



PROF. RENÊ XAVIER

**DESENVOLVIMENTO PARA IOS 10 COM
SWIFT 3**

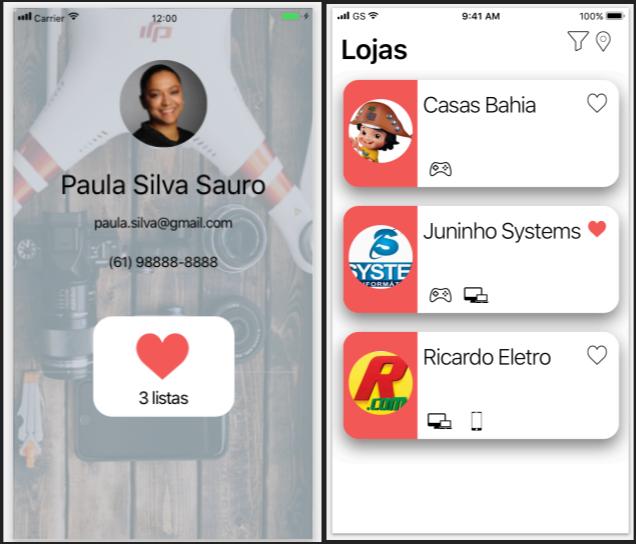
O QUE VAMOS FAZER HOJE?

AGENDA

- ▶ Navegação
 - ▶ Segues
 - ▶ Outlets
 - ▶ Tipos
 - ▶ Tab Bar Controller
 - ▶ Split Controller
 - ▶ Navigation Controller



CRIAR TELAS ✓ COMO LIGAR ELAS?





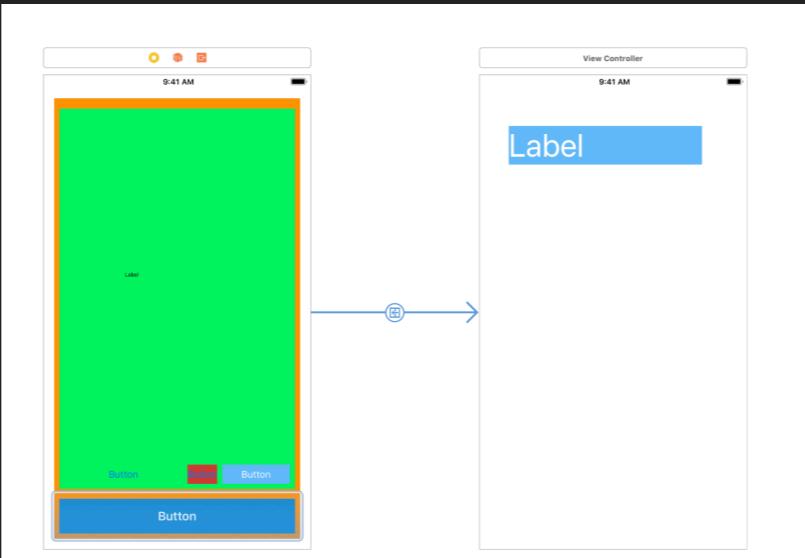
SEGUES

SEGUE - DEFINIÇÃO

- ▶ Conexão entre as ViewControllers;
- ▶ Uma transição de uma ViewController para outra ViewController que não pode ser interrompida;
- ▶ Cria uma nova instância da próxima View Controller;

SEGUE - DEFINIÇÃO

► Visual;



SEGUE - DEFINIÇÃO

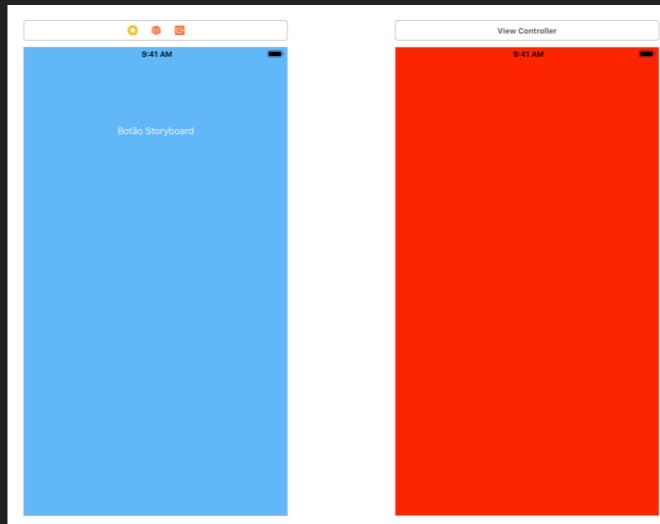
► Programático;

```
16
17    @IBAction func acaoBotao(_ sender: Any) {
18        performSegue(withIdentifier: "SegueIdentifierProximaTela", sender: sender)
19    }
```



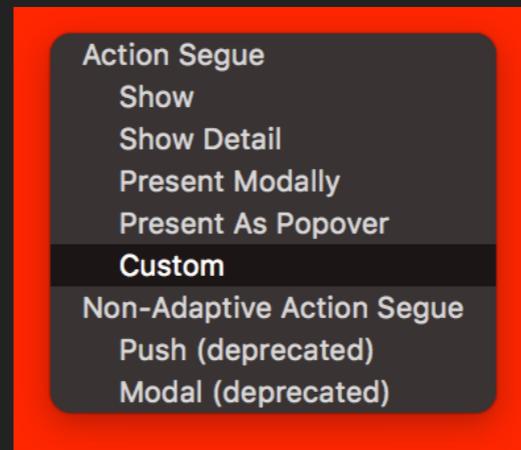
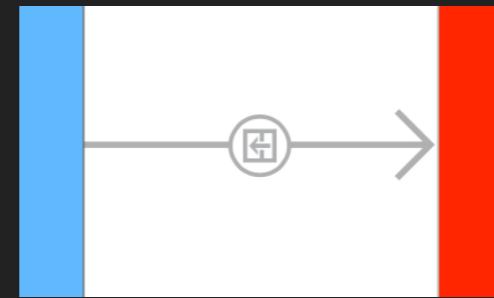
PRÁTICA

- ▶ Criar duas View Controllers novas
- ▶ Colocar backgrounds diferentes
- ▶ Adicionar um botão



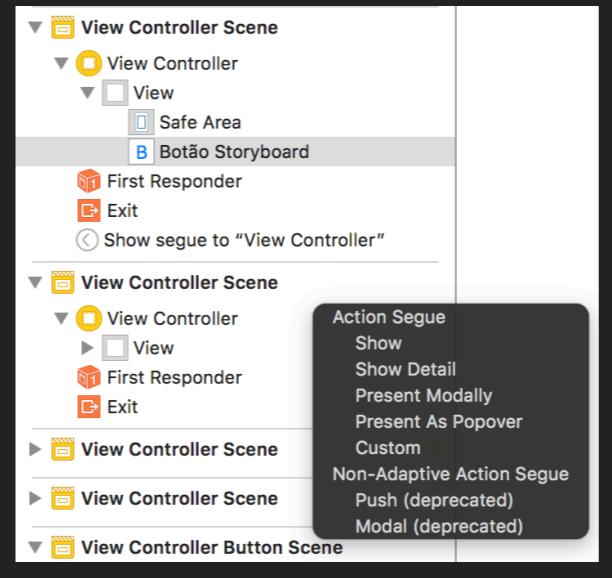
Para conectar as duas telas, precisamos do bom e velho Ctr + Clica + Arrasta

- ▶ Ctr + Clica + Arrasta - Clique no botão da primeira View Controller e arraste até a segunda View Controller
- ▶ A tela ao lado irá exibir. Selecione Show
- ▶ A segue foi criada



Também é possível usar o Ctr + Clica + Arrasta do botão até a hierarquia de views localizada na esquerda.

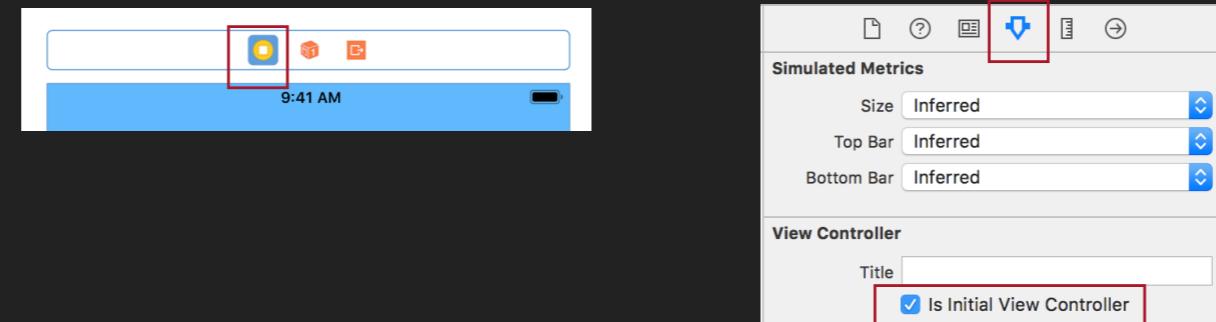
O nome das View Controllers está igual



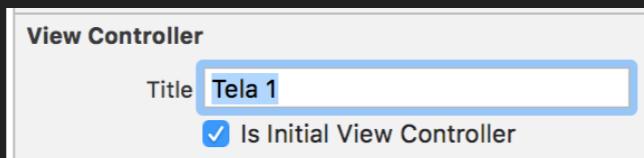
Nome igual View Controllers
Pode ser um pouco confuso, mas iremos resolver isso daqui a pouco.
Por enquanto continue selecionando a VC no Storyboard.

