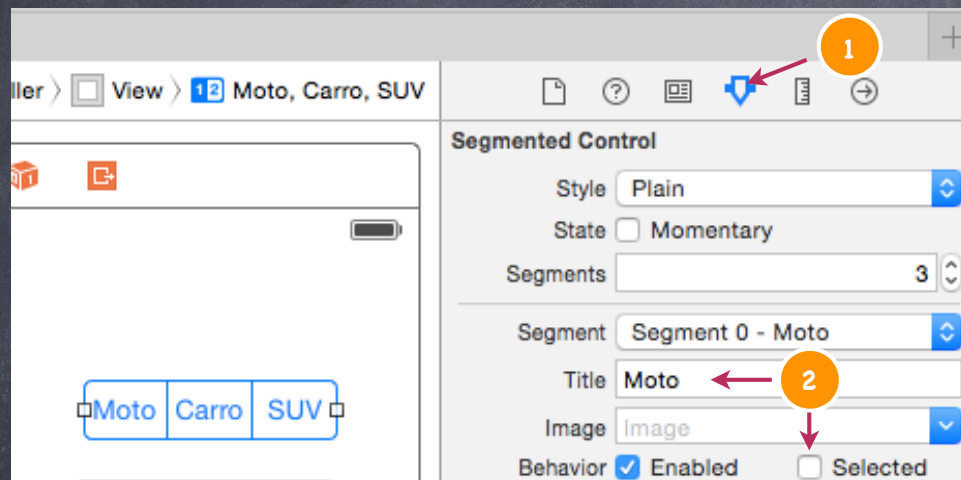
Segmented Control

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



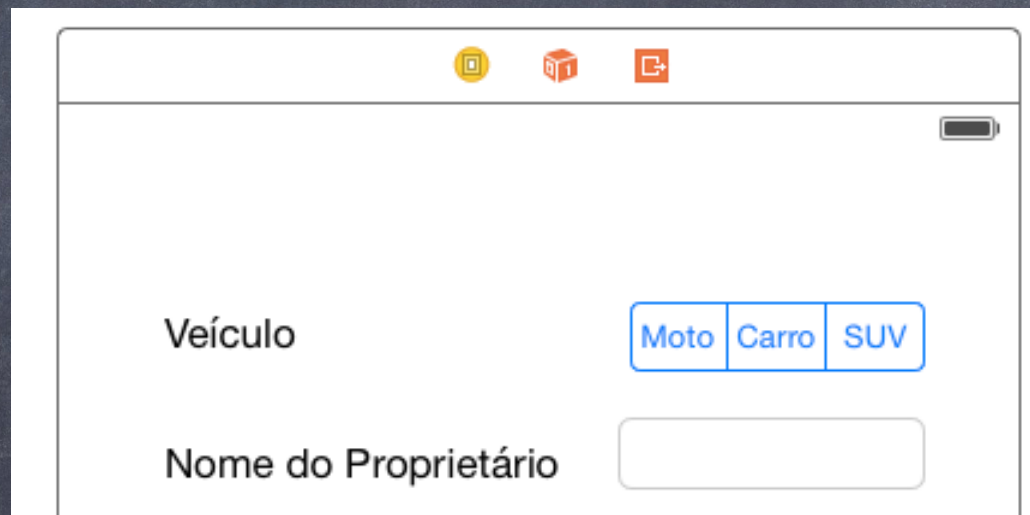
Segmented Control

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



Label's e Text

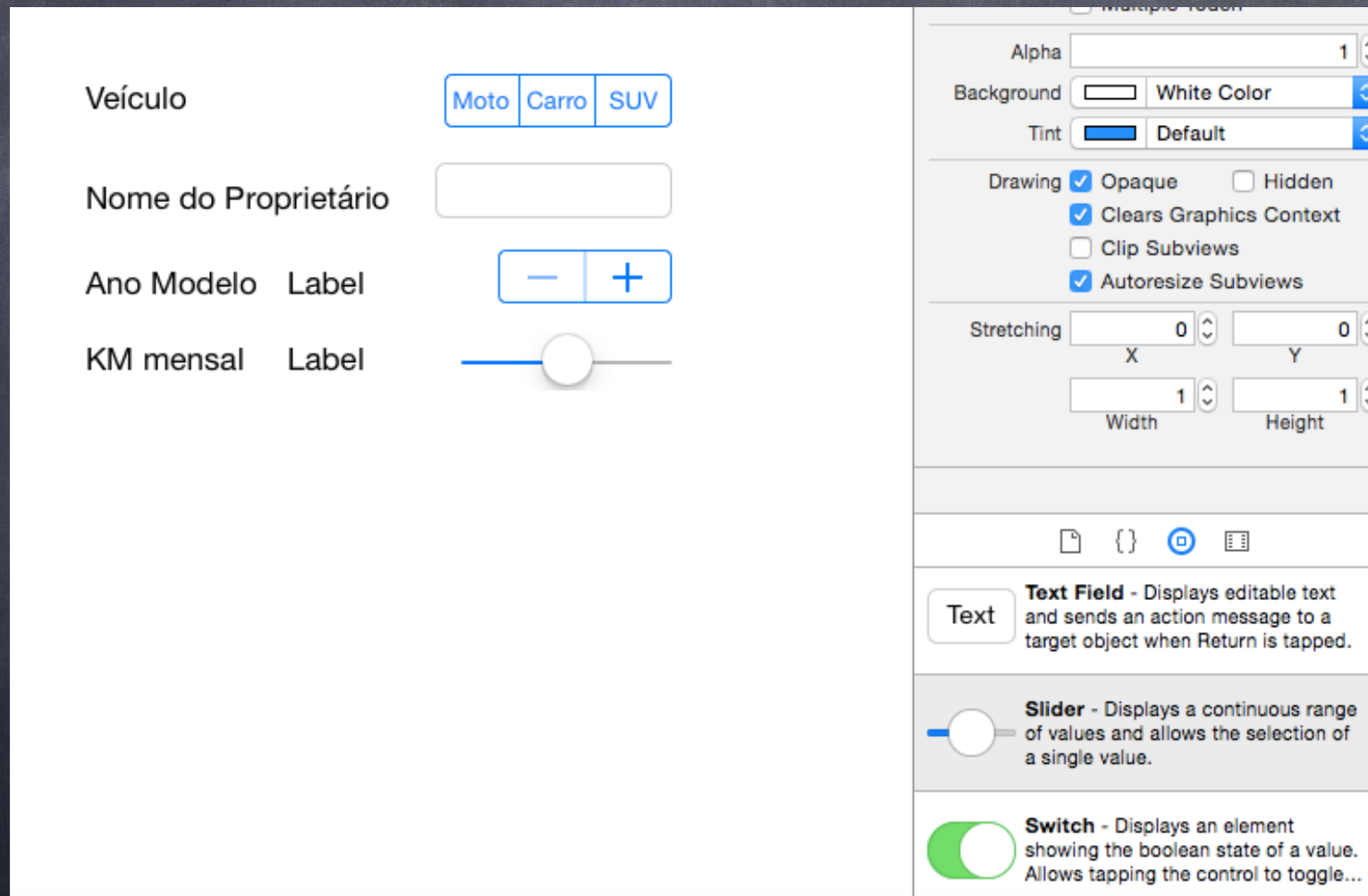
- Inclua em sua tela dois Label's (Veículo e Nome do Proprietário) e um Text conforme a figura abaixo:



