

# Exercício Revisão

Utilizando StoryBoard

X-Code Swift

Prof. Agesandro Scarpioni

[agesandro@fiap.com.br](mailto:agesandro@fiap.com.br)



# Esconder e Exibir

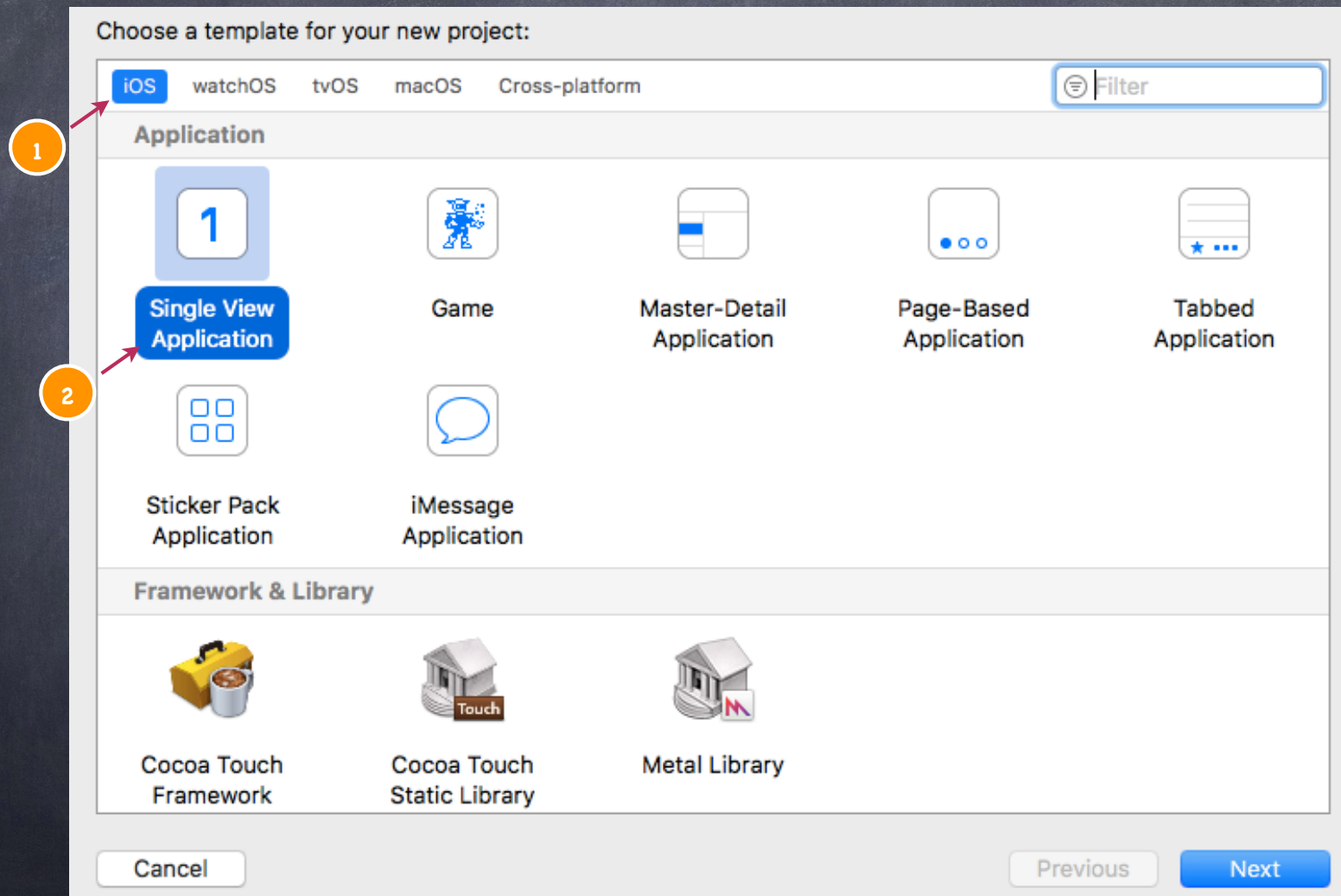
com uso de StoryBoard

- Vamos recordar o exercício de Nome e Idade porém ao invés de mostrar conteúdo dentro de uma label vamos trabalhar com a propriedade alpha para esconder ou exibir uma componente.



# Single View

- Vamos criar um projeto novo do tipo IOS (1) – Single View Application (2)



# Nomeando Projeto

- Nomeie o projeto como “Esconda Exibe” (1), escolha o device iPhone(2).