Antes de testar, temos que definir a View Controller inicial

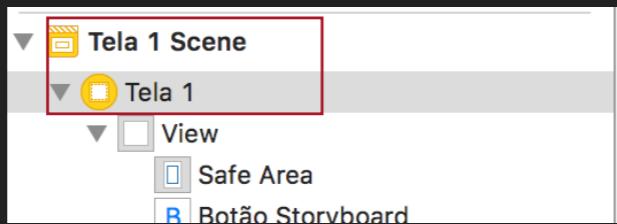
- ▶ Selecione a View Controller
- ▶ Colocar backgrounds diferentes



Também adicione um título



Algo mudou com o título?



TESTAR NO EMULADOR

SEGUE - DEFINIÇÃO

O que vimos:

- ▶ Ctr + Clica + Arrasta Dentro da View Controller - Autolayout
- ▶ Ctr + Clica + Arrasta Para outra View Controller - Segue
- ▶ Ligar na Storyboard/Hierarquia de views

UM DETALHE:

SEGUES NA STORYBOARD SOMENTE PARA CLASSES QUE HERDAM DE UICONTROL

COMO FAZER COM OUTROS CASOS?

PROGRAMATICAMENTE

Gesture recognizer

Ao completar um texto, avança para próxima tela

Um botão dois caminhos

SEGUE - DEFINIÇÃO

► Programático;

```
16
17    @IBAction func acaoBotao(_ sender: Any) {
18        performSegue(withIdentifier: "SegueIdentifierProximaTela", sender: sender)
19    }
```

Antes temos que conversar sobre uma coisa



CONEXÃO
STORYBOARD CÓDIGO
OUTLETS

CONEXÃO STORYBOARD CÓDIGO

Para ligar elementos do Storyboard com o código:

- ▶ **IBOutlets** - Leva a referência do elemento visual para o código.
 - ▶ Alterar propriedades
- ▶ **IBOutlet Collections** - Agrupa as referências de um mesmo tipo de elemento visual em um array no código
- ▶ **IBActions** - Executa a função criada quando certa ação é executada com aquele elemento. Funciona para todos os elementos que herdam de UIControl.

TODAS ESSAS CONEXÕES SÃO FEITAS POR

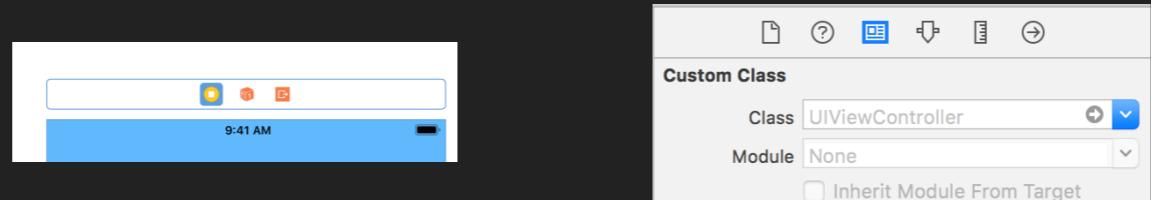
CTRL + CLICA + ARRASTA

UIButton, DatePicker, PageControl, SegmentedControl, TextField, Slider, Switch

CONEXÃO STORYBOARD CÓDIGO

Vamos criar uma conexão.

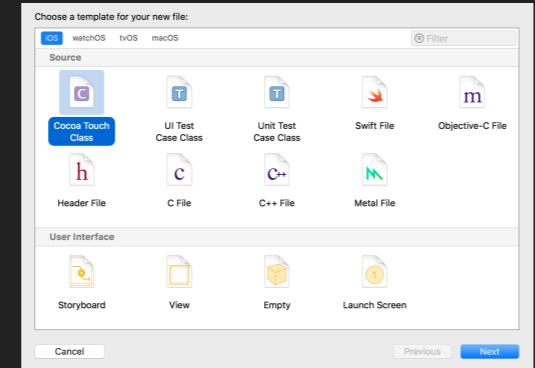
Primeiro vemos se a ViewController inicial possui uma classe:



VAMOS CRIAR

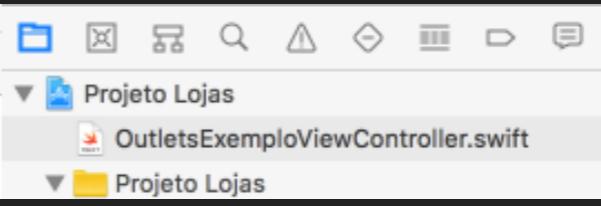
CONEXÃO STORYBOARD CÓDIGO

CMD + N



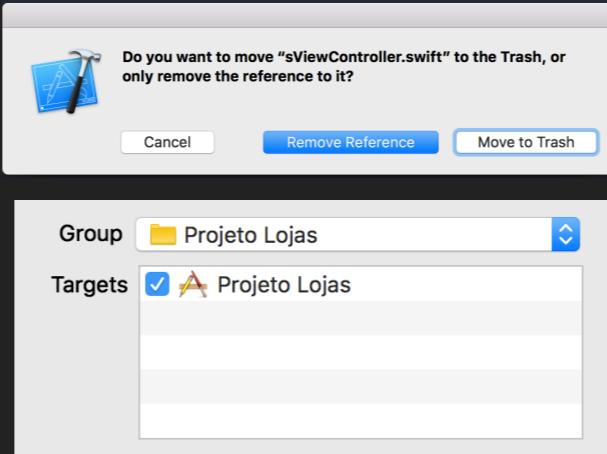
Class: Subclass of:
 Also create XIB file Language:

Group: Targets: Projeto Lojas



CONEXÃO STORYBOARD CÓDIGO

SE VOCÊ FEZ ESSE ERRO, APAGUE E CRIE DE NOVO:

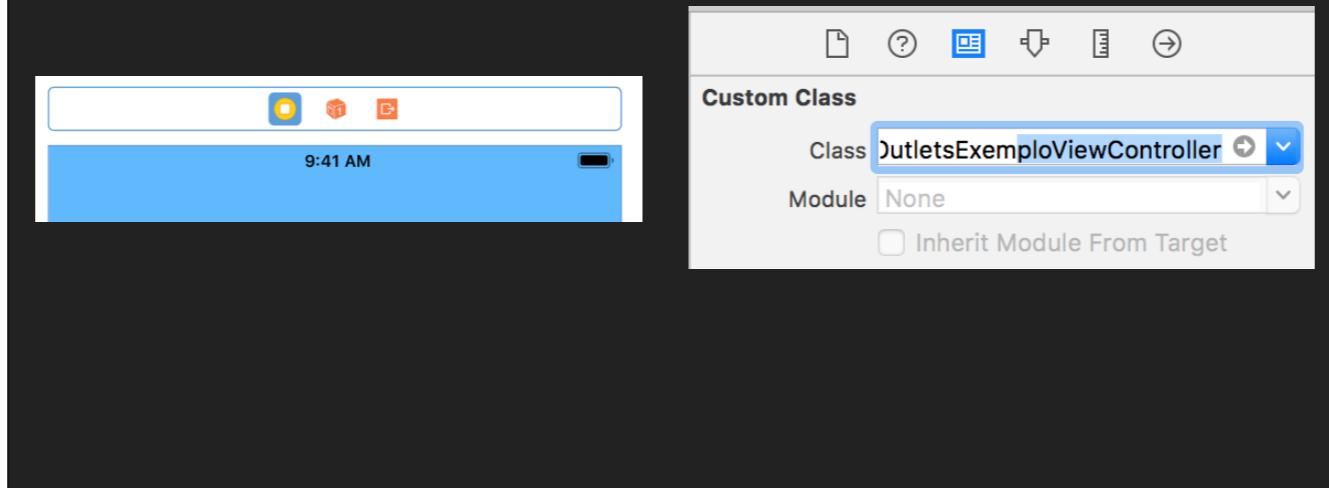


OU ARRASTE PARA A PASTA CORRETA

O importante é estar com o Target selecionado

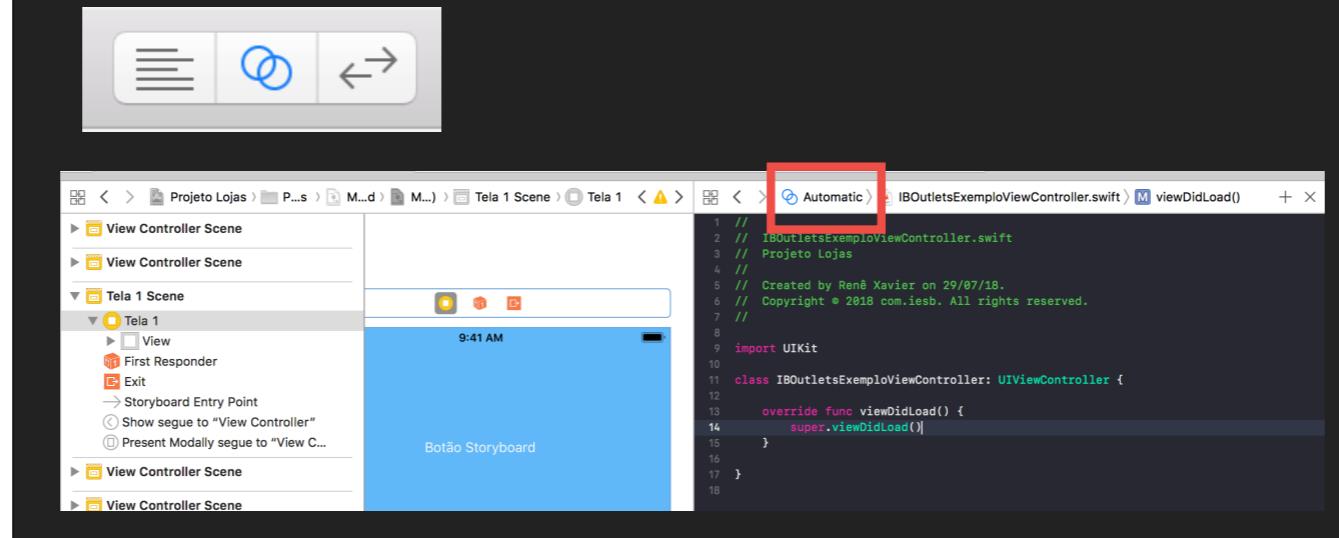
CONEXÃO STORYBOARD CÓDIGO

Voltamos ao Storyboard e referenciamos a classe na ViewController que queremos controlar.