The image shows a UI mockup within a window frame. At the top of the window, there are three icons: a yellow circle with a square inside, a red cube, and a red square with a white plus sign. Below these icons, on the right side, is a battery status icon. The main content area contains two labels and a text input field. The first label is 'Veículo' and is followed by three buttons labeled 'Moto', 'Carro', and 'SUV'. The second label is 'Nome do Proprietário' and is followed by a text input field.

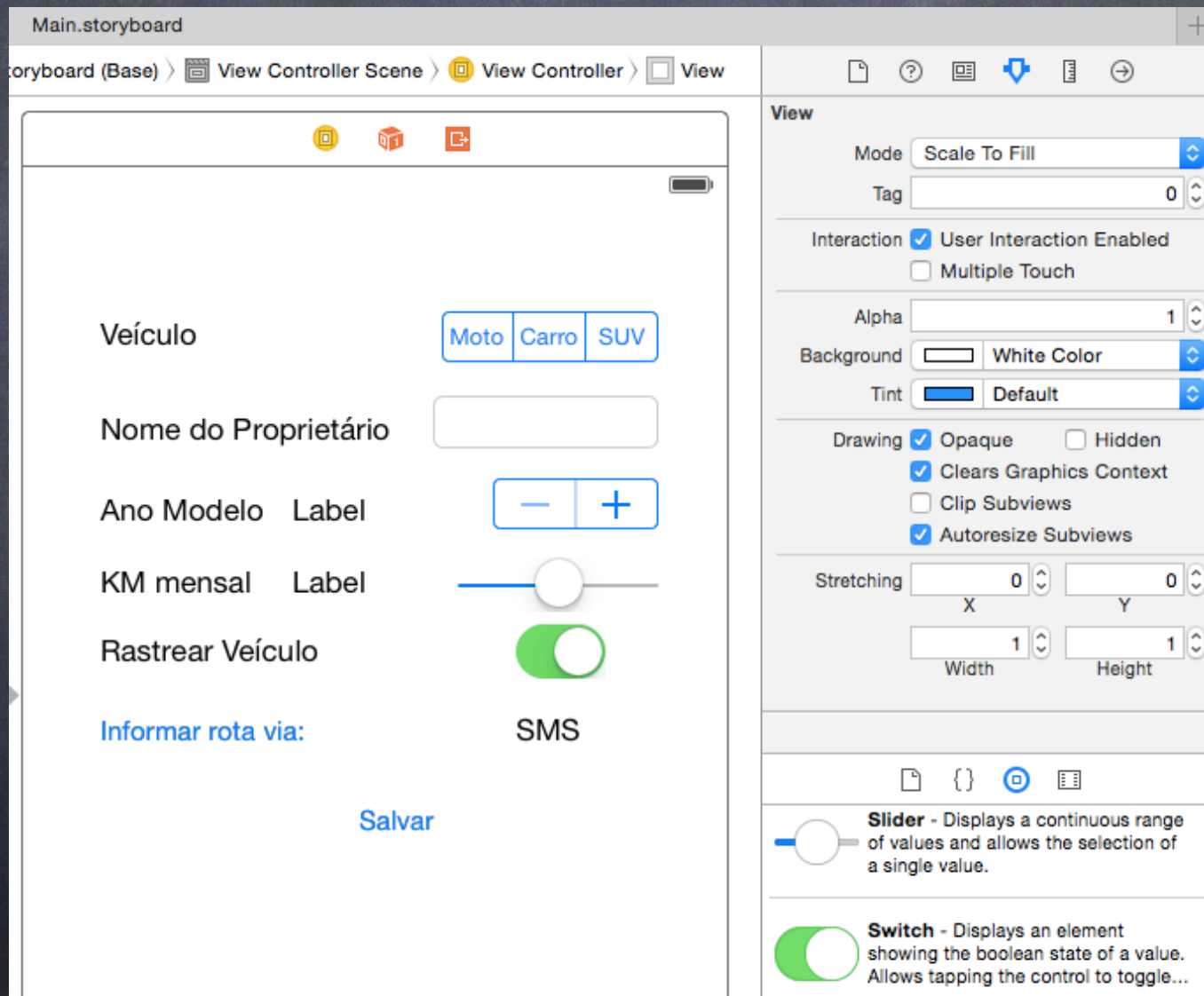
Stepper e Slider

- 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.



