



#digitalReskilling Software Developer

Desenvolvimento de Aplicações Mobile - iOS

Bruno Santos
bruno.santos@cesae.pt

UserDefaults

- *“An interface to the user’s defaults database, where you store key-value pairs persistently across launches of your app.”*
- A utilização do UserDefaults vai permitir às nossas aplicações a salvaguarda de dados entre utilizações. De notar que esta é uma funcionalidade que não substitui a utilização de uma base de dados, e que deve ser usada para elementos simples.

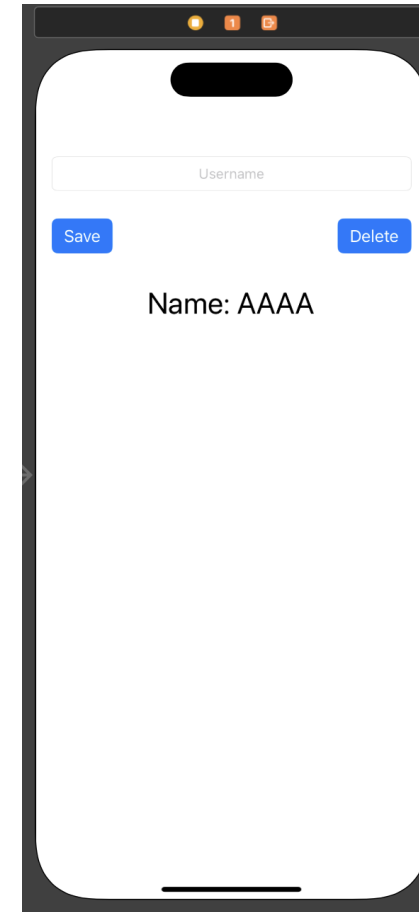
<https://developer.apple.com/documentation/foundation/userdefaults/1416603-standard>

UserDefaults

- Vamos criar uma aplicação com um campo editável para armazenar o username do utilizador.
- Teremos ainda dois botões: um para guardar o valor do username e outro para eliminar o mesmo valor.
- Finalmente vamos adicionar uma label para apresentar o valor a guardar/guardado

UserDefaults

- Text Field:
 - Placeholder: “Username”
- Button
 - Text: “Save”
- Button
 - Text: “Delete”
- Label
 - Text: “Name: AAAA”
 - Font: “System 30.0”



UserDefaults

```
7
8  import UIKit
9
10 class ViewController: UIViewController {
11
12     @IBOutlet weak var label_name: UILabel!
13     @IBOutlet weak var edit_name: UITextField!
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17         // Do any additional setup after loading the view.
18     }
19
20     @IBAction func button_delete(_ sender: Any) {
21     }
22
23     @IBAction func button_save(_ sender: Any) {
24     }
25 }
26
27
```

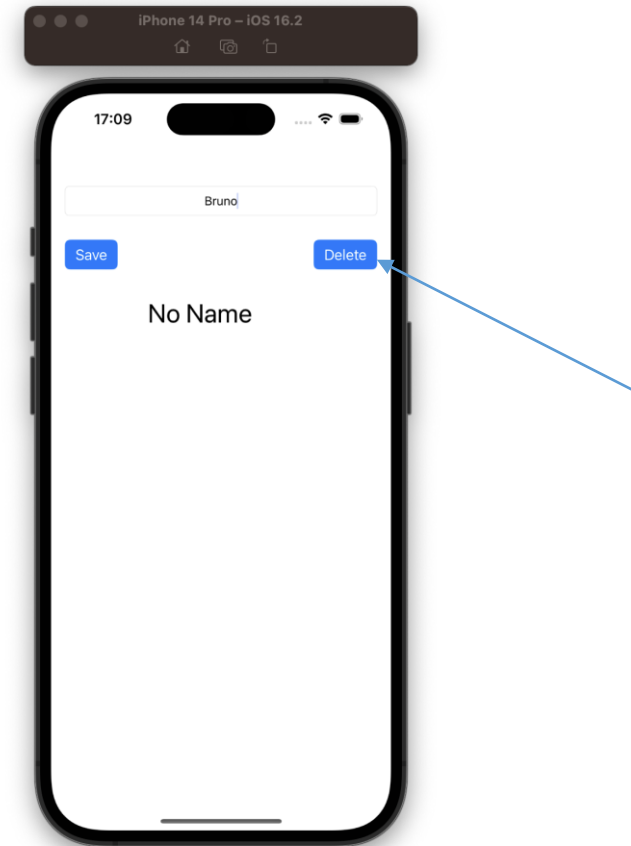
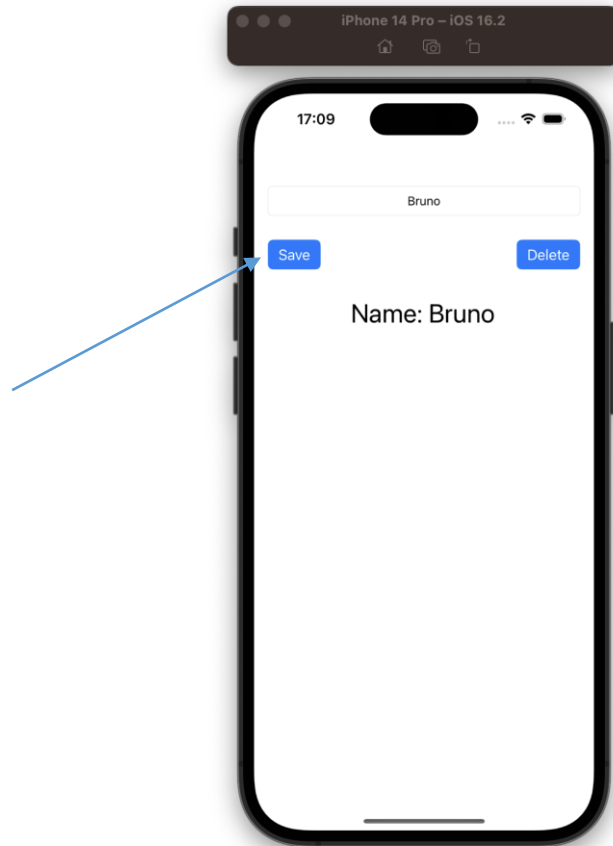
UserDefaults

- Começando pela lógica de preenchimento da label aquando do clique no botão de gravar e de limpar a mesma no botão de eliminar

UserDefaults

```
8  import UIKit
9
10 class ViewController: UIViewController {
11
12
13     @IBOutlet weak var label_name: UILabel!
14     @IBOutlet weak var edit_name: UITextField!
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18         // Do any additional setup after loading the view.
19     }
20
21     @IBAction func button_delete(_ sender: Any) {
22         label_name.text = "No Name"
23     }
24
25     @IBAction func button_save(_ sender: Any) {
26         let storedName = edit_name.text
27         label_name.text = "Name: \(storedName!)"
28     }
29 }
30
```

UserDefaults



UserDefaults

- No entanto é necessário utilizar o UserDefaults para armazenar em memória, entre utilizações da aplicação, o valor armazenado, assim, aquando do clique em cada um dos botões terá de ser armazenado ou eliminado o valor, e no método viewDidLoad carregados os valores em memória.