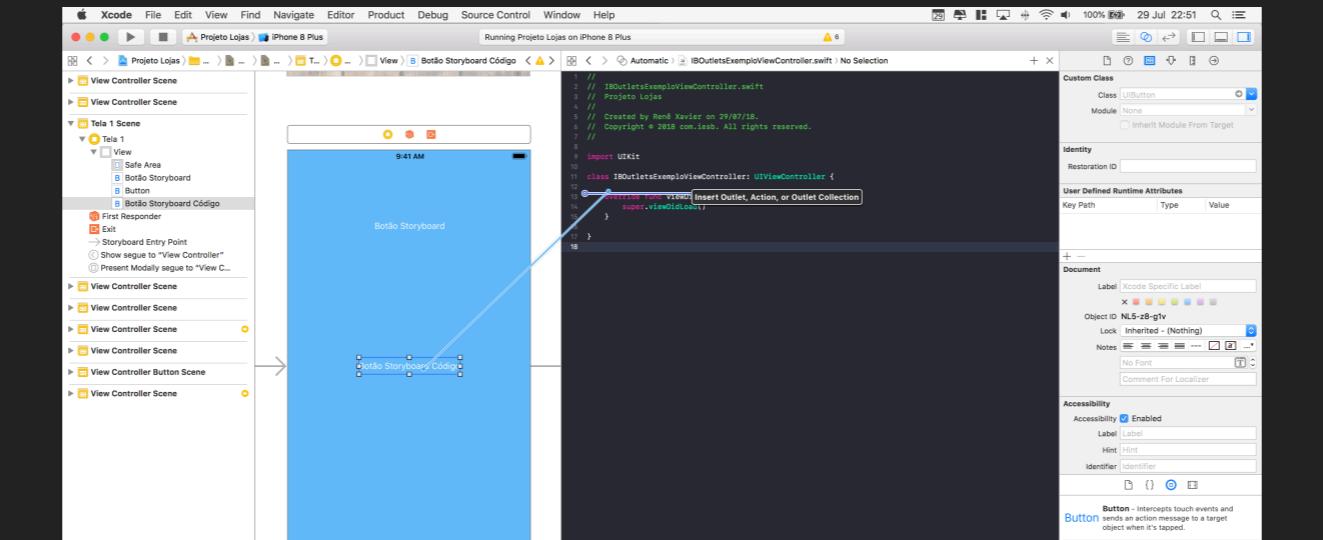
CONEXÃO STORYBOARD CÓDIGO

No Assistant Editor, ele já exibe a classe que criou. (Caso esteja selecionado o Automatic)

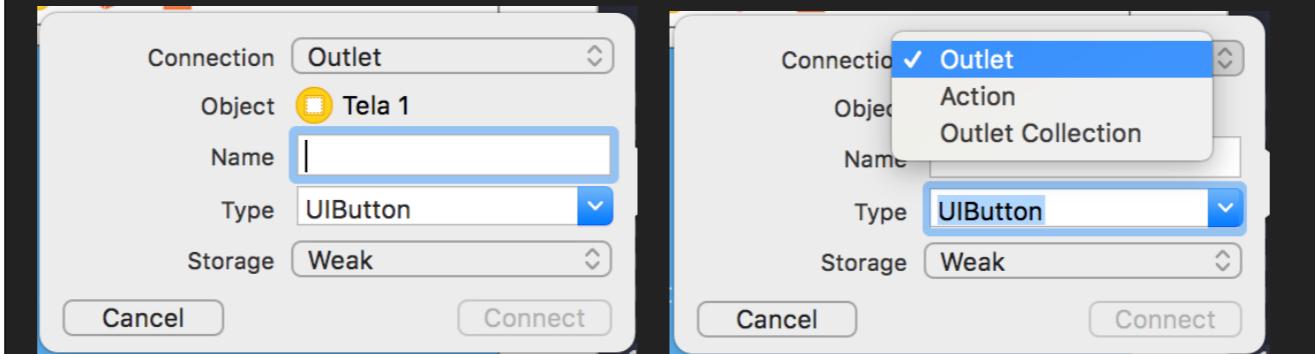


CONEXÃO STORYBOARD CÓDIGO

CTRL + CLICA + ARRASTA

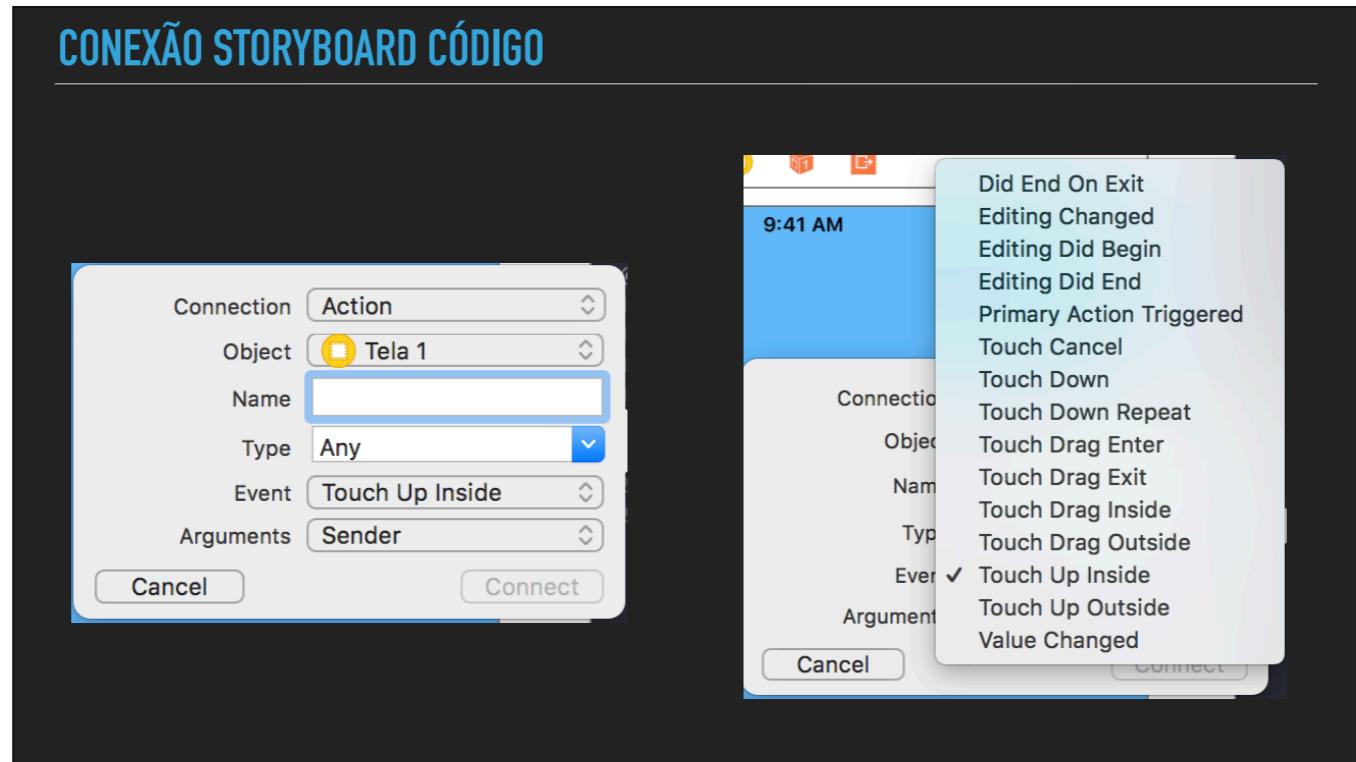


CONEXÃO STORYBOARD CÓDIGO



Falar do Type

CONEXÃO STORYBOARD CÓDIGO



Falar do argument

CONEXÃO STORYBOARD CÓDIGO

```
9 import UIKit
10
11 class IBOOutletsExemploViewController: UIViewController {
12
13     @IBOutlet weak var botaoStoryboardCodigo: UIButton!
14     @IBOutlet var listaComOsDoisBotoesDaTela: [UIButton]! ← IBOutlet Collection
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18     }
19
20     @IBAction func funcaoParaReceberCliqueNoBotao(_ sender: Any) {
21     }
22
23 }
24
```