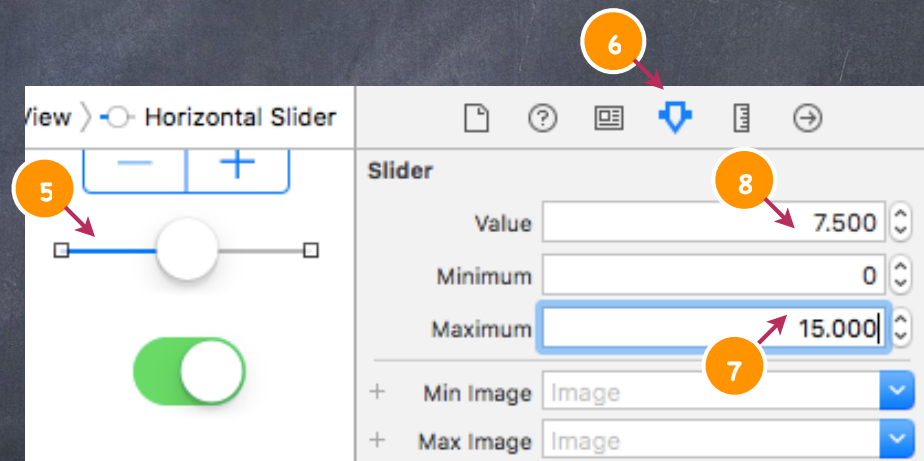
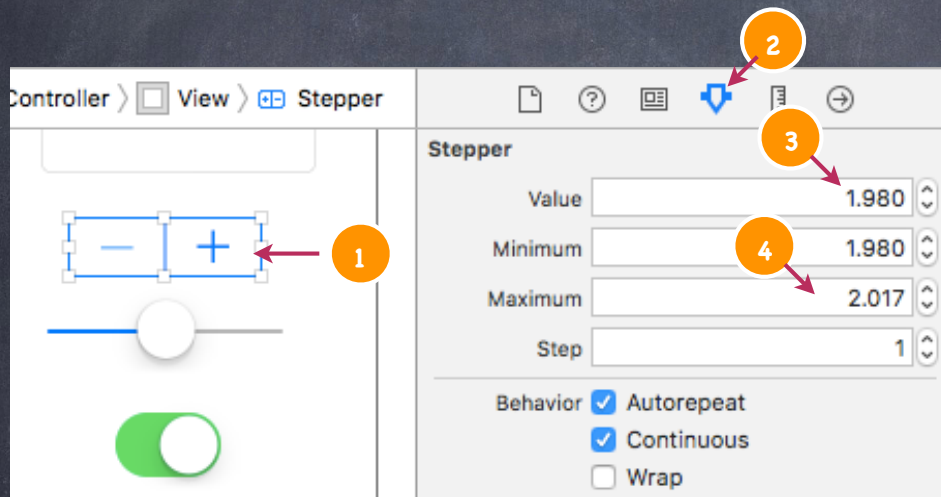
Switch

- 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.



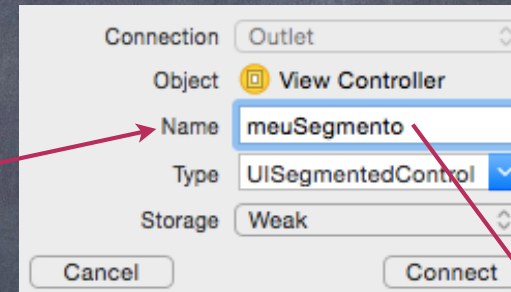
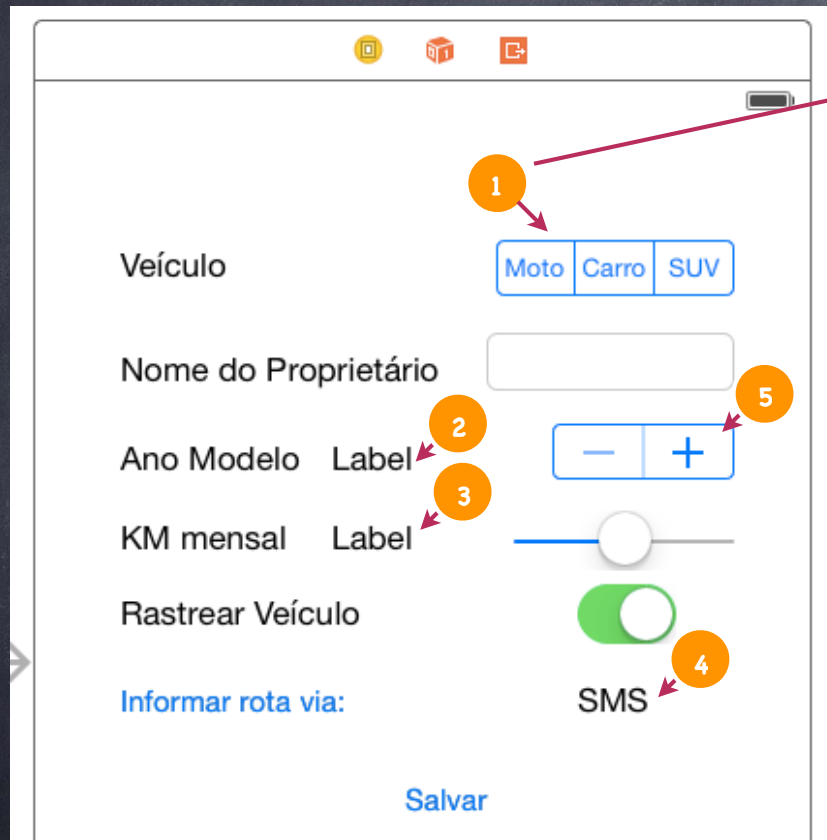
Stepper e Slider

- 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 Arquivo .h

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



```
1 //
2 // ViewController.h
3 // UIKit
4 //
5 // Created by agesandro scarpioni on 01/07/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <UIKit/UIKit.h>
10
11 @interface ViewController : UIViewController
12
13
14 @property (weak, nonatomic) IBOutlet UISegmentedControl *meuSegmento;
15
16 @property (weak, nonatomic) IBOutlet UILabel *lblAno;
17
18 @property (weak, nonatomic) IBOutlet UILabel *lblKm;
19
20 @property (weak, nonatomic) IBOutlet UILabel *lblRota;
21
22 @property (weak, nonatomic) IBOutlet UIStepper *meuStepper;
23
```