UserDefaults

```
override func viewDidLoad() {
    super.viewDidLoad()

    let storedName = UserDefaults.standard.object(forKey: "name")
    if let newName = storedName as? String {
        label_name.text = newName
    }
}

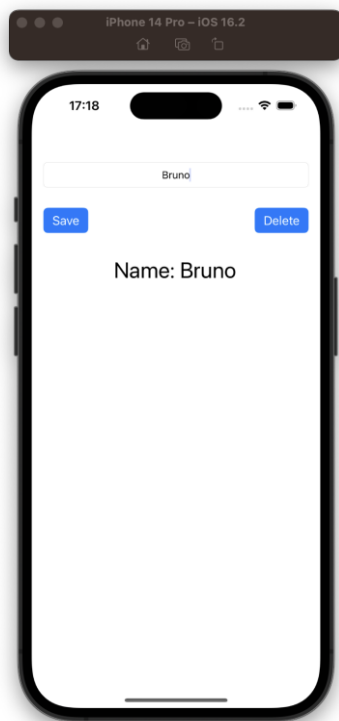
@IBAction func button_delete(_ sender: Any) {
    let storedName = UserDefaults.standard.object(forKey: "name")
    if(storedName as? String) != nil{
        UserDefaults.standard.removeObject(forKey: "name")
        label_name.text = "No Name"
    }
}

@IBAction func button_save(_ sender: Any) {
    label_name.text = "Name: \(edit_name.text!)"

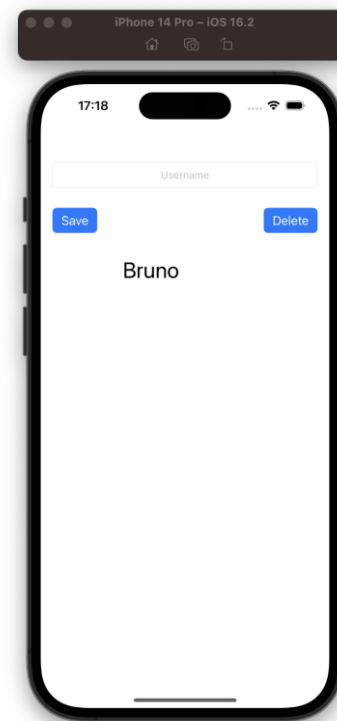
    UserDefaults.standard.set(edit_name.text!, forKey: "name")
}
```

UserDefaults

1ª execução – clique em Save



2ª execução



UserDefaults

1ª execução – clique em Delete



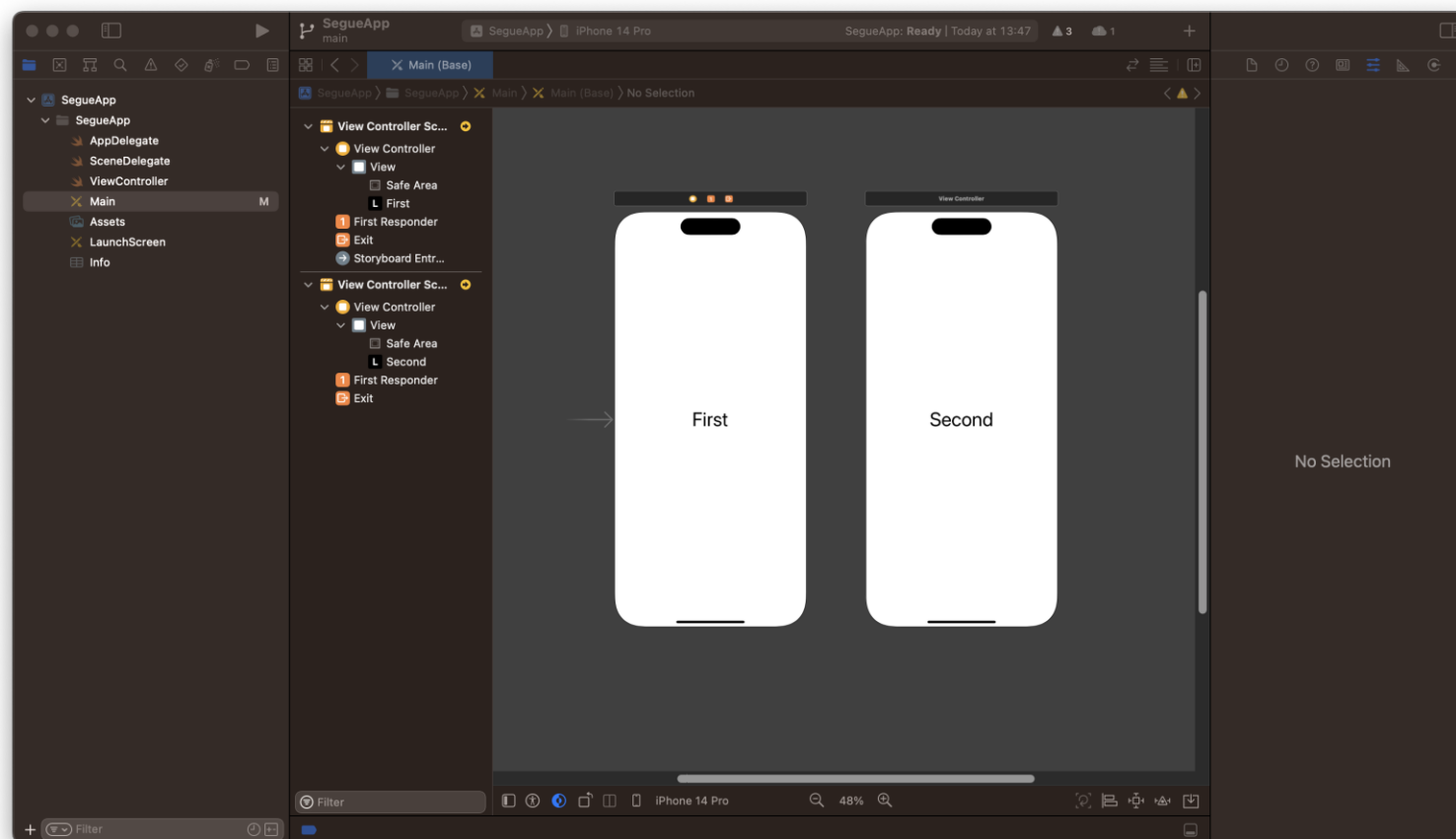
2ª execução



Second Screen

- Vamos criar uma nova aplicação que permita a navegação entre ecrãs.
- Abrindo o Main vamos adicionar um novo ViewController arrastando o mesmo para o ecrã.
- De forma a diferenciar vamos colocar uma label em cada um com a identificação daquela view.