CONEXÃO STORYBOARD CÓDIGO

Apague uma linha do código com um IBOutlet:

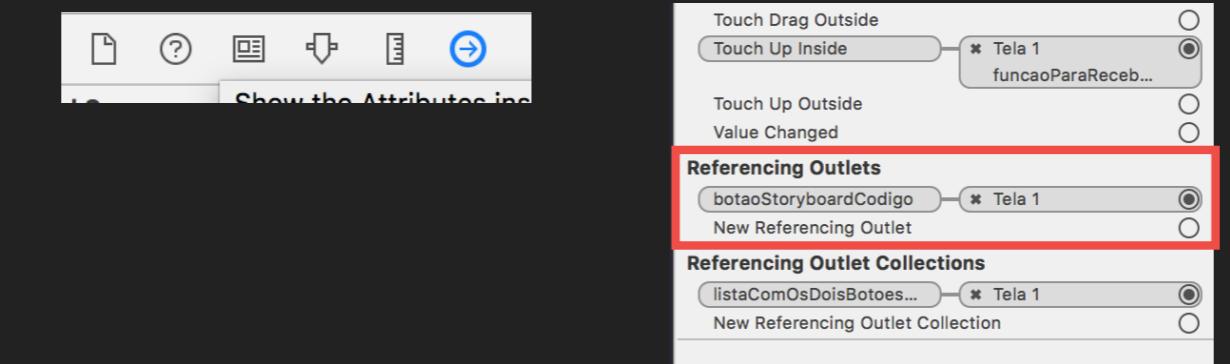
(botaoStoryboardCodigo removido)

```
12 |  
●  @IBOutlet var listaComOsDoisBotoesDaTela: [UIButton]!  
14 |
```

```
2018-07-29 23:22:31.745249-0300 Projeto Lojas[8441:10433326] ***  
Terminating app due to uncaught exception 'NSUnknownKeyException',  
reason: '[<Projeto_Lojas.IBOutletsExemploViewController 0x7fe9fb4358e0>  
setValue:forUndefinedKey:]: this class is not key value coding-compliant  
for the key botaoStoryboardCodigo.'
```

CONEXÃO STORYBOARD CÓDIGO

O outlet permanece ligado no storyboard.



Solução: Remover a referência do outlet do Storyboard ou voltar o código

CÓPIAS DO SEU PROJETO NO MAC
PODEM DAR CONFLITO

COMPRIMIR

GIT

VOLTANDO
AO QUE
QUERÍAMOS





SEGUES
PROGRAMÁTICAS

SEGUES PROGRAMÁTICAS

Conseguimos executar a transição da segue via código de algumas formas.

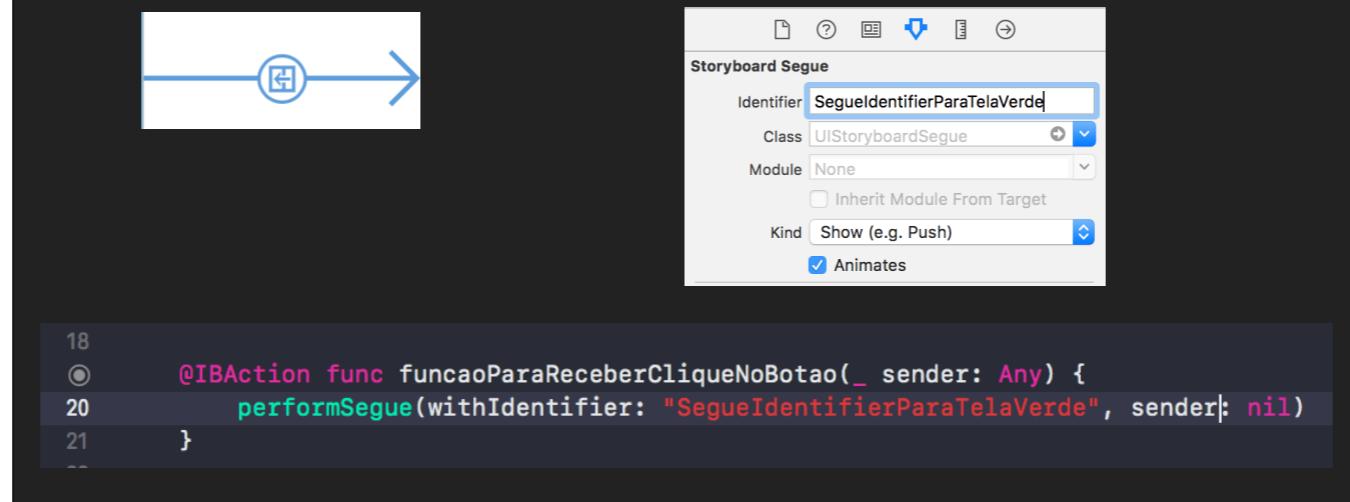
Veremos duas agora:

- ▶ Utilizando um Segue já existente
- ▶ Apresentando a referência da próxima ViewController

SEGUES PROGRAMÁTICAS

- ▶ Utilizando um Segue já existente:

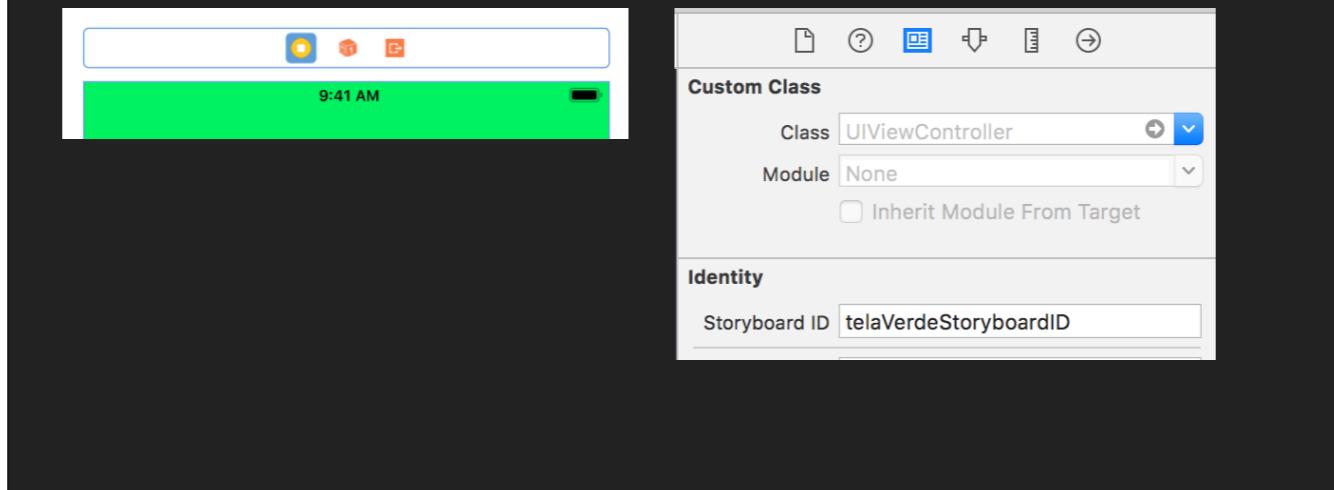
Selecionamos o Segue do Botão Storyboard e criamos um identifier



SEGUES PROGRAMÁTICAS

- ▶ Apresentando a referência da próxima ViewController:

Selecionamos a ViewController que queremos ir e criamos um Storyboard ID



Storyboard IDs são reutilizáveis

SEGUES PROGRAMÁTICAS

- ▶ Apresentando a referência da próxima ViewController:

No código utilizamos o Storyboard ID criado e o nome do Storyboard

```
@IBAction func funcaoParaReceberCliqueNoBotao(_ sender: Any) {  
    //    performSegue(withIdentifier: "SegueIdentifierParaTelaVerde", sender: nil)  
  
    let storyboard = UIStoryboard(name: "Main", bundle: nil)  
    let telaVerdeViewController = storyboard.instantiateViewController(withIdentifier:  
        "telaVerdeStoryboardID")  
    present(telaVerdeViewController, animated: true, completion: nil)  
}
```

Storyboard IDs são reutilizáveis

Storyboard ID vs Segue Identifier reforçar

SEGUES PROGRAMÁTICAS

- ▶ Podemos colocar uma lógica para determinar se a Segue do Storyboard deve executar com:

```
27    override func shouldPerformSegue(withIdentifier identifier: String, sender: Any?) -> Bool {  
28        // lógica para determinar se deve executar  
29        return true  
30    }
```

- ▶ Basta retornar **false** no método **shouldPerformSegue(withIdentifier: String? sender: Any?) -> Bool**
- ▶ O **identifier** é aquele que configuramos no Storyboard;
- ▶ O **sender** é o objeto que provocou a segue (ex.: um botão)

SEGUES PROGRAMÁTICAS

- ▶ Segues programáticas (performSegue, present) não chamam esse método:

```
27    override func shouldPerformSegue(withIdentifier identifier: String, sender: Any?) -> Bool {  
28        // lógica para determinar se deve executar  
29        return true  
30    }
```

VAMOS TESTAR ISSO?