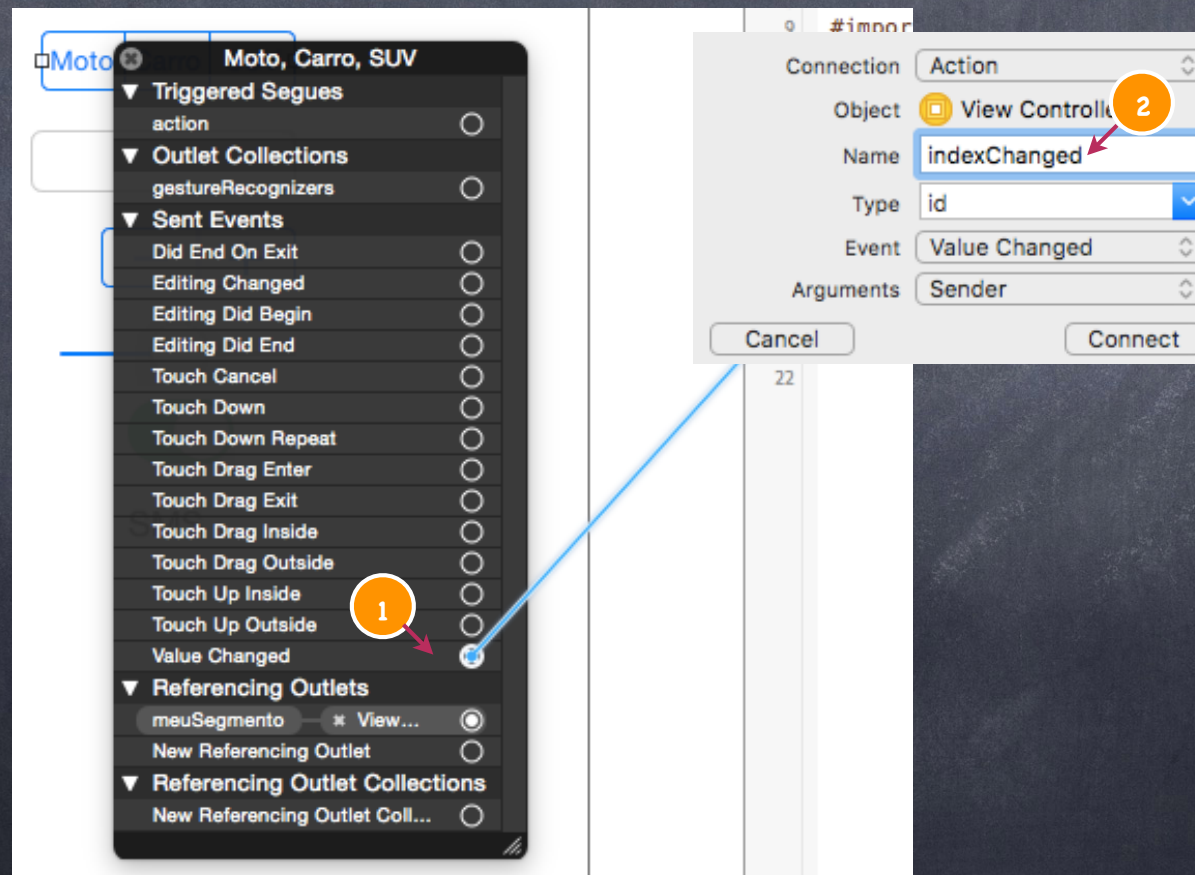

Arquivo .m

- No método viewDidLoad escreva as linhas 19 e 20 para que quando o programa for executado já apareça as informações nos labels.

```
1 //
2 // ViewController.m
3 // UIKit OBJC
4 //
5 // Created by Agesandro Scarpioni on 19/02/17.
6 // Copyright © 2017 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     self.lblAno.text = @"1980";
20     self.lblKm.text = @"7500";
21 }
22
23 - (void)didReceiveMemoryWarning {
24     [super didReceiveMemoryWarning];
25     // Dispose of any resources that can be recreated.
26 }
27
```


IBAction do Segmented Control no .h

- Clique com o botão direito sobre o Segmented Control, escolha Value Changed(1), em Name (2) digite indexChanged.



Arquivo .m

- Para recuperar qual botão foi clicado usamos a propriedade `selectedIndex`, 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 `selectedIndex`.

```
28
29 - (IBAction)indexChanged:(id)sender {
30     switch (self.meuSegmento.selectedIndex) {
31         case 0:
32             NSLog(@"Você escolheu o primeiro botão");
33             break;
34         case 1:
35             NSLog(@"Você escolheu o segundo botão");
36             break;
37         case 2:
38             NSLog(@"Você escolheu %@", [self.meuSegmento titleForSegmentAtIndex:self.meuSegmento.selectedIndex]);
39             break;
40         default:
41             break;
42     }
43 }
```


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.