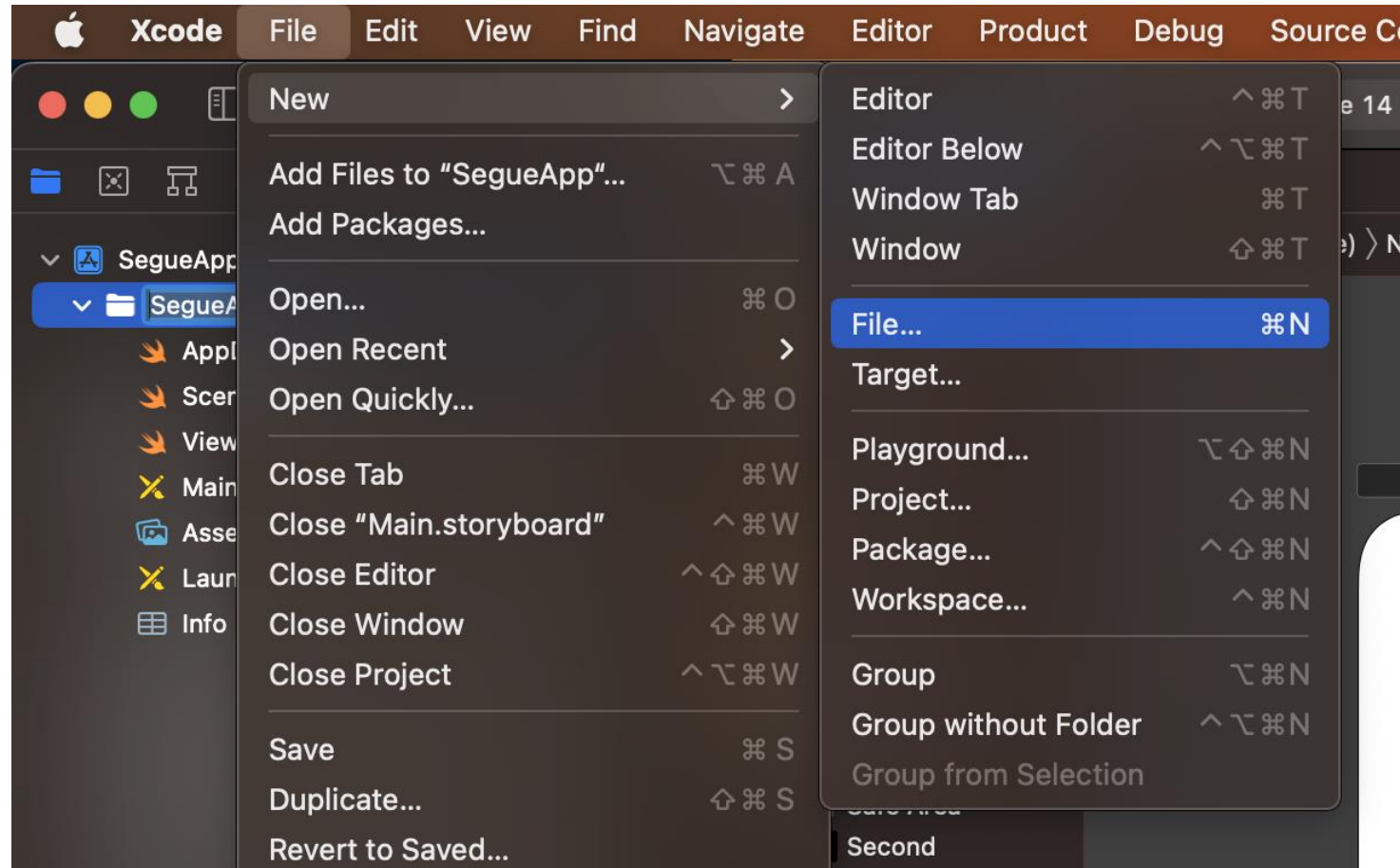
Second Screen



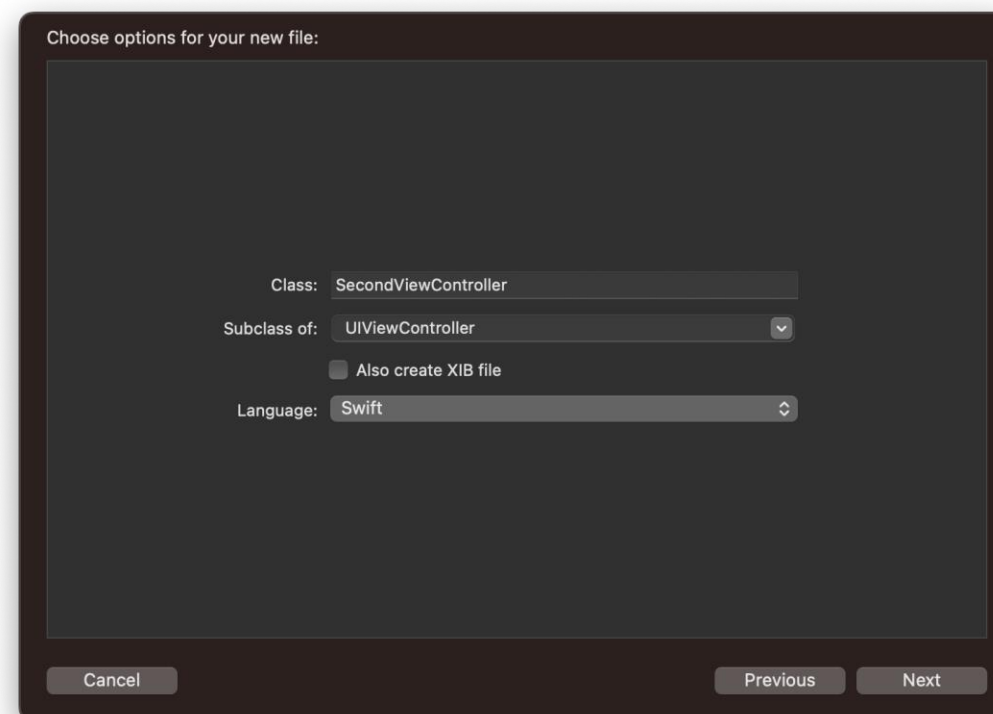
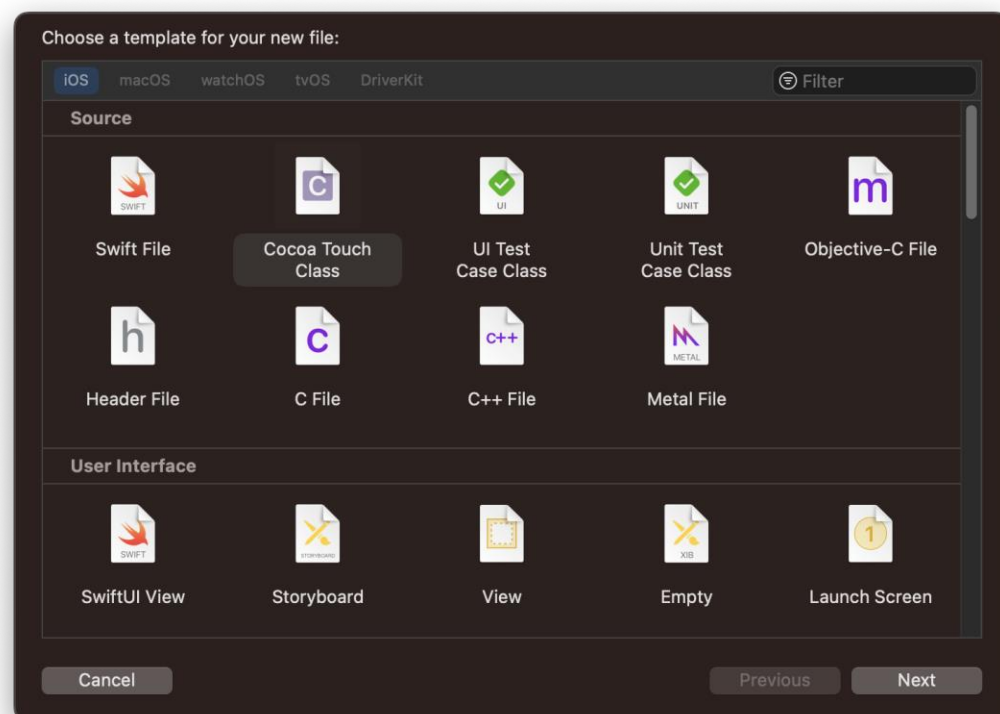
Second Screen

- Como passo seguinte necessitamos criar um ficheiro Swift para referenciar e manipular os elementos de layout da segunda View.

Second Screen



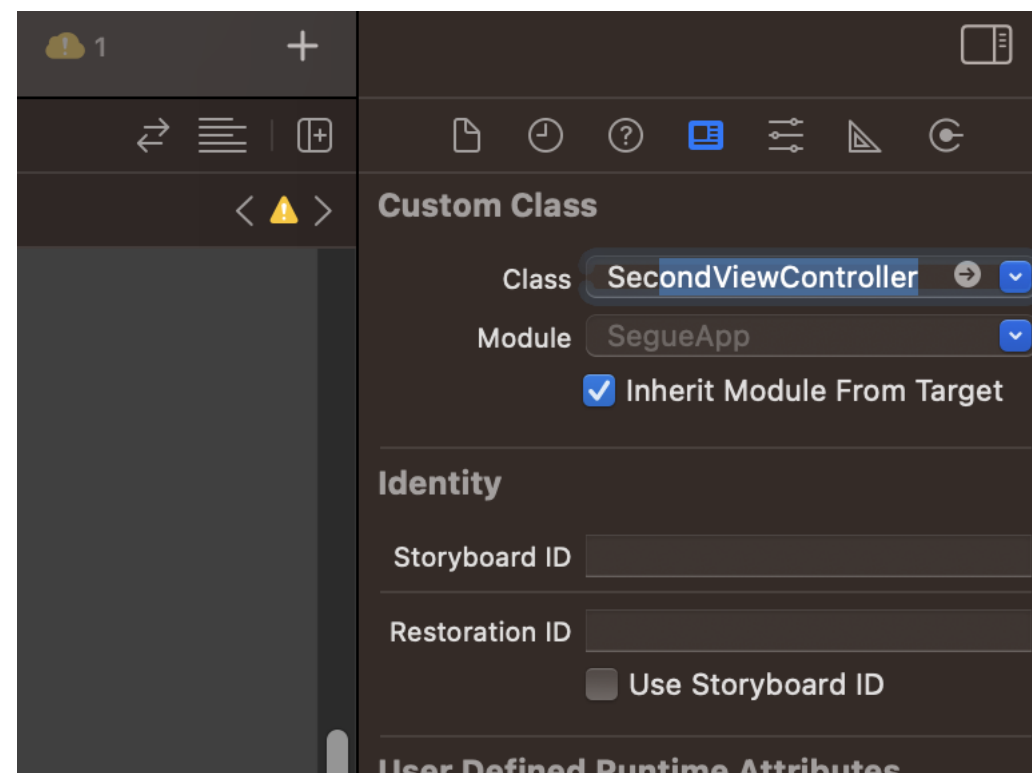
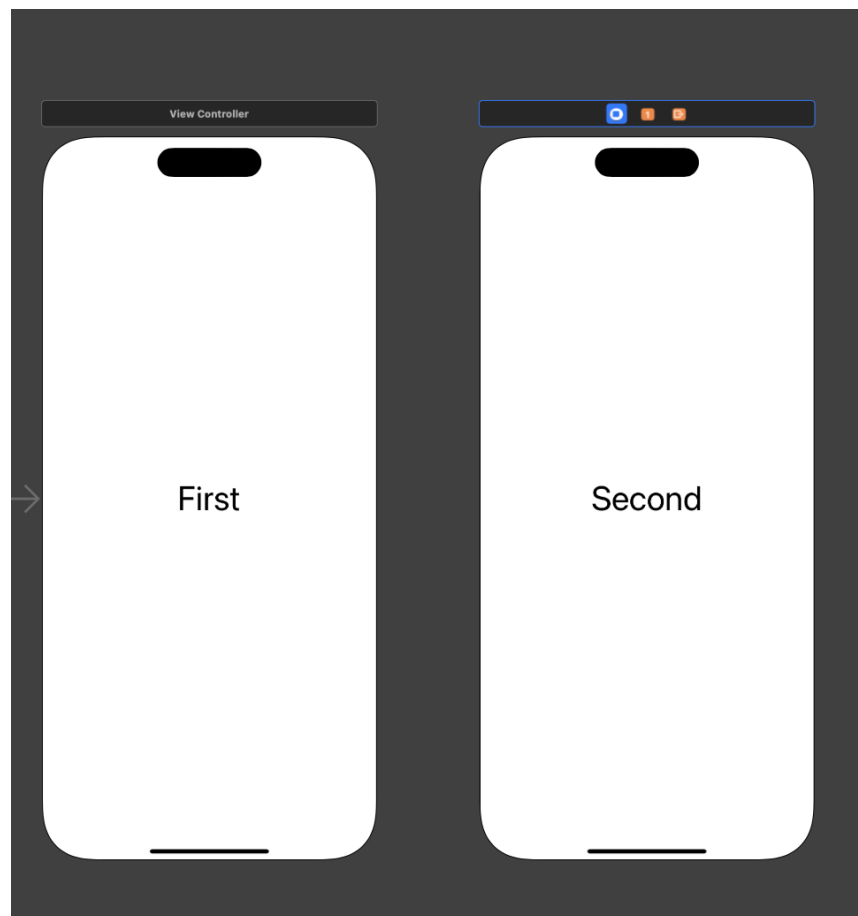
Second Screen