Substituir por false

SEGUES PROGRAMÁTICAS

- ▶ Um método é chamado logo antes de uma segue:

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if segue.identifier == "viewControllerBSegueIdentifier" {  
        // lógica antes de executar a segue  
    }  
}
```

- ▶ Segues programáticas (performSegue, present) não chamam esse método

Pode usar para salvar algo no BD ou enviar parâmetros



TIPOS DE

SEGUES

SEGUE - DEFINIÇÃO

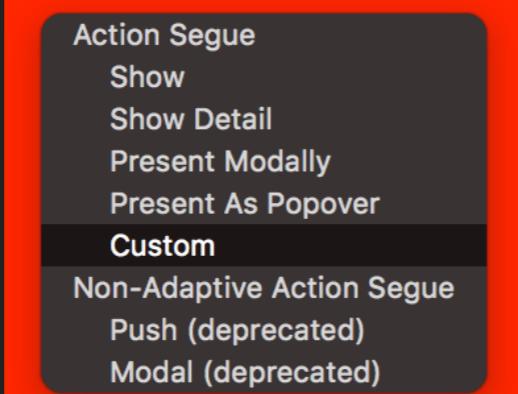
Action Segue
Show
Show Detail
Present Modally
Present AsPopover
Custom
Non-Adaptive Action Segue
Push (deprecated)
Modal (deprecated)

SEGUE - DEFINIÇÃO

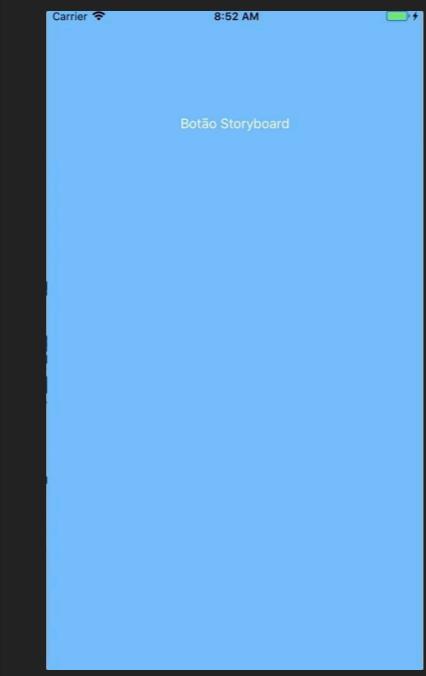
SHOW

Apresenta a ViewController substituindo a anterior

Mais usada



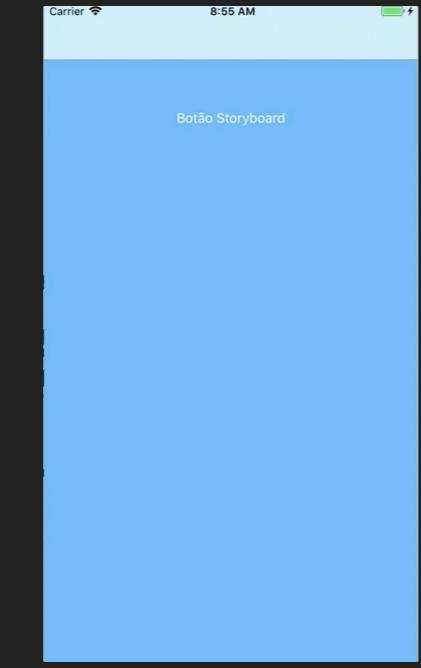
SEGUE - DEFINIÇÃO



SHOW

Apresenta a ViewController substituindo a anterior

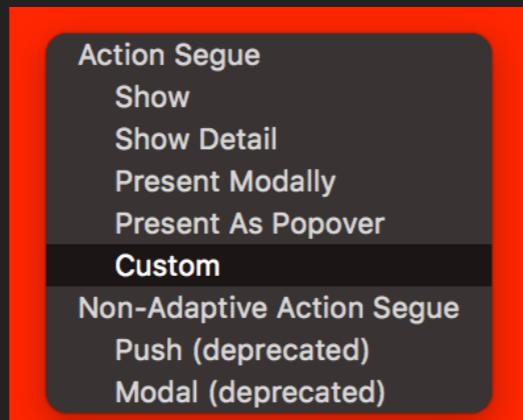
SEGUE - DEFINIÇÃO



SHOW

Apresenta a ViewController substituindo a anterior

SEGUE - DEFINIÇÃO



SHOW DETAIL

Quando se tem um SplitViewController (iPad/horizontal), ela substitui a ViewController do detalhe

Nos outros casos, funciona como a show

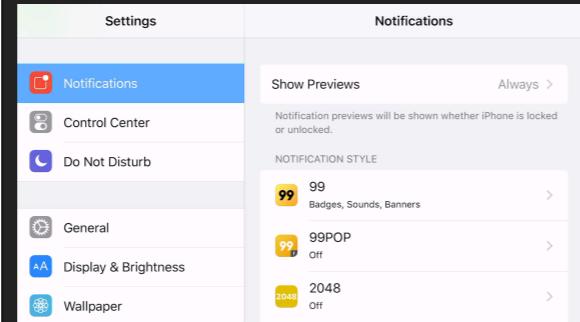
Usar como show não é recomendado

SEGUE - DEFINIÇÃO

SHOW DETAIL

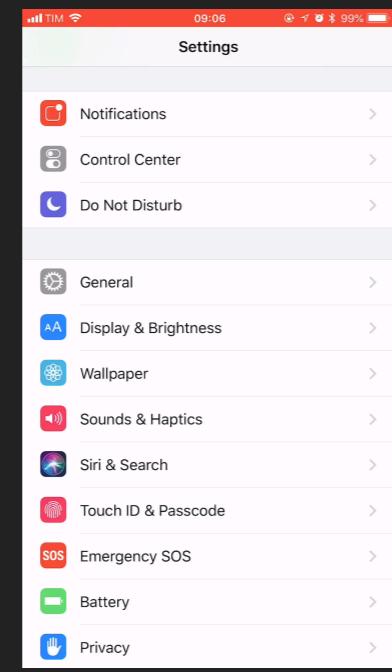
Quando se tem um SplitViewController (iPad/horizontal), ela substitui a ViewController do detalhe

Nos outros casos, funciona como a show



Usar como show não é recomendado

SEGUE - DEFINIÇÃO

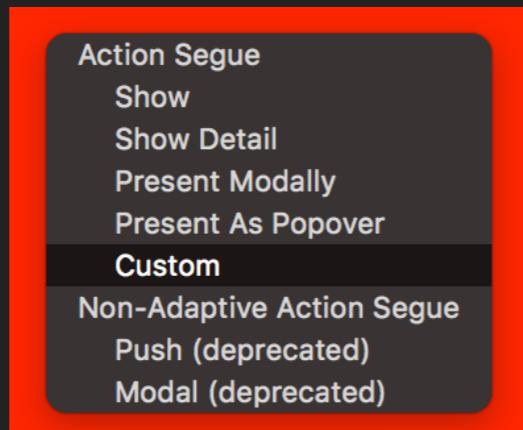


SHOW DETAIL

Quando se tem um SplitViewController (iPad),
ela substitui a ViewController do detalhe

Nos outros casos, funciona como a show

SEGUE - DEFINIÇÃO

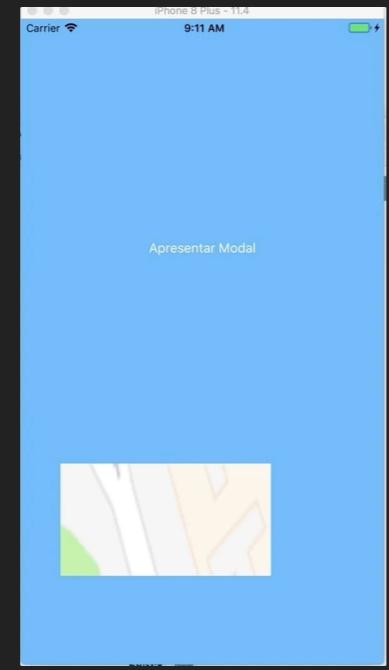


PRESENT MODALLY

Apresenta uma ViewController completamente por cima da atual.

Pop ups

SEGUE - DEFINIÇÃO

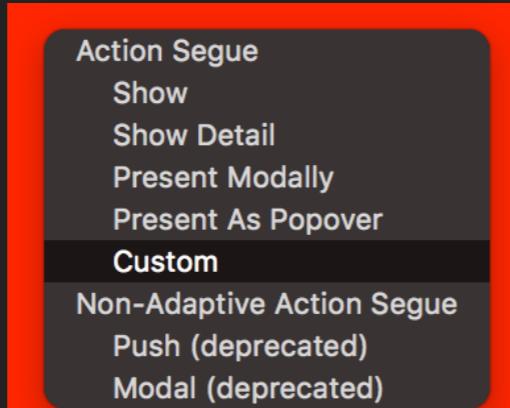


PRESENT MODALLY

Apresenta uma ViewController completamente por cima da atual.

Pop ups

SEGUE - DEFINIÇÃO



PRESENT AS POPOVER

Quando se está em um iPad, apresenta a ViewController parcialmente por cima da atual, mas ligada à um elemento visual

Nos outros casos, apresenta a ViewController completamente por cima da atual (como a Modally)

iPad ao clicar fora dismiss

SEGUE - DEFINIÇÃO



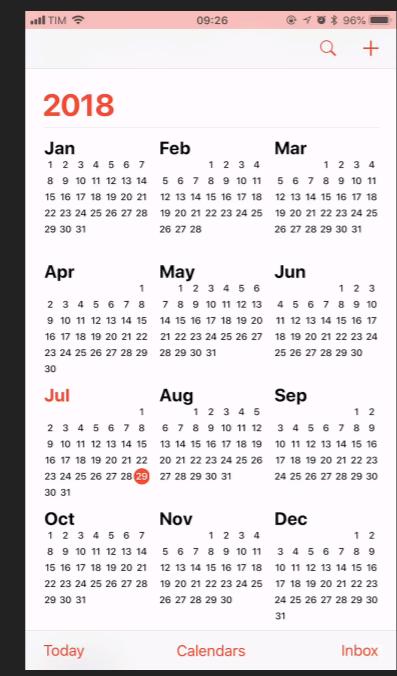
PRESENT AS POPOVER

Quando se está em um iPad, apresenta a ViewController por cima da atual, mas ligada à um elemento visual

Nos outros casos, apresenta a ViewController completamente por cima da atual (como a Modally)

iPad ao clicar fora dismiss

SEGUE - DEFINIÇÃO



PRESERVAR AS POPOVERS

Quando se está em um iPad, apresenta a ViewController por cima da atual, mas ligada à um elemento visual

Nos outros casos, apresenta a ViewController completamente por cima da atual (como a Modally)

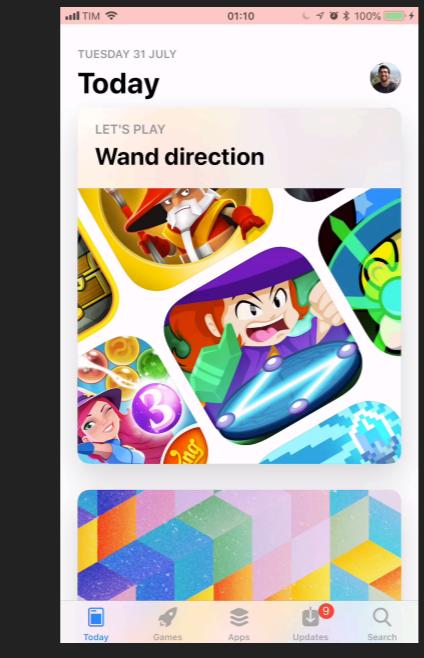
SEGUE - DEFINIÇÃO

CUSTOM

Controle total sobre a aparência e a forma da transição

- Action Segue
- Show
- Show Detail
- Present Modally
- Present AsPopover
- Custom
- Non-Adaptive Action Segue
- Push (deprecated)
- Modal (deprecated)

SEGUE - DEFINIÇÃO



CUSTOM

Controle total sobre a aparência e a forma da transição

COMO FAZER PARA VOLTAR PARA A VIEW CONTROLLER ANTERIOR?