Veículo Moto Carro SUV

Nome do Proprietário

Ano Modelo 1980 — +

KM Mensal 7500 1

Rastrear Veículo ☒

Informar Rota Via: SMS

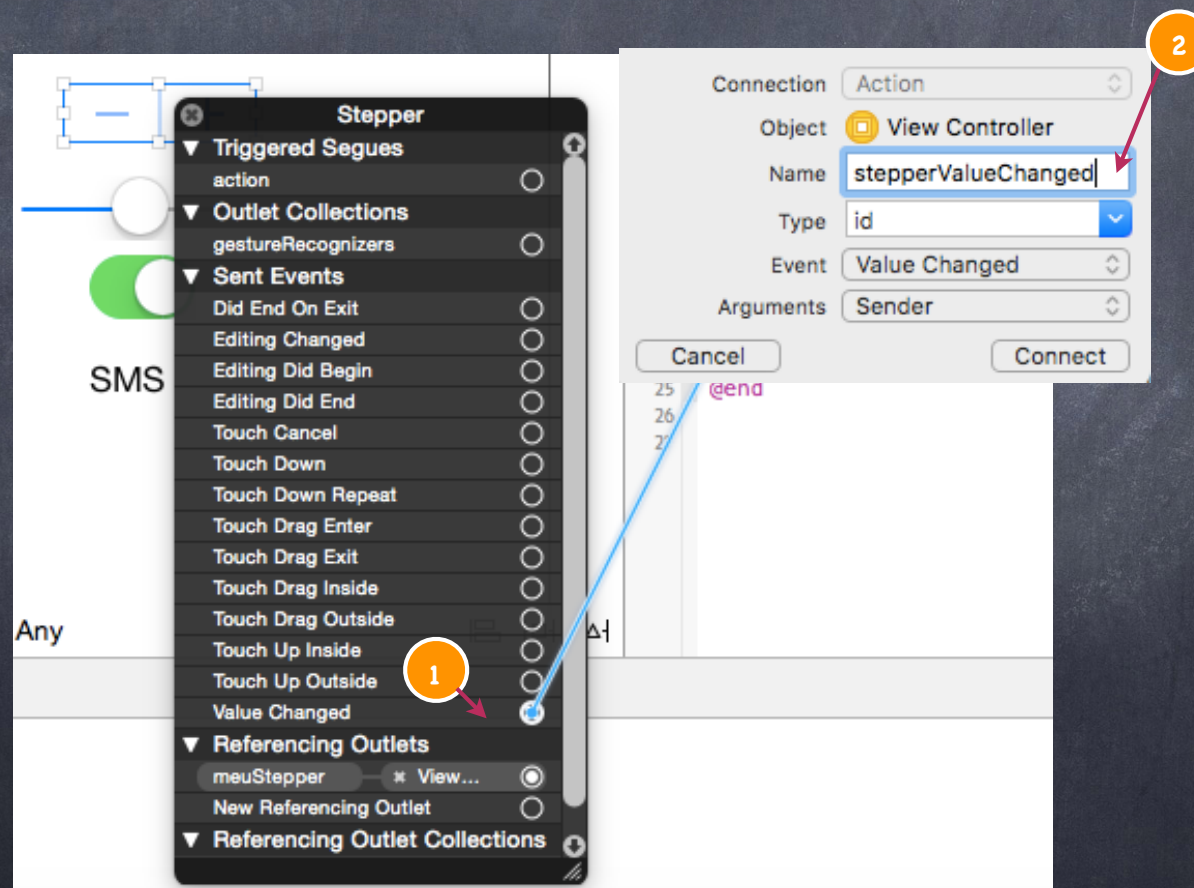
Salvar

2

```
2017-02-19 12:25:37.778 UIKIT OBJC[4021:132278] Você escolheu o primeiro botão
2017-02-19 12:25:41.559 UIKIT OBJC[4021:132278] Você escolheu o segundo botão
2017-02-19 12:25:42.806 UIKIT OBJC[4021:132278] Você escolheu SUV
```


IBAction no Arquivo .h do Stepper

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



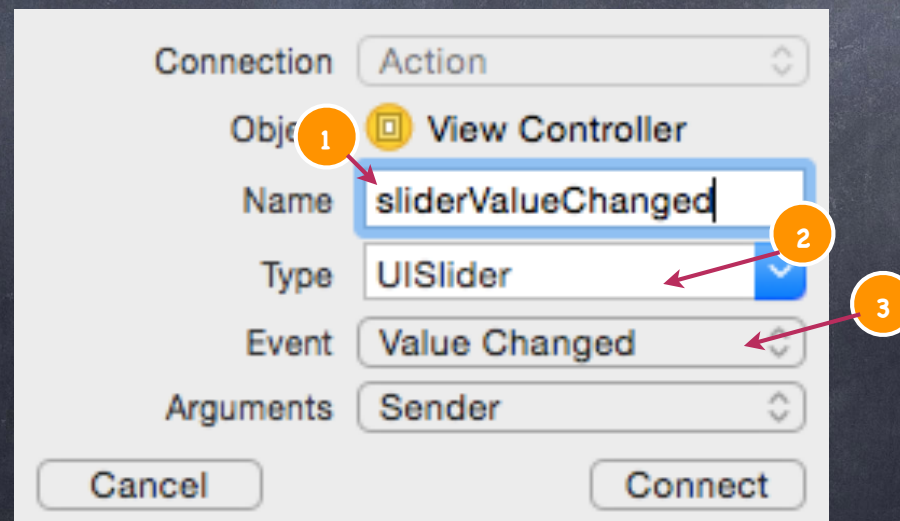
Implementação do Stepper no Arquivo .m

- A implementação do Stepper é muito simples, basta ir no arquivo.m e 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 método stringWithFormat da classe NSString para que informação seja convertida para String, inclusive formatamos para que tenha 1 casa decimal utilizando %0.0f.
- 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.

```
46 - (IBAction)stepperValueChanged:(id)sender {  
47     self.lblAno.text = [NSString stringWithFormat:@"%0.0f", self.meuStepper.value];  
48 }  
49
```


IBAction do Slider no .h

- 1 Clique com o botão direito sobre o Slider, escolha Value Changed leve até o arquivo.h, em Name (1) digite sliderValueChanged, troque o type (2) de ID 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 ID.