Second Screen

- Voltando ao Controller, clicamos no ícone do ViewController e selecionamos o SecondViewController

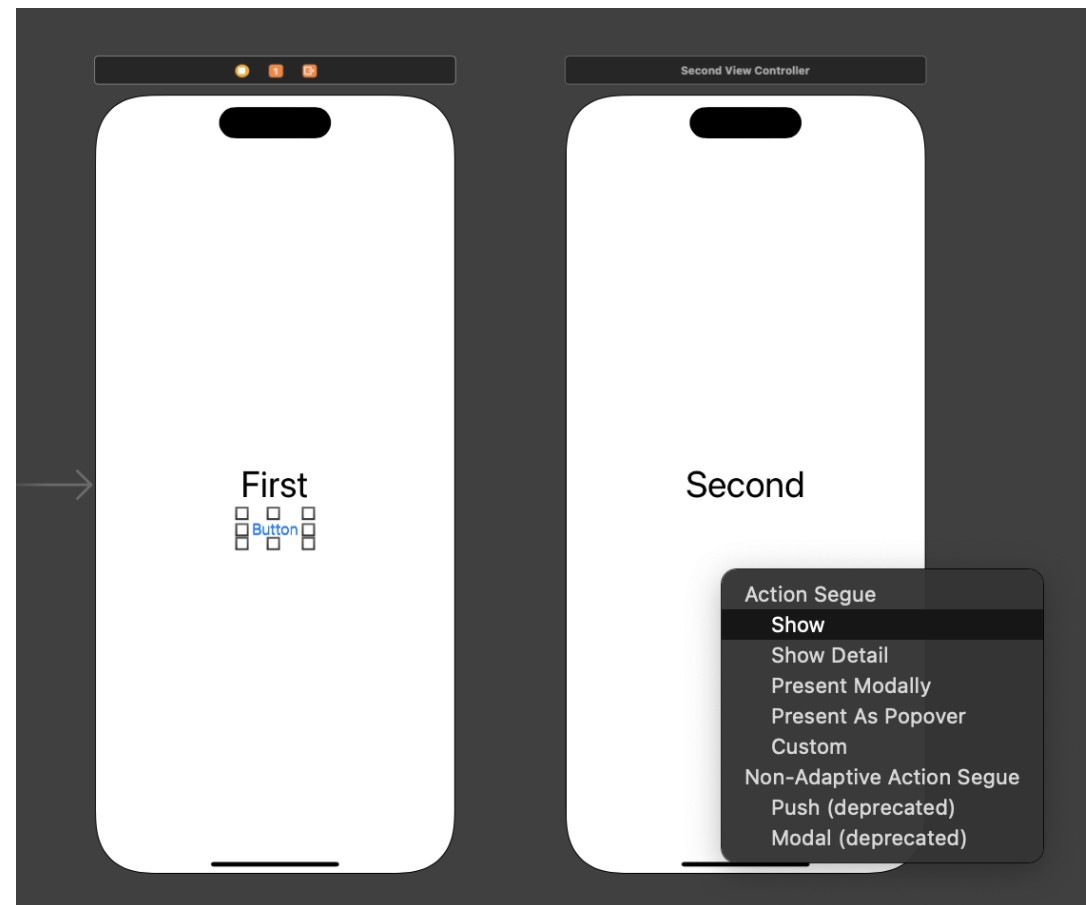
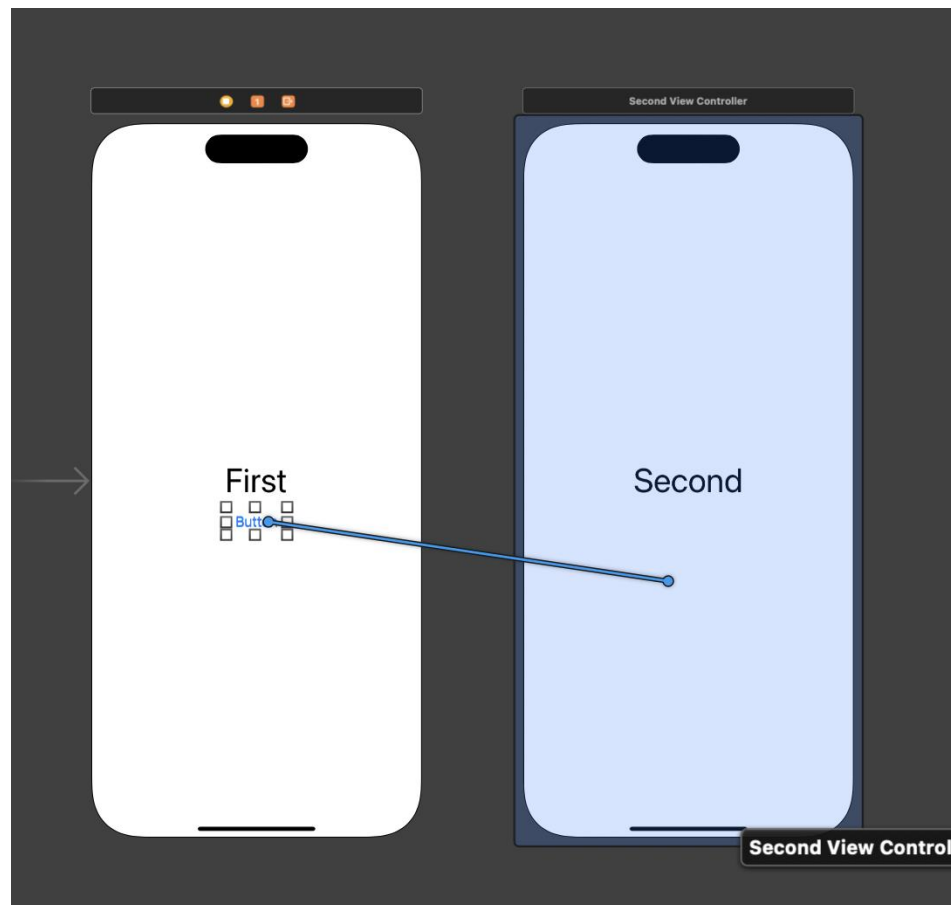
Second Screen