```
self.dismiss(animated: true, completion: nil)
```



TAB BAR CONTROLLER

UITABBARCONTROLLER

► Deixa você escolher entre MVCs diferentes...

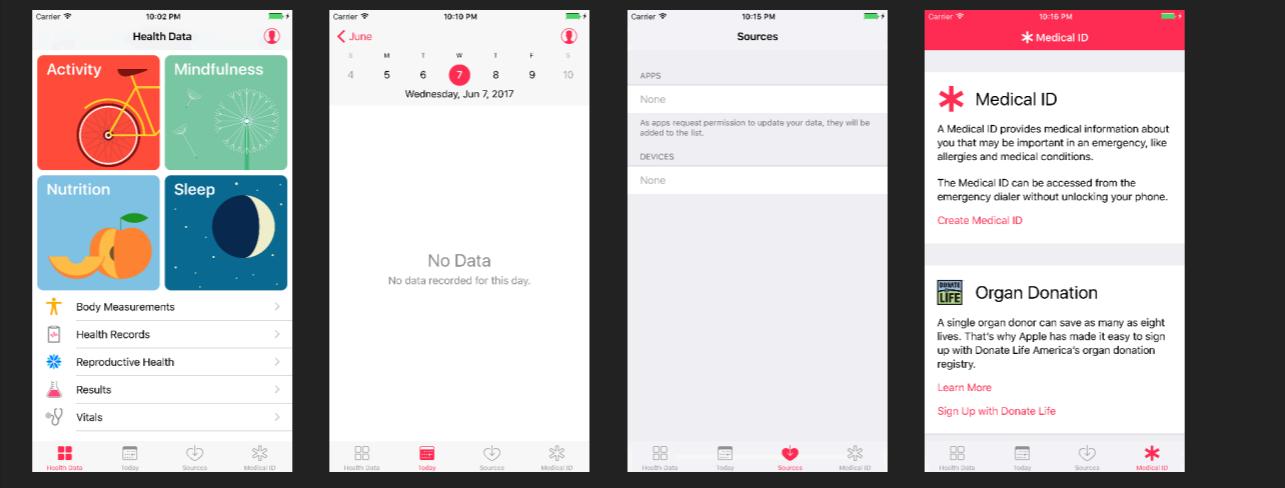


UM MVC CHAMADO "HEALTH DATA"

O ícone e o título podem ser configurados através do storyboard (mais comum) ou programaticamente, através da propriedade tabBarItem, que todo TabBarController tem.

UITABBARCONTROLLER

► Deixa você escolher entre MVCs diferentes...

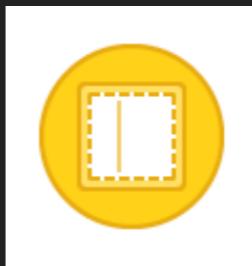
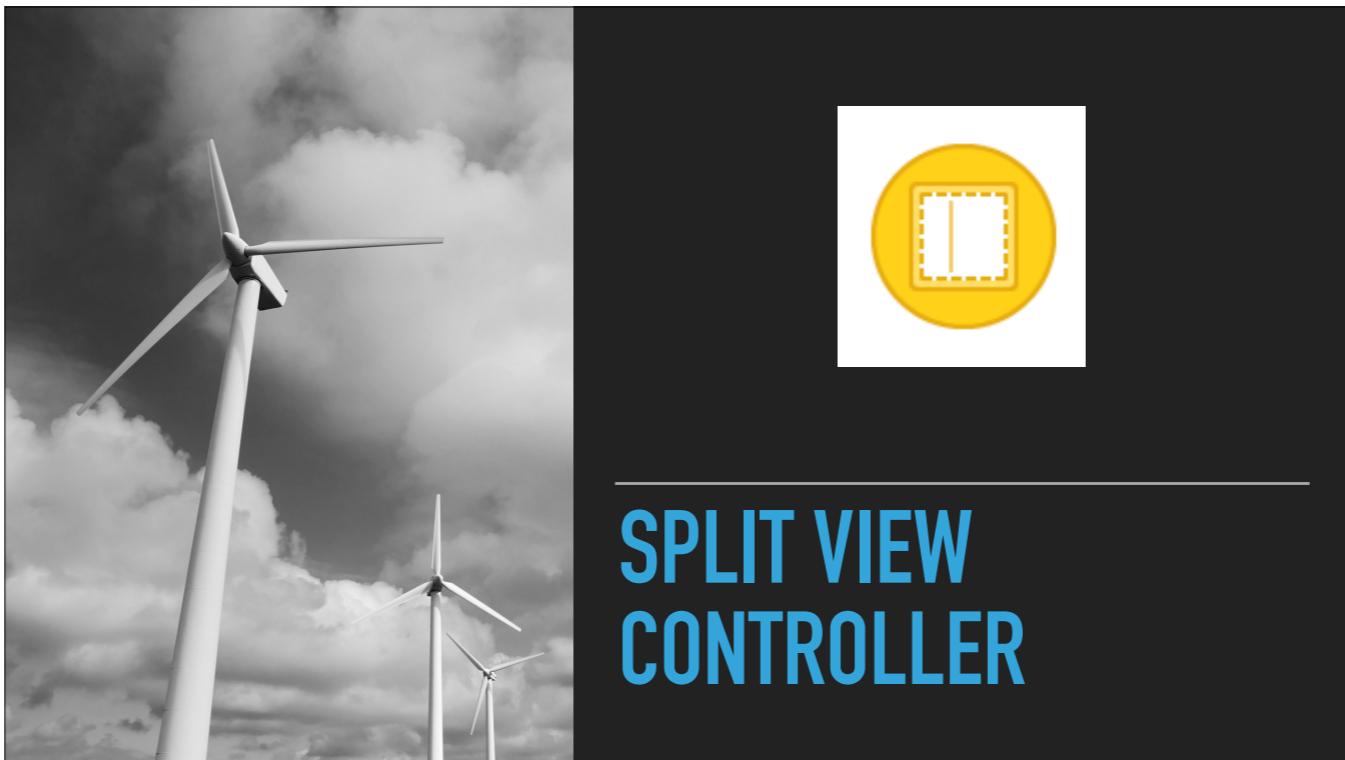


UITABBARCONTROLLER

