





Enum



```
enum CalculatorOperator {  
    case plus  
    case minus  
}
```

```
enum CalculatorOperator {  
    case plus  
    case minus  
}  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        var operation: CalculatorOperator = .minus  
  
        operation = .plus  
  
        if operation == .plus {  
            print("É mais")  
        }  
    }  
}
```



```
enum ViewSelectedOption {  
    static let developer = 0  
    static let people = 1  
}
```

```
enum ViewSelectedOption {  
    static let developer = 0  
    static let people = 1  
}  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        let option = 1  
  
        if option == ViewSelectedOption.developer {  
            print("É Developer")  
        }  
    }  
}
```



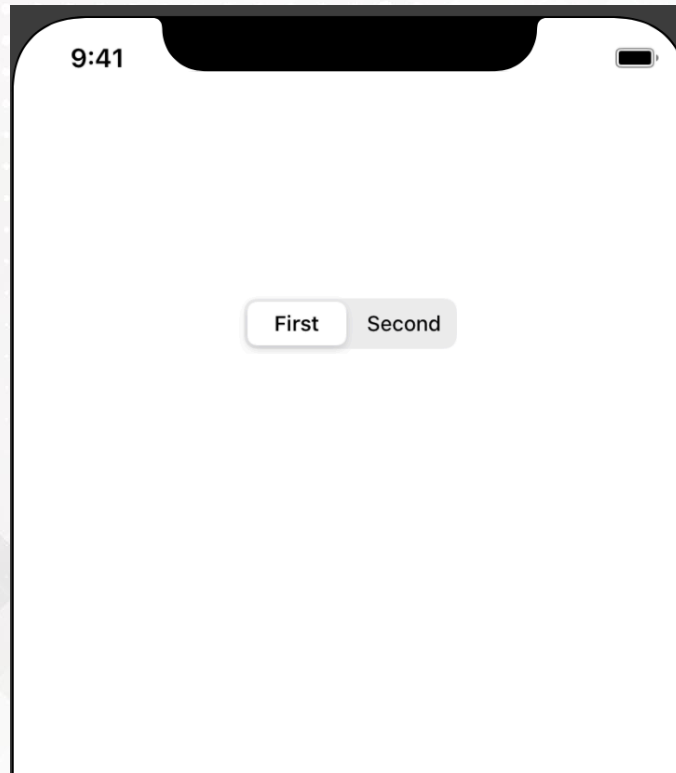
O que é
SegmentedControl ?



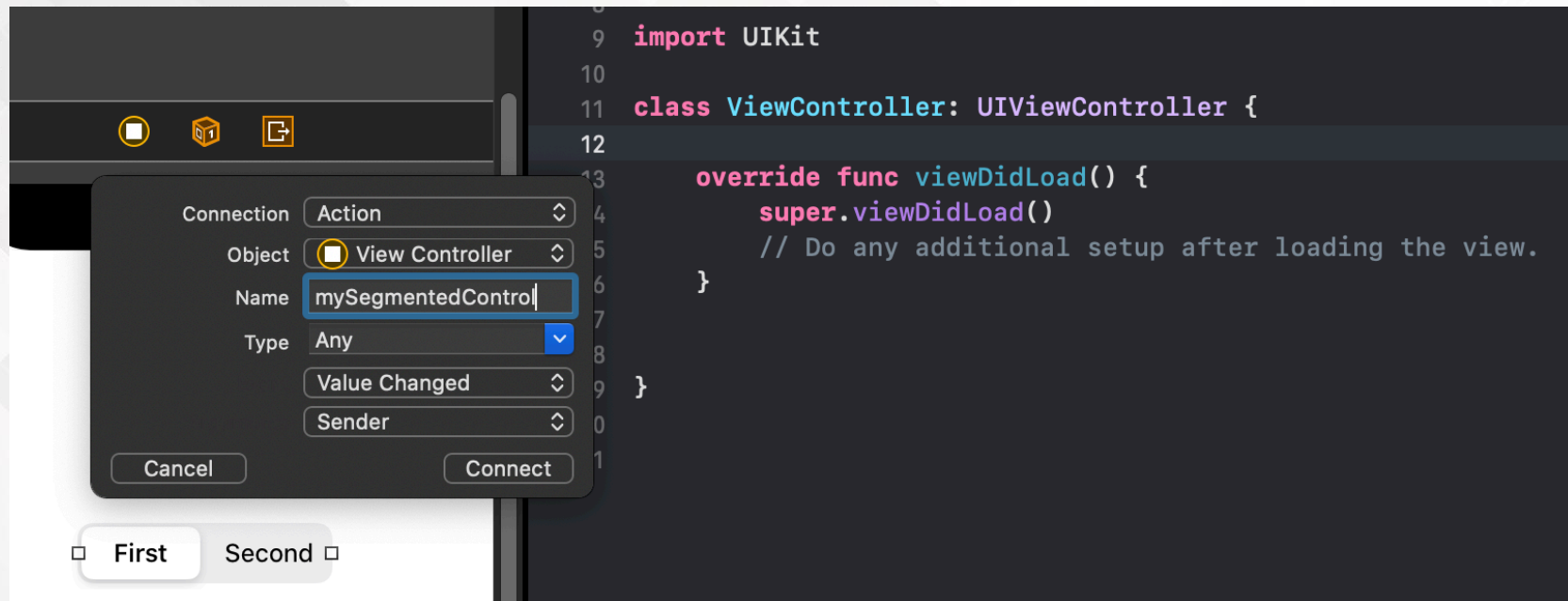
Um SegementedControl(controle segmentado) é um conjunto linear de dois ou mais segmentos, cada um dos quais funciona como um botão mutuamente exclusivo. Dentro do controle, todos os segmentos são iguais em largura. Como os botões, os segmentos podem conter texto ou imagens. Controles segmentados são frequentemente usados para exibir diferentes visualizações. No Google Maps, por exemplo, um controle segmentado permite alternar entre as visualizações Mapa, Trânsito e Satélite