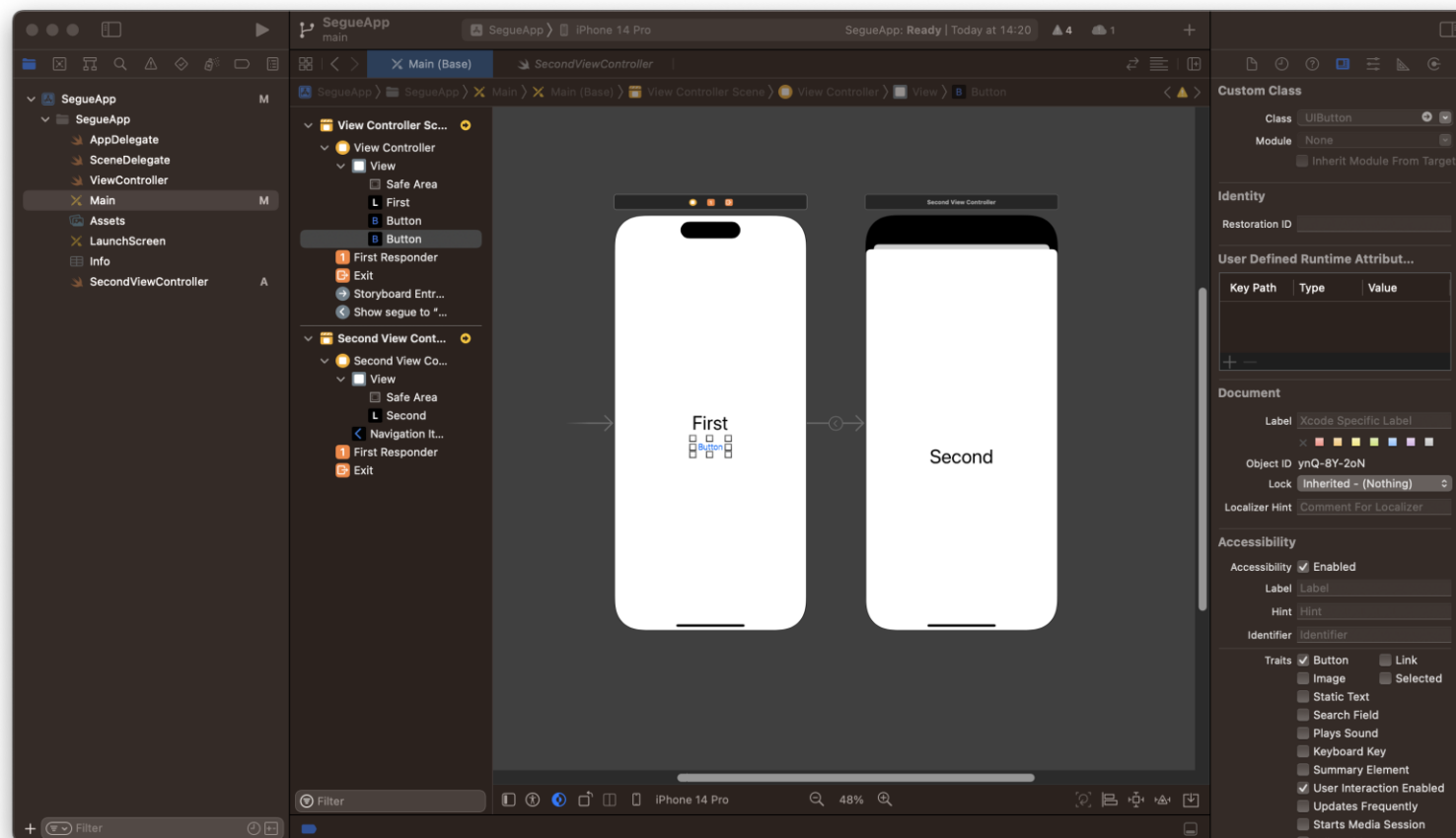
Segue

- O Segue permite-nos navegar entre os dois ecrãs criados anteriormente, neste caso vamos adicionar um botão e criar uma ação para navegar para o segundo ecrã