Choose options for your new project:

1 → Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

2 → Devices:

☐ Use Core Data

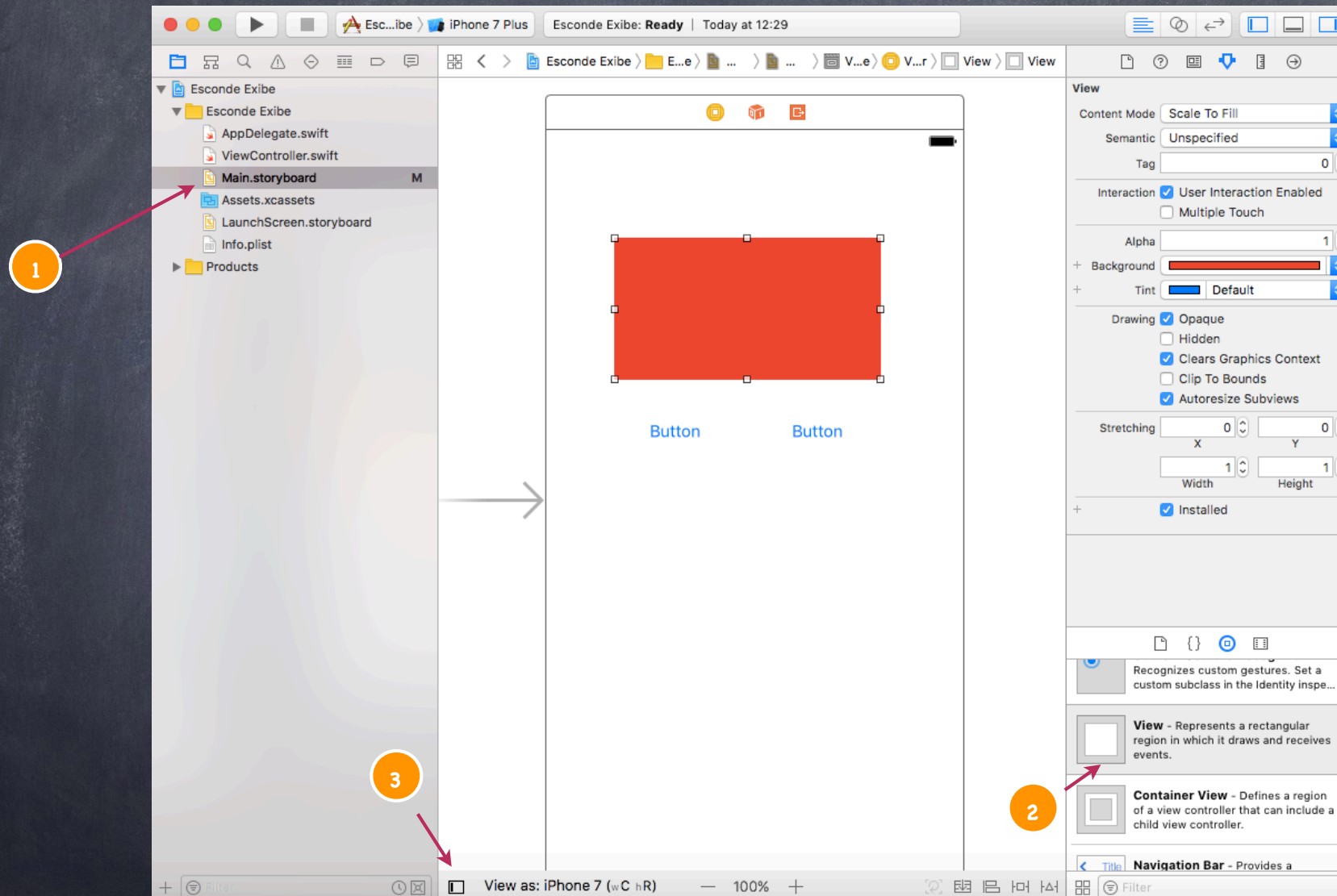
☐ Include Unit Tests

☐ Include UI Tests



# Storyboard

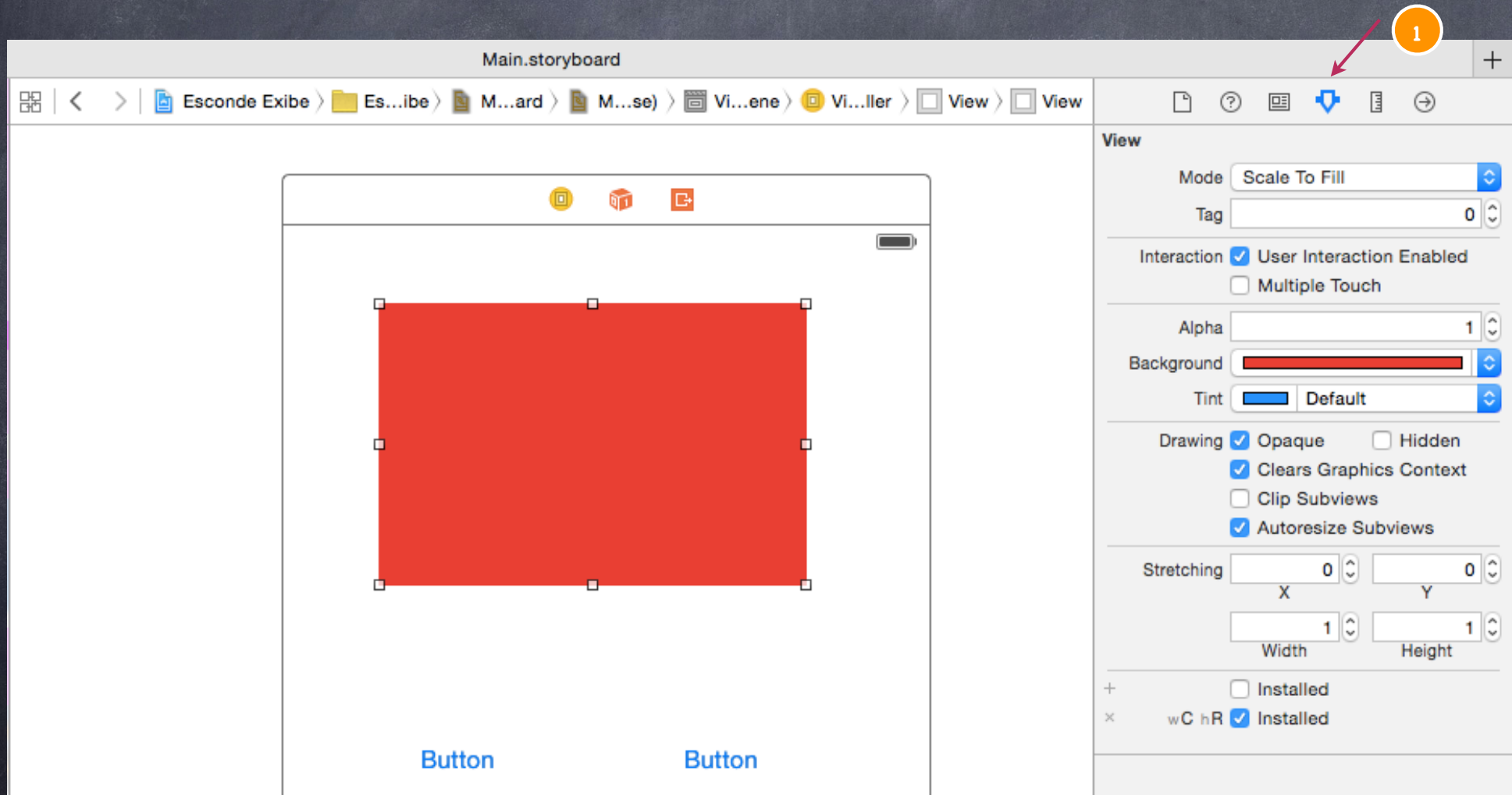
- Vá em StoryBoard(1), inclua um componente view (2) e mais dois botões, clique em (3) para recolher uma janela e ficar com mais espaço.



Obs: Uma View é um objeto container, nele podemos inserir outros objetos.