Vamos Adicionar um SegmentedControl na ViewController



Para fazer a conexão do elemento, observe que vamos fazer com o tipo “Value Changed”, porque queremos saber quando o usuário modificou a escolha dele



The screenshot shows the Xcode interface. On the left, a connection dialog is open. It has the following fields:

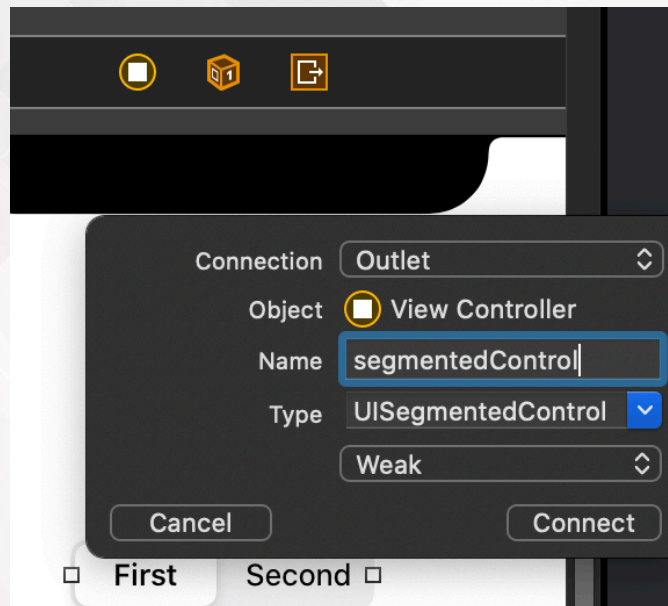
- Connection: Action
- Object: View Controller
- Name: mySegmentedControl
- Type: Any
- Value Changed
- Sender

At the bottom of the dialog are "Cancel" and "Connect" buttons. Below the dialog, there are two buttons: "First" and "Second".

On the right, the Swift code is visible:

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16     }
17
18 }
19
20 }
```

Feita a conexão



```
4 //
5 // Created by Nadilson Santana on 22/04/20.
6 // Copyright © 2020 Nadilson Santana. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16     }
17
18     @IBAction func mySegmentedControl(_ sender: Any) {
```

Agora vamos conectar o nosso IBOutlet para termos acesso as propriedades do SegmentedControl



Feita a conexão

12



```
@IBOutlet weak var segmentedControl: UISegmentedControl!
```

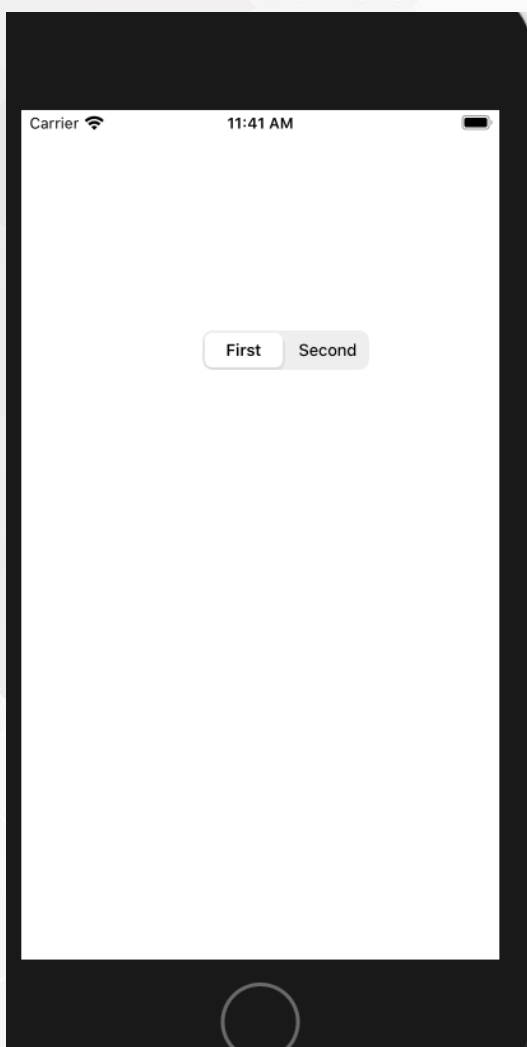
14

Vamos implementar o código para detectar quando o usuário trocou a opção no SegmentedControl

```
19  
20 ○ @IBAction func mySegmentedControl(_ sender: Any) {  
21     switch segmentedControl.selectedSegmentIndex {  
22     case 0:  
23         print("selecionou o primeiro item")  
24     case 1:  
25         print("selecionou o segundo item")  
26     default:  
27         break  
28     }  
29 }
```

Agora toda vez que o usuário trocar a opção, ira ser impresso no console

```
19  
20 ○ @IBAction func mySegmentedControl(_ sender: Any) {  
21     switch segmentedControl.selectedSegmentIndex {  
22     case 0:  
23         print("selecionou o primeiro item")  
24     case 1:  
25         print("selecionou o segundo item")  
26     default:  
27         break  
28     }  
29 }
```

```
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var segmentedControl: UISegmentedControl!
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17         // Do any additional setup after loading the view.
18     }
19
20     @IBAction func mySegmentedControl(_ sender: Any) {
21         switch segmentedControl.selectedSegmentIndex {
22             case 0:
23                 print("selecionou o primeiro item")
24             case 1:
25                 print("selecionou o segundo item")
26             default:
27                 break
28         }
29     }
30 }
31
32
```



Bora Coda

Links

SegmentedControl