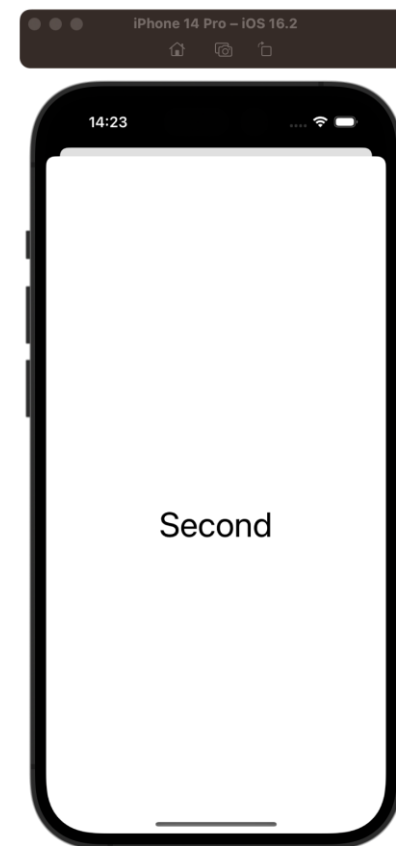
Segue



Segue



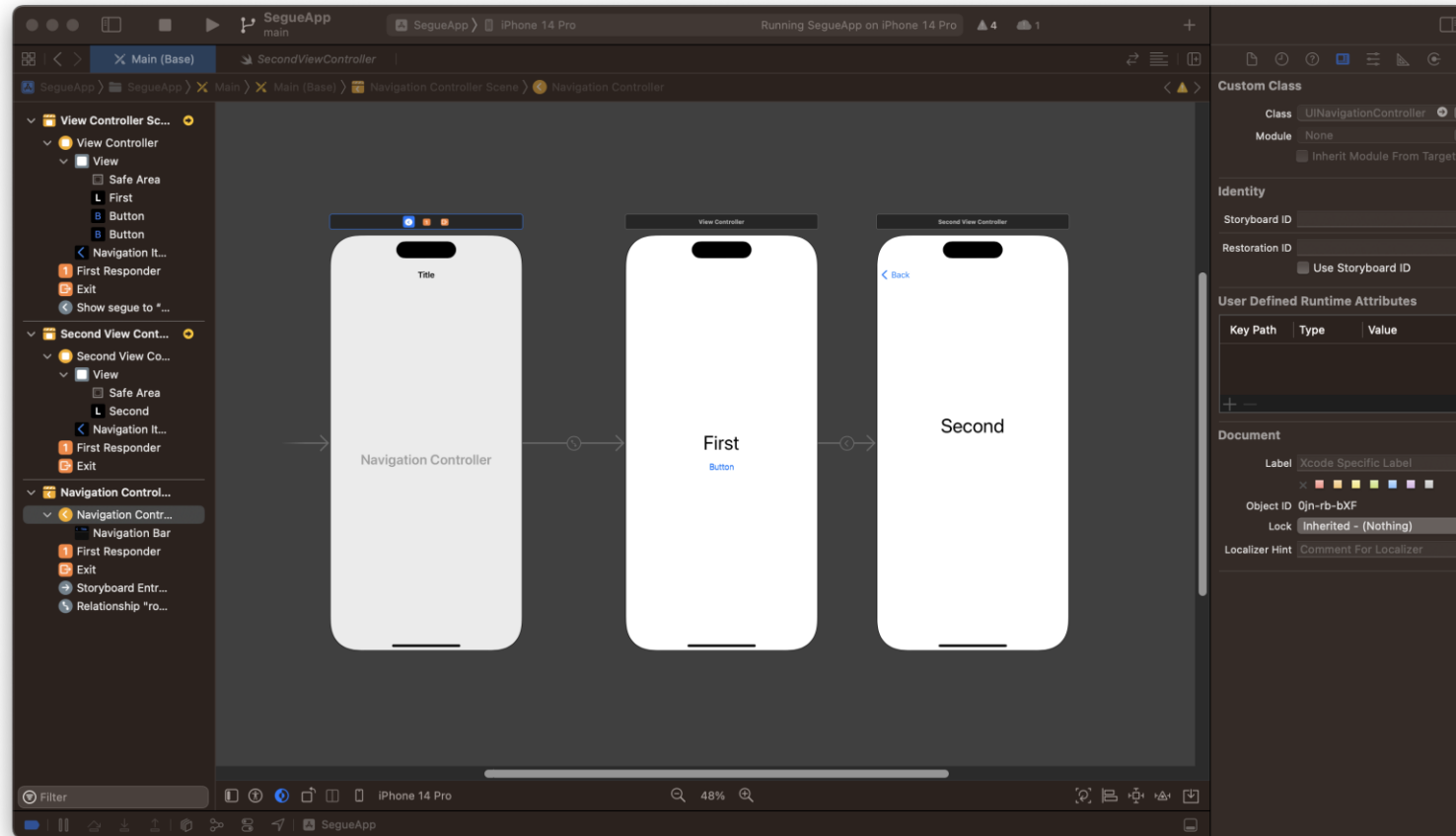
Segue



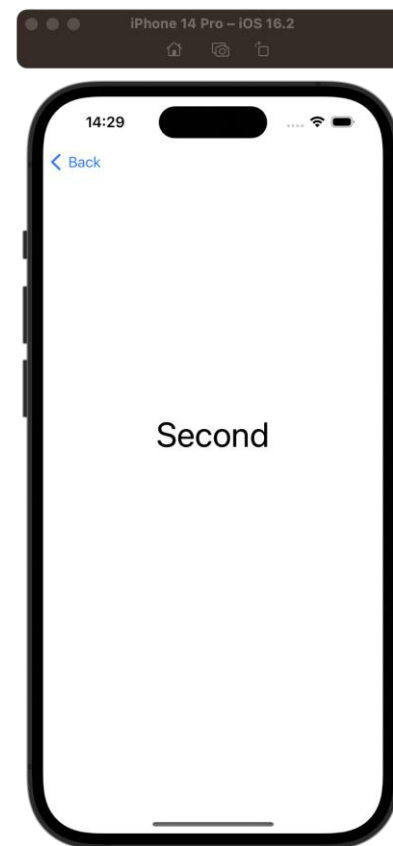
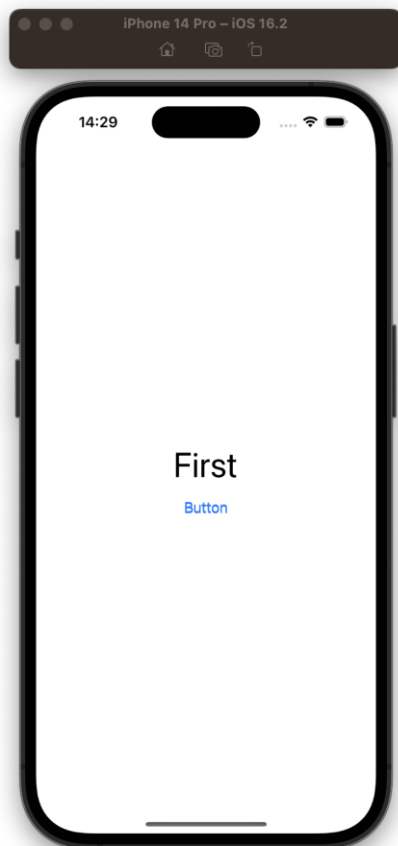
Navigation Controller

- Esta é uma solução que nos permite a navegação entre ecrãs, no entanto não é a mais comum, assim vamos seleccionar o ecrã principal e seguida no menu Editor → Embed In → Navigation Controller

Navigation Controller



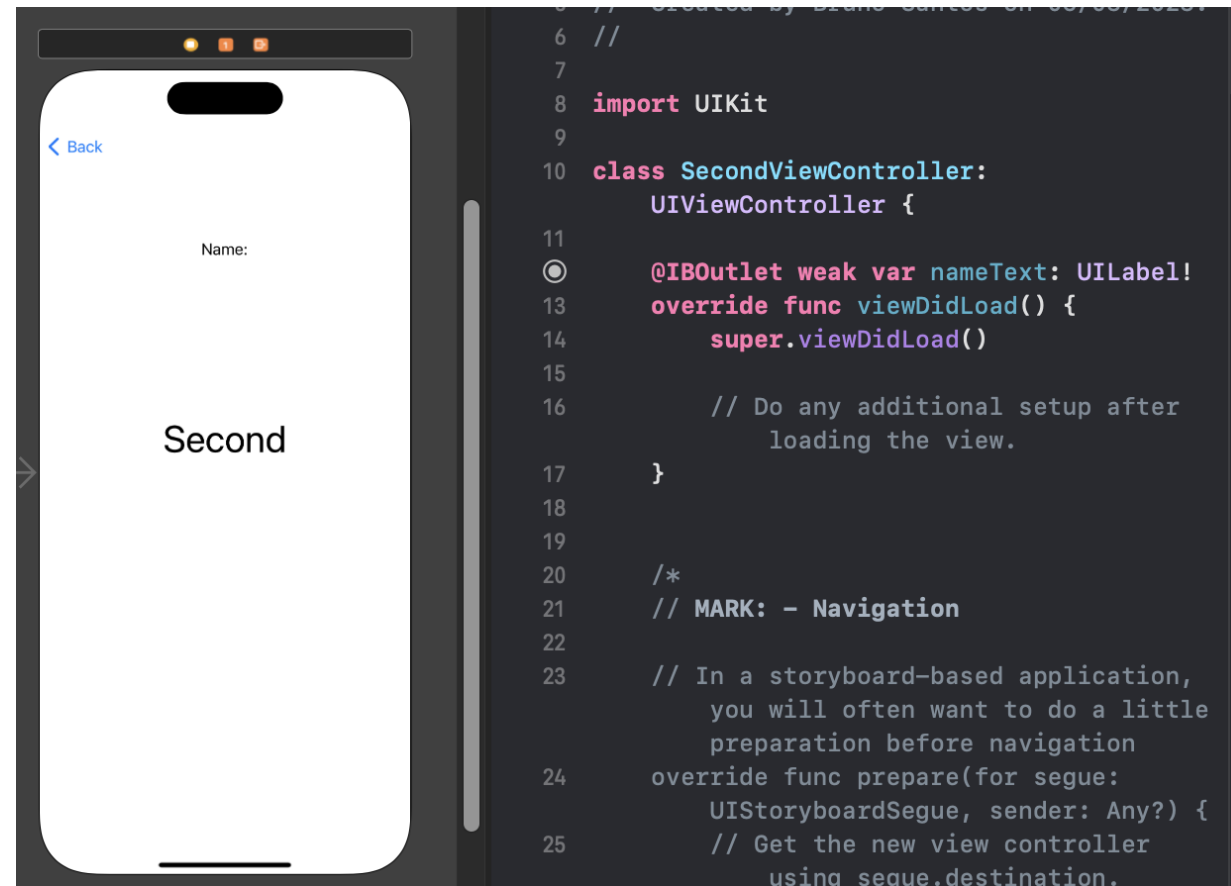
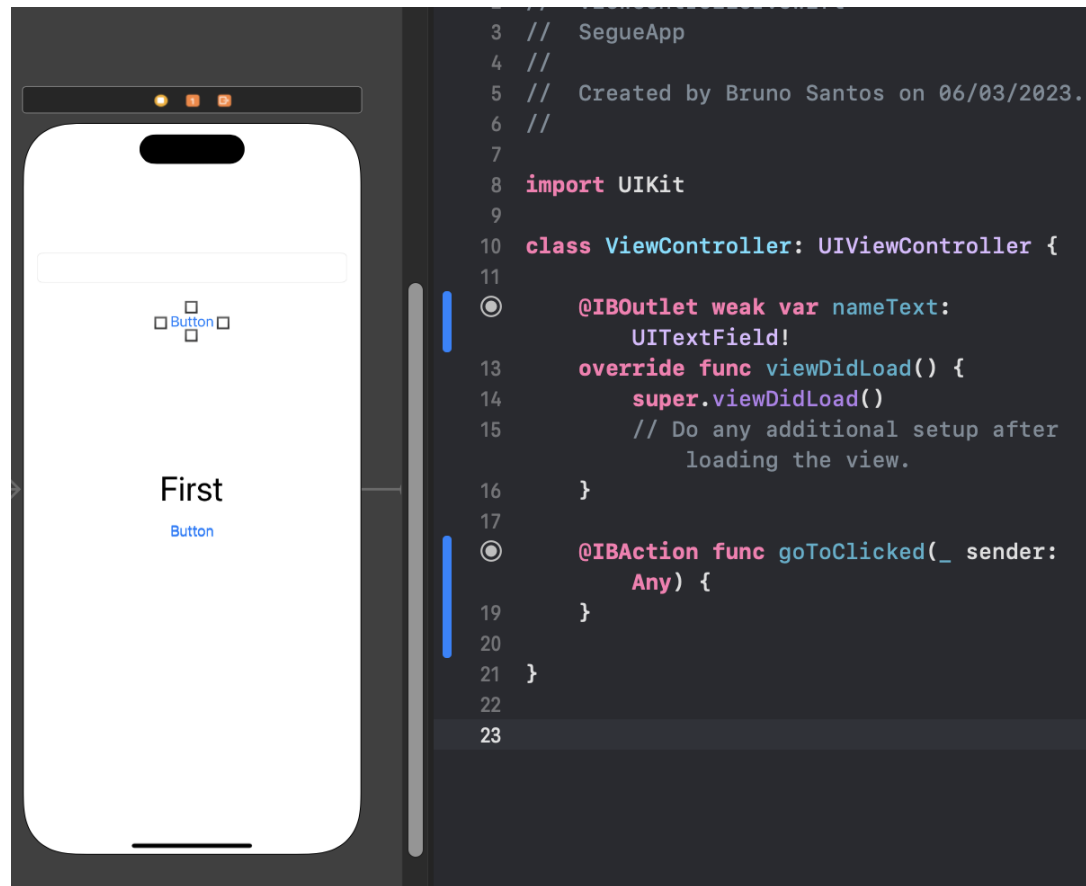
Navigation Controller