Implementação do Slider FIAP no Arquivo .m

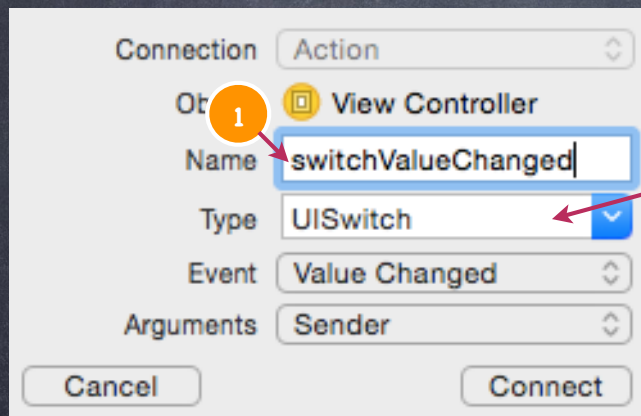
- A implementação do Slider é simples como o Stepper, basta ir no arquivo.m e passar o value do Slider para o lblKm, veja que como mudamos o type, ao invés de utilizarmos o nome de um Outlet para o slider usamos sender.value.

```
49  
50 - (IBAction)sliderValueChanged:(UISlider *)sender {  
51     self.lblKm.text = [NSString stringWithFormat:@"%0.0f", sender.value];  
52 }  
53
```


IBAction do Switch no .h

e implementação no .m

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

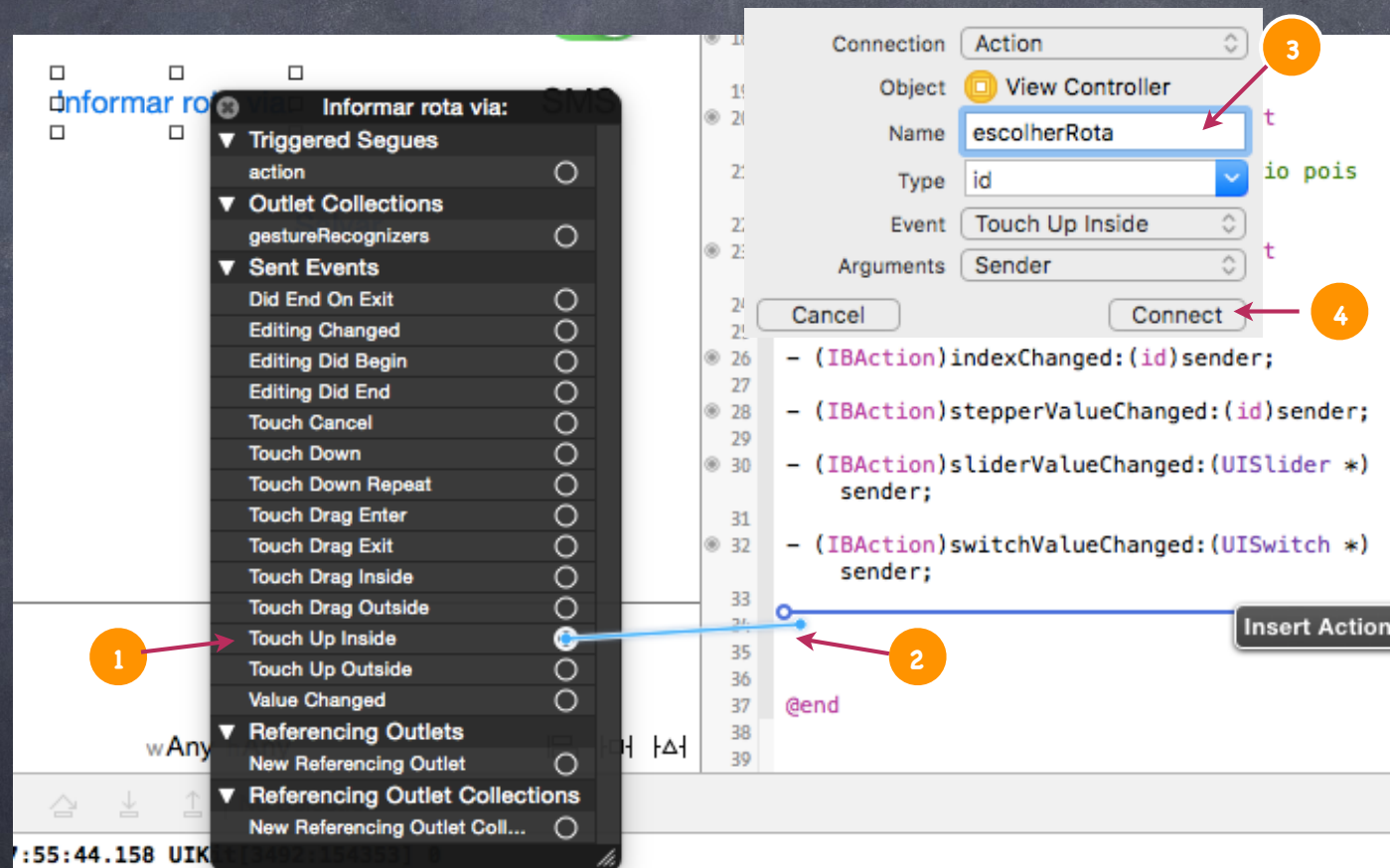


```
53  
54 - (IBAction)switchValueChanged:(UISwitch *)sender {  
55     NSLog(@"%d", sender.on );  
56 }
```

```
2017-02-19 12:42:46.286 UIKIT OBJC[4372:146124] 0  
2017-02-19 12:42:47.358 UIKIT OBJC[4372:146124] 1
```


IBAction do botão

"informar rota via" no .h



Botão no .m

```
56
57 - (IBAction)escolherRota:(id)sender {
58     UIAlertController *aviso = [UIAlertController
59         alertControllerWithTitle:@"Forma para receber a Rota"
60         message:@"Faça sua escolha"
61         preferredStyle:UIAlertControllerStyleActionSheet];
62     UIAlertAction *sms = [UIAlertAction
63         actionWithTitle:@"SMS"
64         style:UIAlertActionStyleDefault
65         handler:^(UIAlertAction * _Nonnull action) {
66             self.lblRota.text = action.title;
67         }];
68
69     UIAlertAction *email = [UIAlertAction
70         actionWithTitle:@"E-mail"
71         style:UIAlertActionStyleDefault
72         handler:^(UIAlertAction * _Nonnull action) {
73             self.lblRota.text = action.title;
74         }];
75     UIAlertAction *cancel = [UIAlertAction
76         actionWithTitle:@"Cancel"
77         style:UIAlertActionStyleCancel
78         handler:^(UIAlertAction * _Nonnull action) {
79             [aviso dismissViewControllerAnimated:YES completion:nil];
80         }];
81
82     [aviso addAction:sms];
83     [aviso addAction:email];
84     [aviso addAction:cancel];
85     [self presentViewController:aviso animated:YES completion:nil];
86 }
87
88
```