```
tabBar.selectedIndex = 1
```

APPDELEGATE

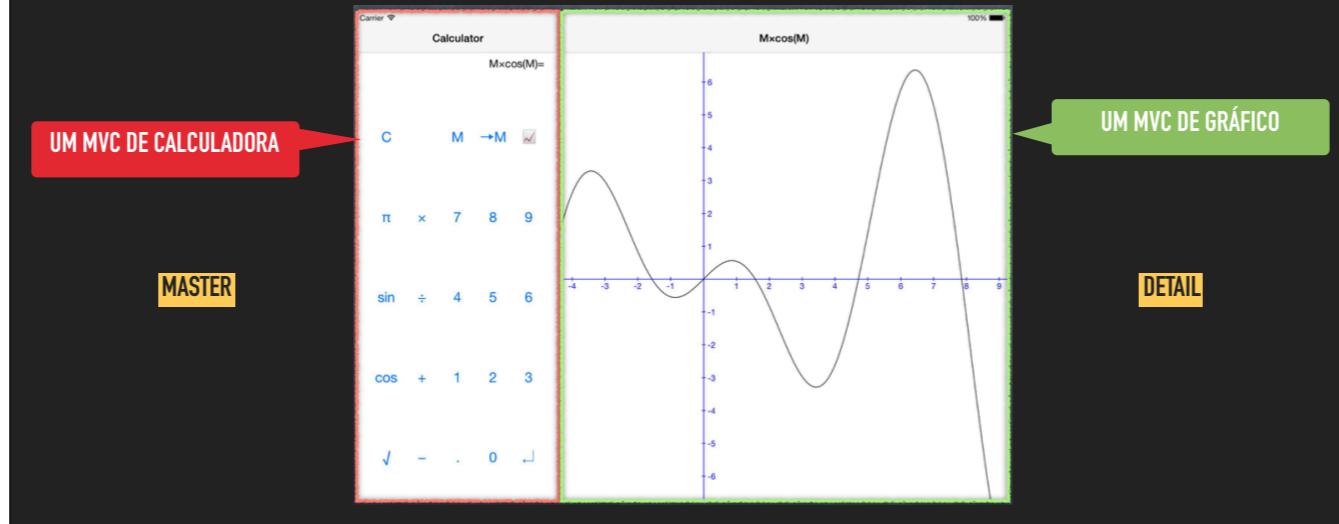
```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {  
    let tabBar = self.window?.rootViewController as! UITabBarController  
    tabBar.selectedIndex = 1  
    return true  
}
```



SPLIT VIEW
CONTROLLER

UISPLITVIEWCONTROLLER

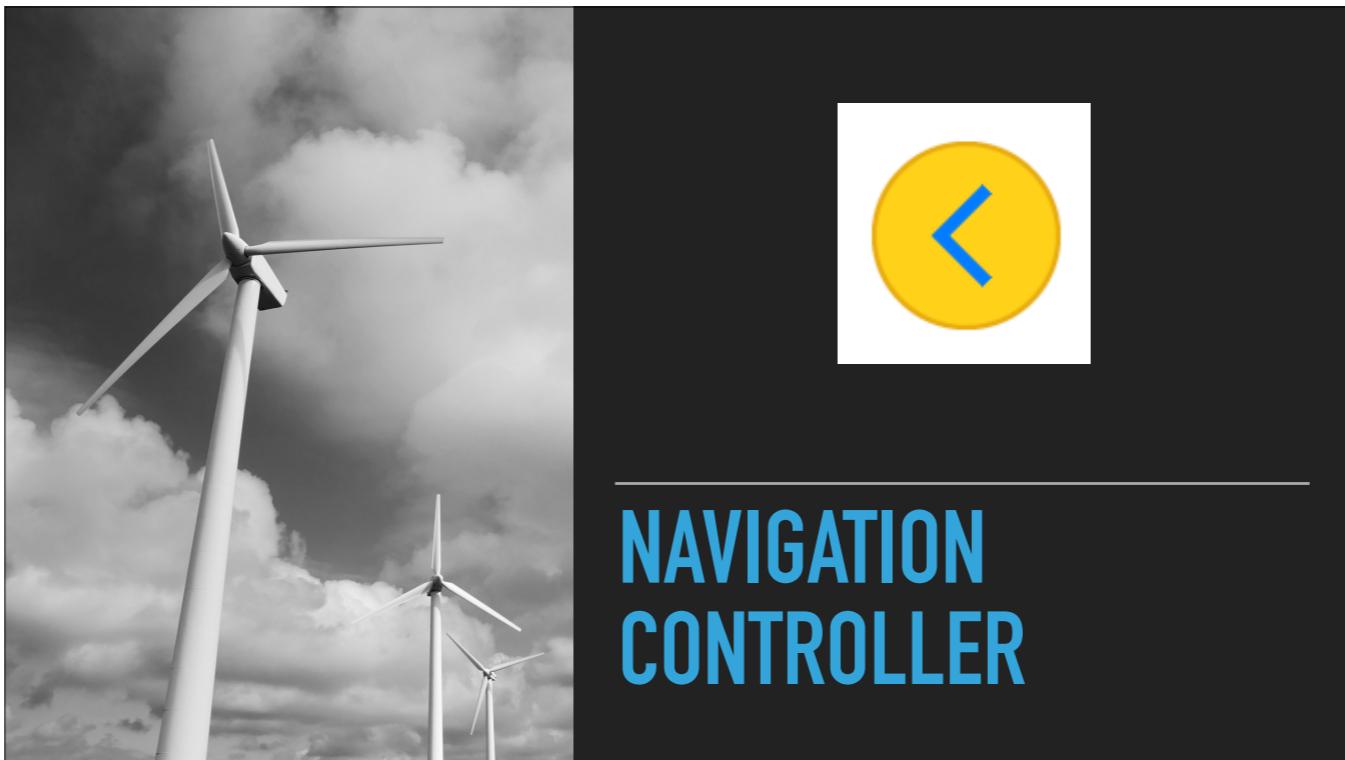
- Coloca dois MVCs lado-a-lado...



Se lembram do show Detail?

UISPLITVIEWCONTROLLER

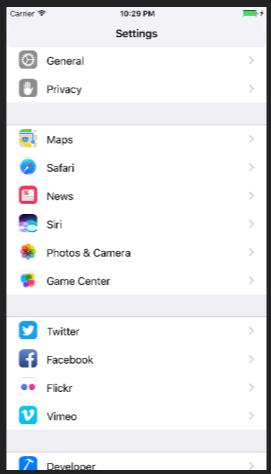
```
let storyboard = UIStoryboard(name: "Main", bundle: nil)
let detailViewController = storyboard.instantiateViewController(withIdentifier: "detailViewControllerStoryboardID")
self.showDetailViewController(detailViewController, sender: nil)
```



NAVIGATION CONTROLLER

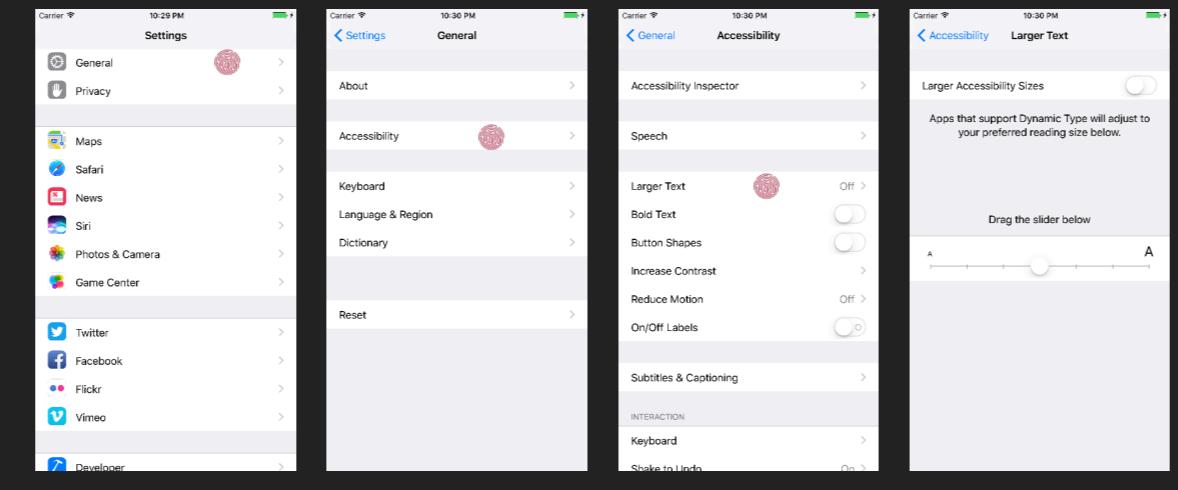
UINavigationController

- ▶ Empilha outros MVCs, funcionando como uma pilha de cartas...



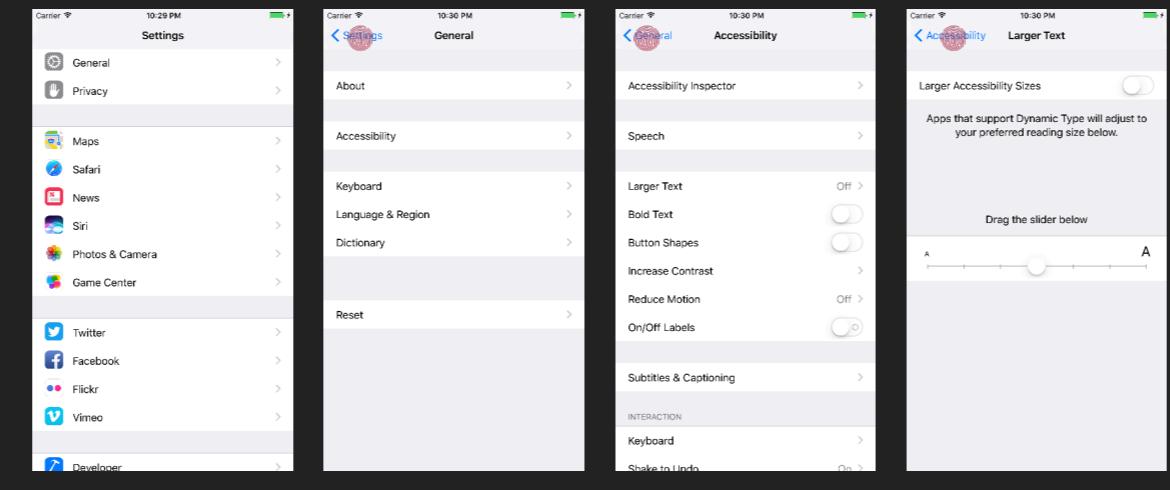
UINavigationController

► Empilha outros MVCs, funcionando como uma pilha de cartas...



UINavigationController

► Empilha outros MVCs, funcionando como uma pilha de cartas...



UINavigationController