Segue

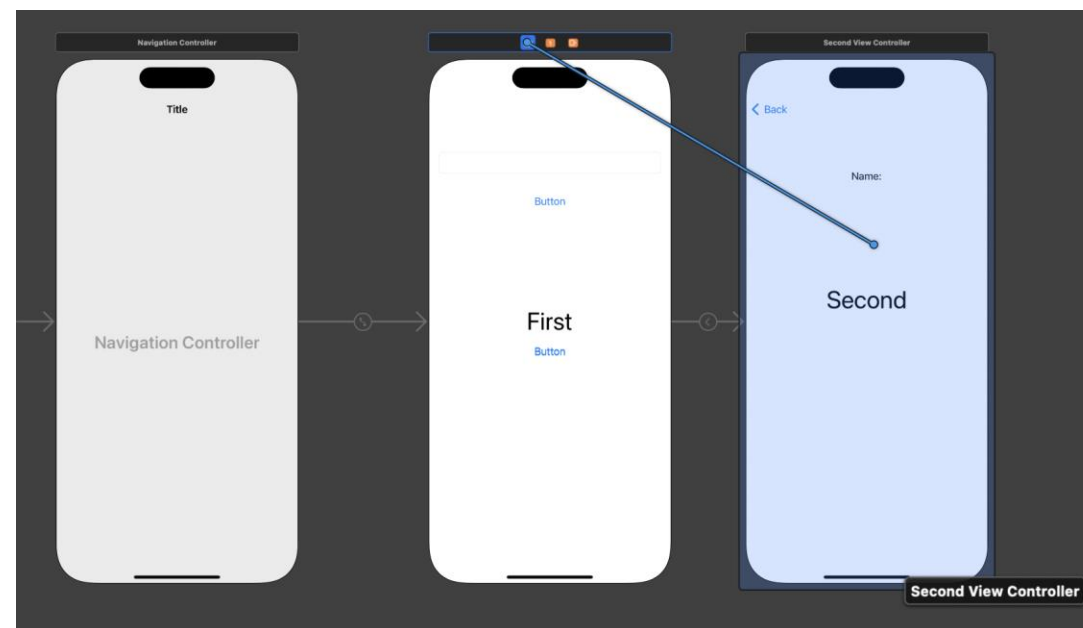
- O objetivo agora será alterar o ViewController aplicando um campo editável para, ao passar para o segundo ViewController transportar o valor inserido pelo utilizador

Segue



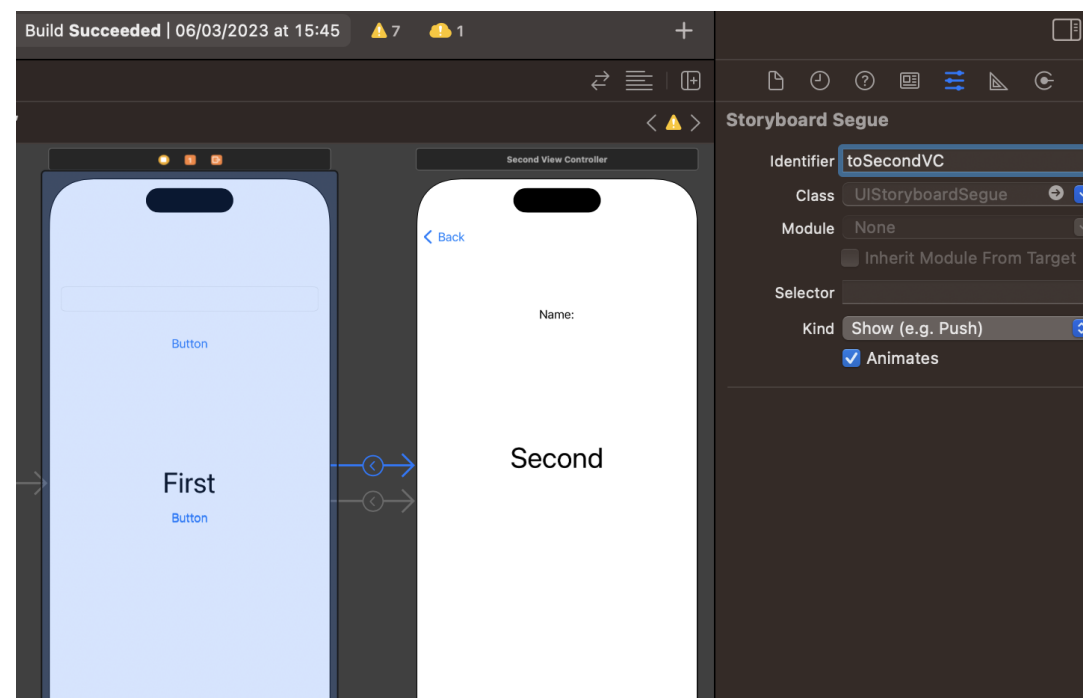
Segue

- Clicando no ViewController arrastar para o SecondViewController para criar o Segue.



Segue

- Alteramos agora o identificador, neste caso vamos colocar toSecondVC



Segue

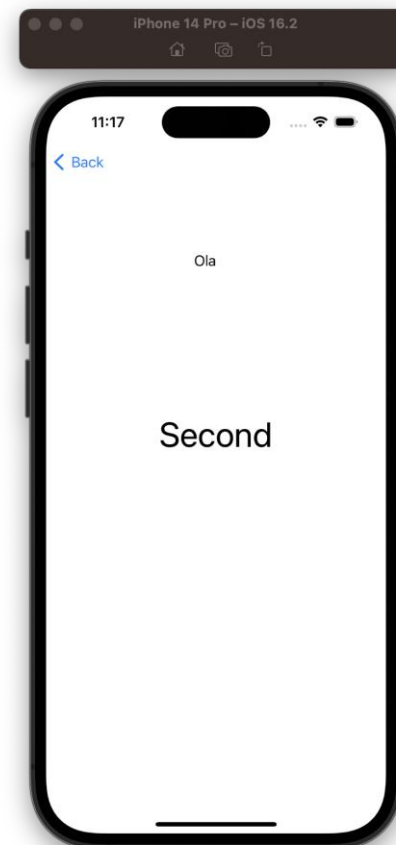
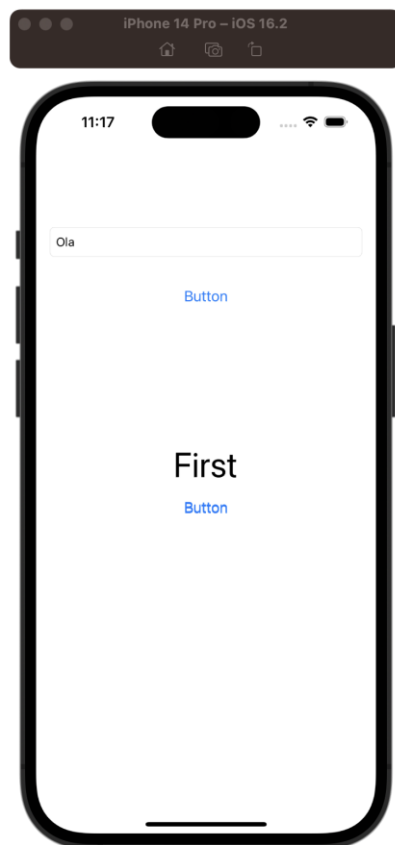
- No evento de clique, criado anteriormente, adicionamos o código para realizar o Segue.