# Attributes Inspector

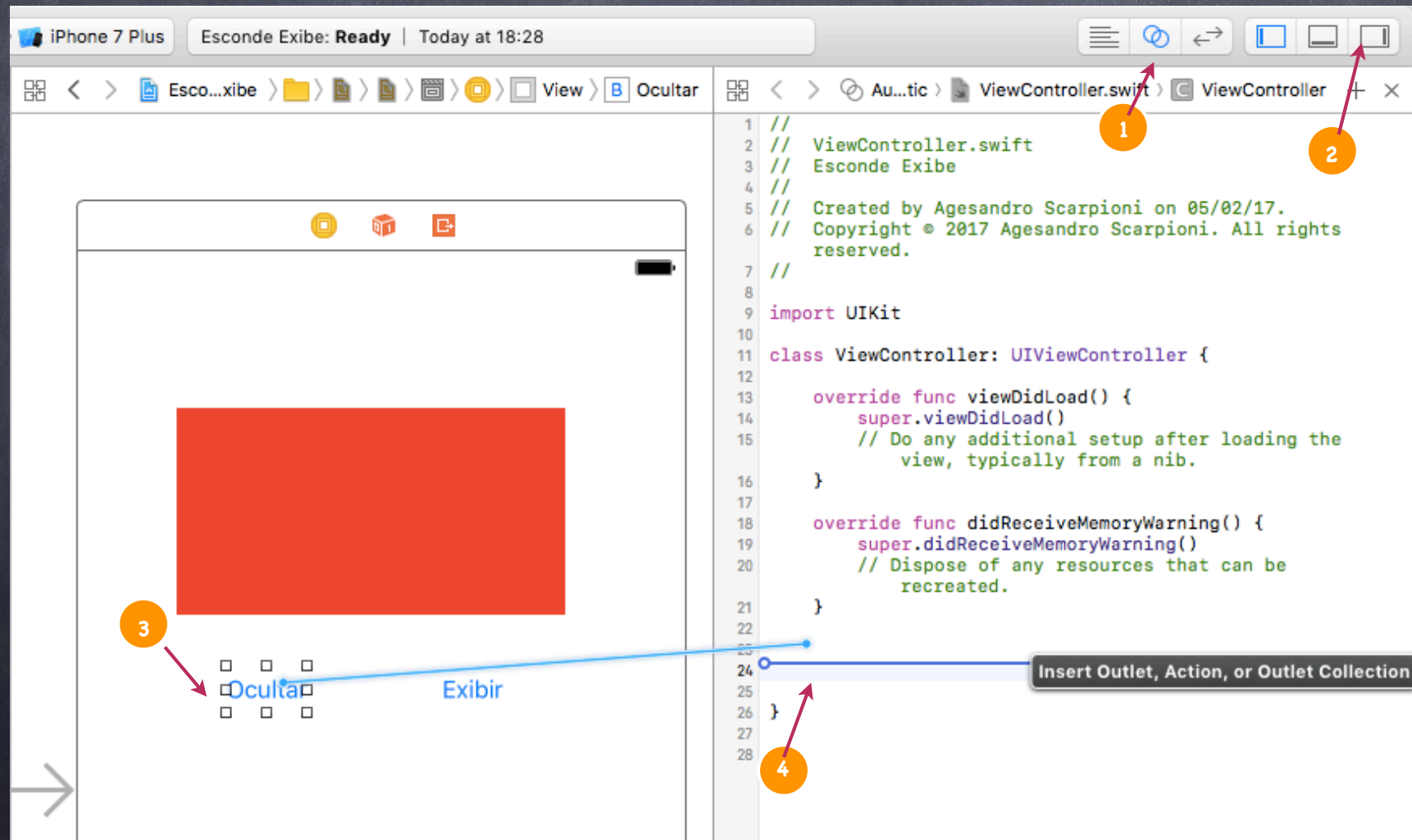
- Vá até o Attributes Inspector (1) e altere o Background do componente View, e altere os nomes dos botões como na imagem abaixo.





# Definindo os Actions

- Primeiro deixe aberto simultaneamente as telas de storyboard e a tela.h passos (1) e (2). Vamos criar o Action de duas formas, a primeira selecionamos o botão ocultar (3) e com o control pressionado arrastamos até a área indicada abaixo (4).



# Definindo os Actions

- Irá aparecer uma janela, altere o tipo de conexão para Action (1), nomeie como ocultarView (2) e clique em connect.

The screenshot shows the Xcode interface with a storyboard on the left and a Swift file on the right. A 'Connect' dialog box is open in the center, with three numbered callouts: 1 points to the 'Connection' dropdown set to 'Action', 2 points to the 'Name' text field containing 'ocultarView', and 3 points to the 'Connect' button. The dialog also shows 'Object' as 'View Controller', 'Type' as 'Any', 'Event' as 'Touch Up Inside', and 'Arguments' as 'Sender'. The Swift file on the right shows the following code:

```
1 //  
2 // ViewController.swift  
3 // Esconde Exibe  
4 //  
5 // Created by Agesandro Scarpioni on 05/02/17.  
6 // Copyright © 2017 Agesandro Scarpioni. 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  
16         // view, typically from a nib.  
17     }  
18  
19     override func didReceiveMemoryWarning() {  
20         super.didReceiveMemoryWarning()  
21         // Dispose of any resources that can be  
22         // recreated.  
23     }  
24  
25 }  
26  
27  
28
```



# Definindo os Actions

- Agora vamos criar o Action de uma segunda forma, veja o que você prefere. Clique com o botão direito sobre o botão Exibir e escolha Touch Up Inside (1) arrastando até a área indicada (2).