```
let storyboard = UIStoryboard(name: "Main", bundle: nil)
let viewControllerB = storyboard.instantiateViewController(withIdentifier: "viewControllerBStoryboardID")
self.navigationController?.pushViewController(viewControllerB, animated: true)
```

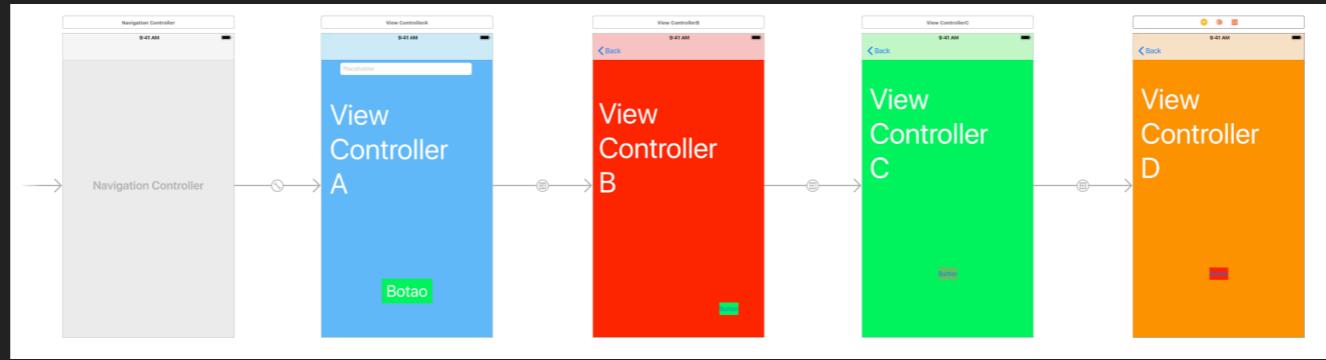
```
self.navigationController?.popViewController(animated: true)
```

```
self.navigationController?.popToRootViewController(animated: true)
```

Não é mais present

UINavigationController

- Unwind Segues permitem você voltar para uma ViewController específica.
- Digamos da D para a B



UINavigationController

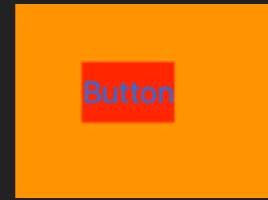
View
Controller
B

- ▶ Na classe da ViewControllerB criamos uma IBAction
- ▶ O tipo (classe) da segue que será recebido DEVE ser UIStoryboardSegue
- ▶ NÃO conectaremos com o storyboard

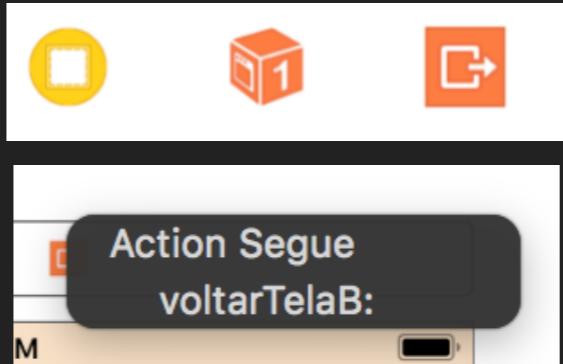
○ @IBAction func voltarTelaB(_ segue: UIStoryboardSegue) {
26 }

UINavigationController

View
Controller
D



- ▶ Na classe da ViewControllerD criamos um Button no Storyboard (que ativará a ação de voltar)
- ▶ Usamos do Ctrl + Clique + Arrasta do botão até o Exit
- ▶ Selecione o método criado na tela B





TRABALHANDO EM
CONJUNTO

TRABALHANDO EM CONJUNTO

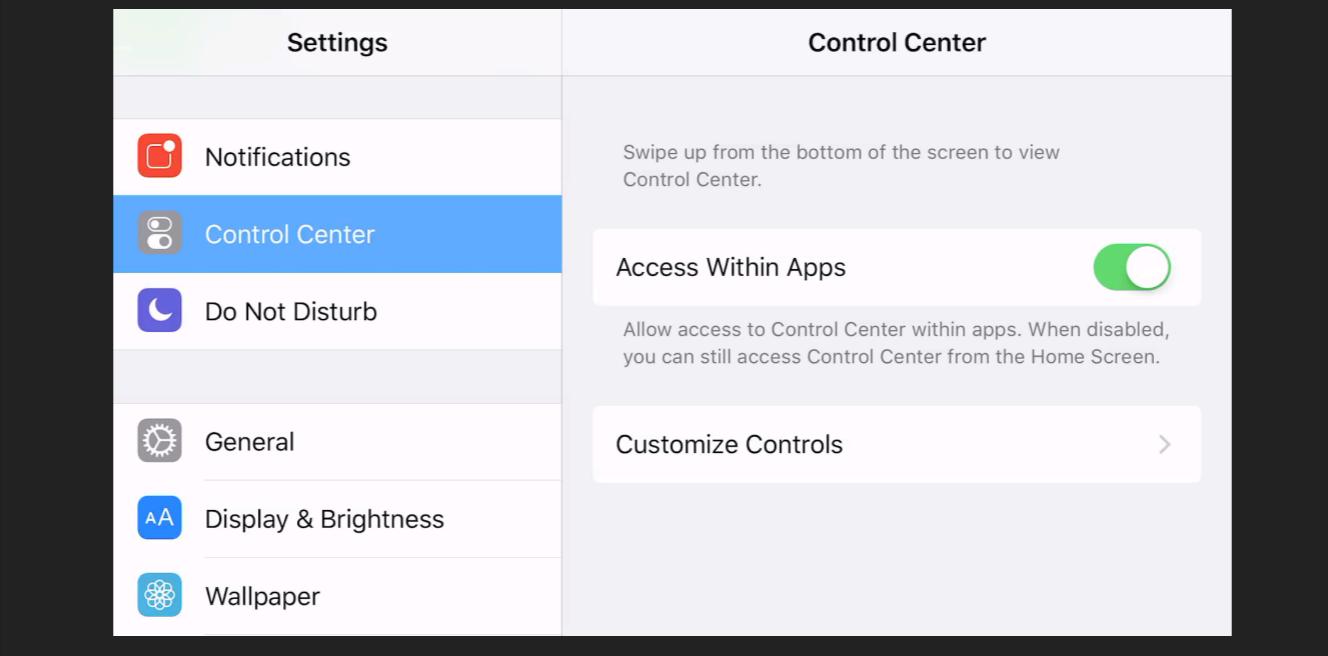
| Folders | Edit | |
|------------|------|--|
| ICLOUD | | |
| All iCloud | 55 > | |
| Notes | 27 > | |
| Android | 2 > | |
| BTC | 2 > | |
| Everton | 1 > | |
| Filmes | 1 > | |
| | | |
| | | |
| | | |

Jogos campus party
[https://github.com/wfleshman/
Evolving_To_Walk](https://github.com/wfleshman/Evolving_To_Walk)
[https://github.com/tvdboom/Reinforcement-
Learning](https://github.com/tvdboom/Reinforcement-Learning)

New Folder

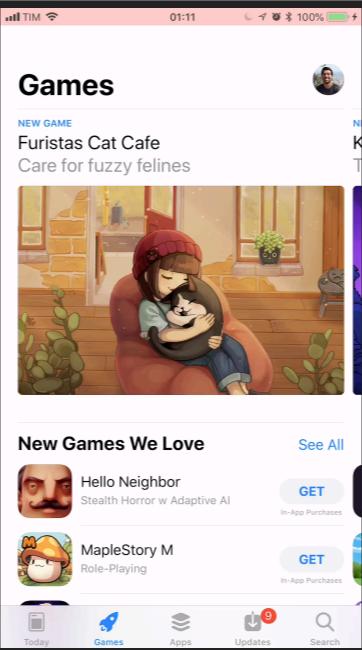
Split + Navigation

TRABALHANDO EM CONJUNTO



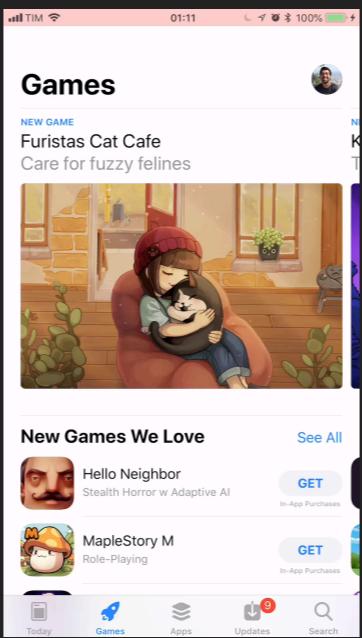
Split + Navigation

TRABALHANDO EM CONJUNTO



Tab + Navigation

TRABALHANDO EM CONJUNTO



Tab + Present Modally