Segue

```
8 import UIKit
9
10 class ViewController: UIViewController {
11     var userName = ""
12
13     @IBOutlet weak var nameText: UITextField!
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         // Do any additional setup after loading the view.
17     }
18
19     @IBAction func goToClicked(_ sender: Any) {
20         userName = nameText.text!
21         performSegue(withIdentifier: "toSecondVC", sender: nil)
22     }
23
24     override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
25         if segue.identifier == "toSecondVC" {
26             let destinationVC = segue.destination as! SecondViewController
27             destinationVC.myName = userName
28         }
29     }
30 }
```

```
8 import UIKit
9
10 class SecondViewController: UIViewController {
11
12     var myName = ""
13
14     @IBOutlet weak var nameText: UILabel!
15     override func viewDidLoad() {
16         super.viewDidLoad()
17
18         nameText.text = myName
19
20         // Do any additional setup after loading the view.
21     }
22 }
```

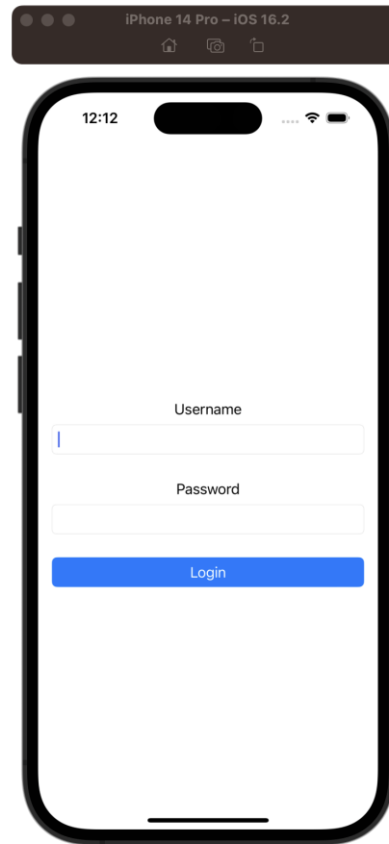

Segue



Exercício 1

- Crie uma nova aplicação e implemente as seguintes operações:
 1. Permita ao utilizador ter campos para inserir o username e password e quando clicar no botão apareça “Login” valide a autenticação.
 2. Caso o utilizador seja user e password pass considere login correto, caso contrário login incorreto

Exercício 1



Exercício 2

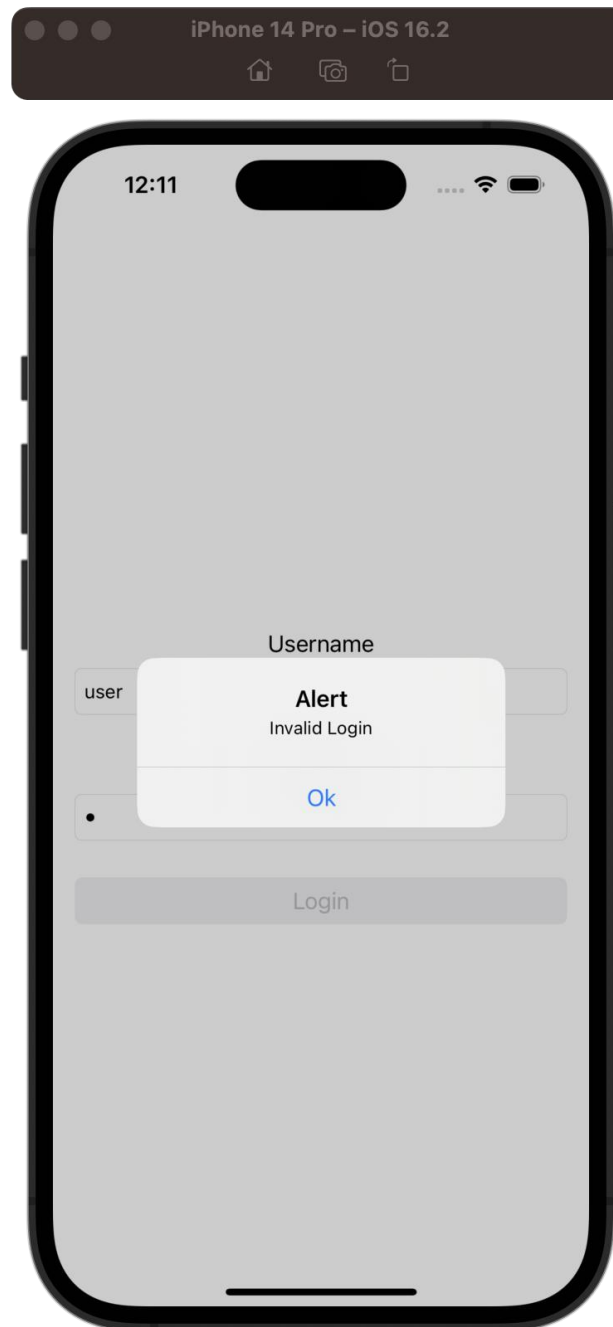
- Crie uma nova aplicação semelhante à anterior mas não coloque a Label que irá apresentar Olá e o nome do utilizador. Alternativamente utilize um UIAlertController colocando o seguinte código:

```
let alert = UIAlertController(title: "Alert", message: "Invalid Login", preferredStyle: UIAlertController.Style.alert)

alert.addAction(UIAlertAction(title: "Ok", style: UIAlertAction.Style.default, handler: nil))

self.present(alert, animated: true, completion: nil)
```

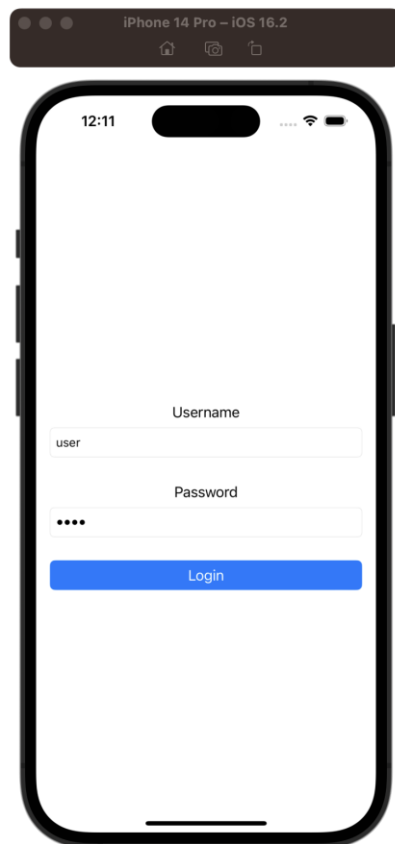
Exercício 2



Exercício 3

- Altere o exercício anterior:
 - Aplique o Navigation Controller
 - Quando voltar ao primeiro ViewController apague os dados inseridos no formulário

Exercício 3



Exercício 4

- Programe um conversor de moeda de forma a que preenchido o valor em Euros, quando clicado num dos três botões apareça uma mensagem com respetivo valor em reais, dólares ou pesos.
- O layout fica ao seu critério.
- Considere como taxa de conversão:
- 1 euro (€) = 4,4 reais (R\$)
- 1 euro (€) = 1,20 dólares (\$)
- 1 euro (€) = 31,5 pesos (\$)

Exercício 5

- Crie uma aplicação que pede ao utilizador para inserir um número e apresente:
 - A indicação se o número é par ou ímpar (utilize uma Label).
 - A indicação se o número é ou não primo (utilize uma Label).
- NOTA: Um número é primo se for divisível apenas por 1 e por ele mesmo. Exemplos: 2, 3, 5, 7,...

Exercício 6

- Pretende-se determinar o número de semanas, dias e horas a que corresponde um dado valor de horas pedido ao utilizador. Exemplo: 300h são 1 semana, 5 dias e 12 horas.
- Crie os elementos de layout necessários para apresentar os valores de semanas, dias e horas em elementos separados.