Execute - Command + R

FIAP

The screenshot shows an iPhone screen with a settings form. At the top, the status bar displays 'Carrier', signal strength, '5:59 PM', and battery level. The form has a light gray background. It includes a 'Veículo' section with buttons for 'Moto', 'Carro', and 'SUV'. Below is a text field for 'Nome do Proprietário'. 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 (green). The 'Informar Rota Via:' section has a dropdown menu currently showing 'SMS'. A 'Salvar' button is at the bottom of the form. A white modal dialog is open at the bottom, titled 'Forma para receber a Rota' with the subtitle 'Faça sua escolha'. It contains three buttons: 'SMS', 'E-mail', and 'Cancel'.

Carrier 5:59 PM

Veículo Moto Carro SUV

Nome do Proprietário

Ano Modelo 1980 - +

KM Mensal 7500

Rastrear Veículo ☒

Informar Rota Via: SMS

Salvar

Forma para receber a Rota
Faça sua escolha

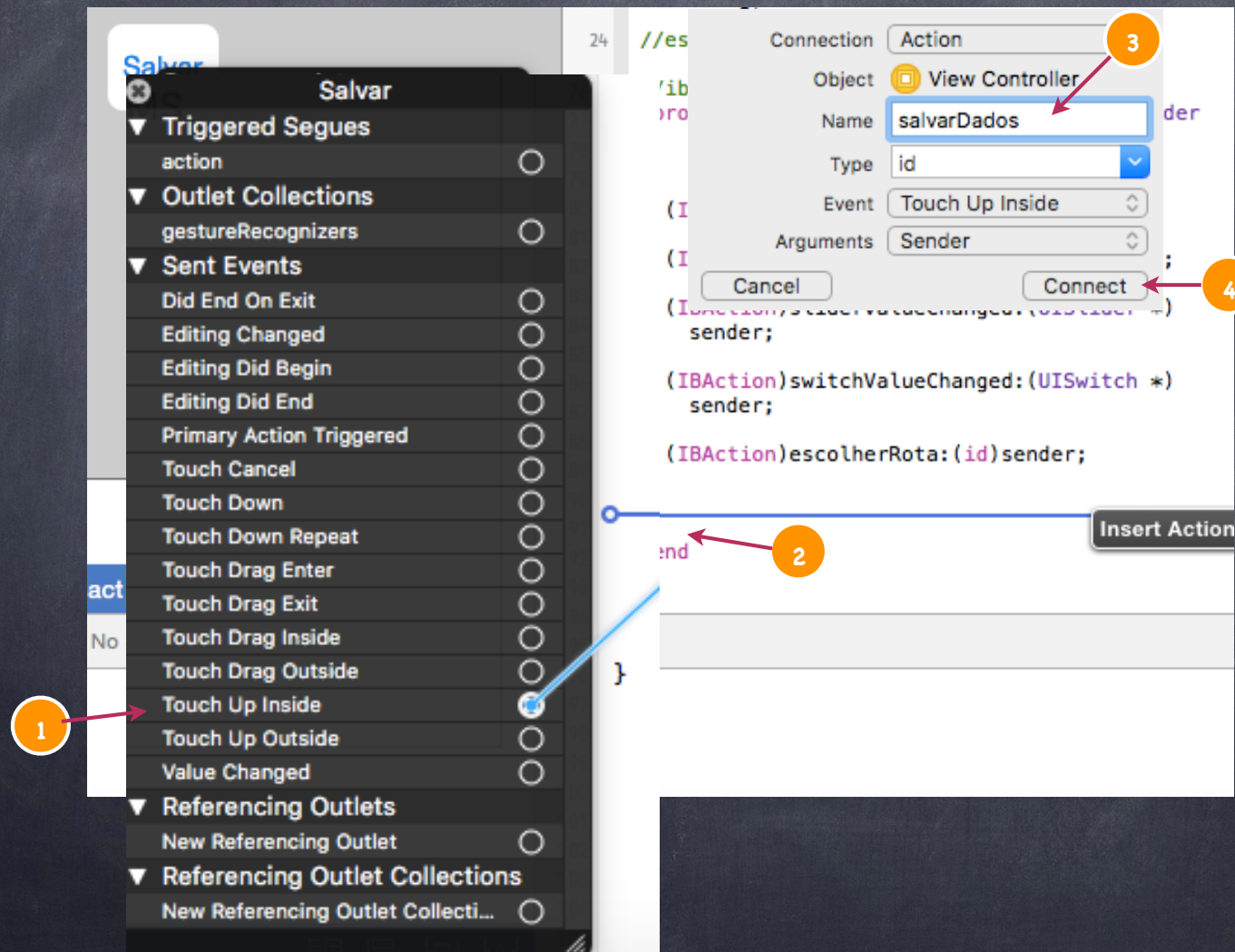
SMS

E-mail

Cancel

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

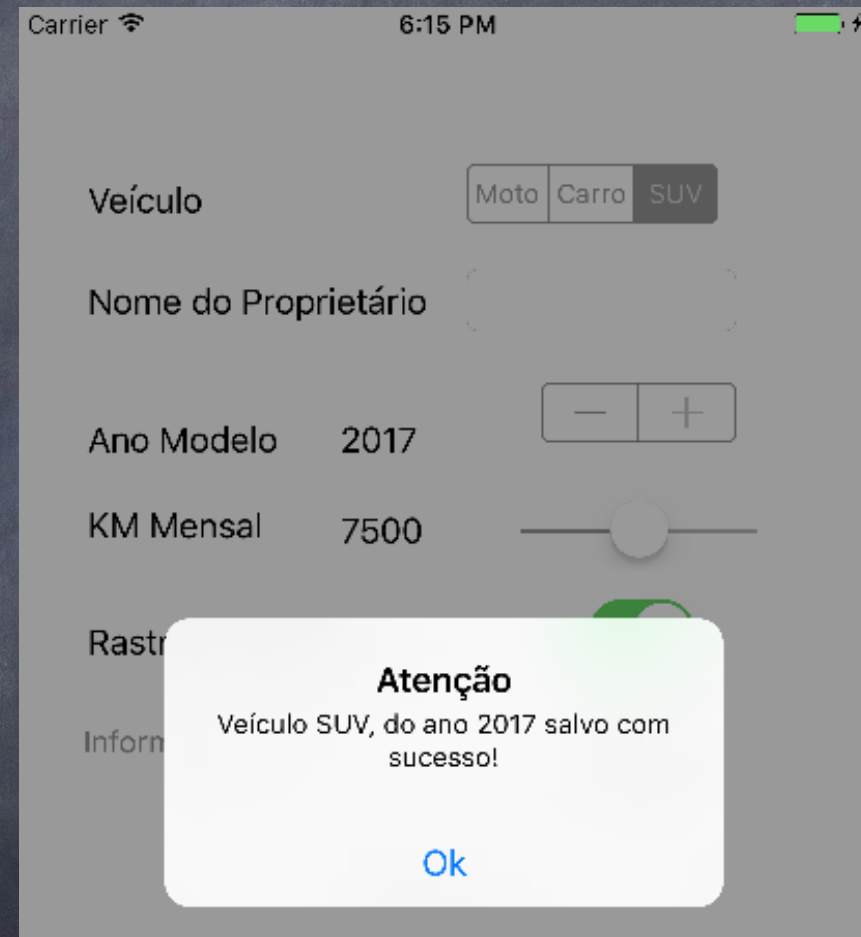
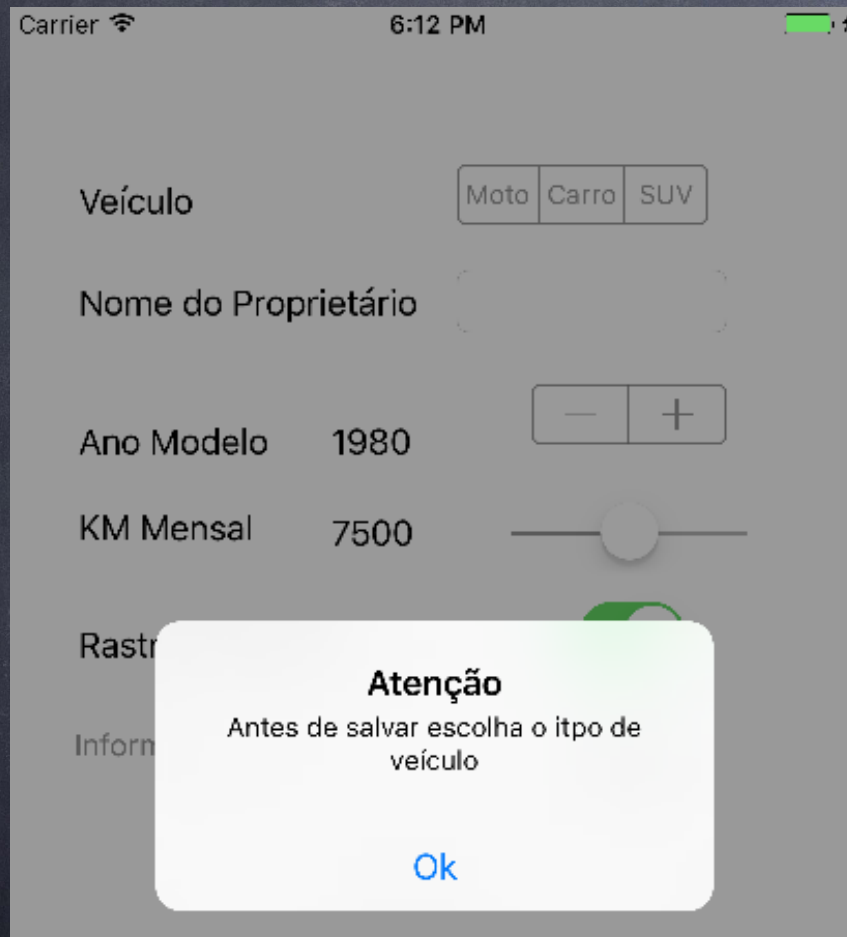
IBAction do botão "Salvar" no .h



Botão Salvar no .m

```
88
89 - (IBAction)salvarDados:(id)sender {
90     NSString *msg;
91     NSString *tipo;
92
93     if (self.meuSegmento.selectedSegmentIndex >=0){
94         tipo = [self.meuSegmento titleForSegmentAtIndex:self.meuSegmento.selectedSegmentIndex];
95         msg = [NSString stringWithFormat:@"Veículo %@, do ano %@ salvo com sucesso!", tipo, [self.lblAno text]];
96     }else{
97         tipo = @"";
98         msg = @"Antes de salvar escolha o itpo de veículo";
99     }
100
101     UIAlertController *alerta = [UIAlertController
102                                 alertControllerWithTitle:@"Atenção"
103                                 message:msg
104                                 preferredStyle:UIAlertControllerStyleAlert];
105
106     UIAlertAction *ok = [UIAlertAction
107                          initWithTitle:@"Ok"
108                          style:UIAlertActionStyleDefault
109                          handler:^(UIAlertAction * _Nonnull action) {
110                              [alerta dismissViewControllerAnimated:YES completion:nil];
111                          }];
112
113
114     [alerta addAction:ok];
115     [self presentViewController:alerta animated:YES completion:nil];
116 }
117
```


Execute - Command + R



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