The screenshot illustrates the steps to define an action for a button in Xcode. The interface shows a red button labeled "Exibir" and a blue button labeled "Ocultar". A context menu is open over the "Exibir" button, with "Touch Up Inside" selected (1). A blue arrow points from the selected menu item to the "Insert Action" button (2). The "Insert Action" dialog is open, showing "View Controller" as the object, "exibirView" as the name (3), "Any" as the type, "Touch Up Inside" as the event, and "Sender" as the argument. The "Connect" button is highlighted (4). The background shows the Swift code for ViewController.swift.

```
// ViewController.swift
// Esconde Exibe
//
// Created by Agesandro Scarpioni on 05/02/17.
// Copyright © 2017 Agesandro Scarpioni. All rights reserved.
//
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBAction func ocultarView(_ sender: UIButton) {
        // ...
    }
}
```



# Definindo o Outlet

- Precisamos associar o UIView ao código, para isso vamos criar um Outlet. Selecione o UIView(1), segure o CTRL e arraste até a área indicada(2), nomeie (3) conforme a imagem abaixo e conecte (4).

```
// ViewController.swift
// Esconde Exibe
//
// Created by Agesandro Scarpioni on 05/02/17.
// Copyright © 2017 Agesandro Scarpioni. All rights reserved.
//
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the
        // view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be
        // recreated.
    }

    @IBAction func ocultarView(_ sender: Any) {
    }

    @IBAction func exibirView(_ sender: Any) {
    }

}
```



# Arquivo .swift

- O arquivo .swift ficará com os seguintes códigos de Actions e Outlets gerados automaticamente, observe que já estão conectados (1).

```
1 //
2 // ViewController.swift
3 // Esconde Exibe
4 //
5 // Created by Agesandro Scarpioni on 05/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. 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, typically from a nib.
16     }
17
18     override func didReceiveMemoryWarning() {
19         super.didReceiveMemoryWarning()
20         // Dispose of any resources that can be recreated.
21     }
22
23
24     @IBAction func ocultarView(_ sender: Any) {
25     }
26
27     @IBAction func exibirView(_ sender: Any) {
28     }
29
30     @IBOutlet weak var minhaView: UIView!
31
32 }
33
34
```

1



# Arquivo .swift

- Vamos implementar os Actions alterando a propriedade alpha para exibir e ocultar o UIView.

```
1 //
2 // ViewController.swift
3 // Esconde Exibe
4 //
5 // Created by Agesandro Scarpioni on 05/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. 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, typically from a nib.
16     }
17
18     override func didReceiveMemoryWarning() {
19         super.didReceiveMemoryWarning()
20         // Dispose of any resources that can be
          recreated.
21     }
22
23
24     @IBAction func ocultarView(_ sender: Any) {
25         minhaView.alpha = 0
26     }
27
28     @IBAction func exibirView(_ sender: Any) {
29         minhaView.alpha = 1
30     }
31
32     @IBOutlet weak var minhaView: UIView!
33
34 }
35
36
```

1



# Teste seu App

- Command + R e veja o resultado.

```
1 //
2 // ViewController.swift
3 // Esconde Exibe
4 //
5 // Created by Agesandro Scarpioni on 05/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. 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, typically from a nib.
16     }
17
18     override func didReceiveMemoryWarning() {
19         super.didReceiveMemoryWarning()
20         // Dispose of any resources that can be
           recreated.
21     }
22
23
24     @IBAction func ocultarView(_ sender: Any) {
25         minhaView.alpha = 0
26     }
27
28     @IBAction func exibirView(_ sender: Any) {
29         minhaView.alpha = 1
30     }
31
32     @IBOutlet weak var minhaView: UIView!
33
34 }
35
36
```



# Animando

- Inclua uma linha antes e uma linha depois dentro de cada IBAction e teste novamente.

```
23
24  @IBAction func ocultarView(_ sender: Any) {
25      UIView.beginAnimations(nil, context: nil)
26      minhaView.alpha = 0
27      UIView.commitAnimations()
28  }
29
30  @IBAction func exibirView(_ sender: Any) {
31      UIView.beginAnimations(nil, context: nil)
32      minhaView.alpha = 1
33      UIView.commitAnimations()
34  }
35
```

Diagram illustrating the code structure for animating views. Two callouts, labeled 1 and 2, point to the first and second lines of code in the two IBAction methods respectively.

OBS: A animação é apenas um FADE na tela.



# Resultado

- Depois desses passos relembramos como fizemos em nosso último conjunto de slides, realizando os passos para desenhar e ligar a interface com seus Actions e Outlets de forma rápida e sem digitação dos códigos ou ligação posterior de seus componentes aos códigos.