



# Hilton Pintor

Desenvolvedor (iOS/tvOS/watchOS)



hiltonpintor@gmail.com



# // Aula 07



# // Dúvidas da Aula 06



/\*

Como passar a célula  
escolhida?

\*/

// algoritmo

1. Criar **segue** da célula para a próxima tela
  1. Dar um **identificador** para a segue
2. Override do **prepare(for:sender:)**
  - 2.1. identificar **qual segue** foi chamada
  - 2.2. **passar dados** para VC destino
3. Atualizar interface



// 1. VCs

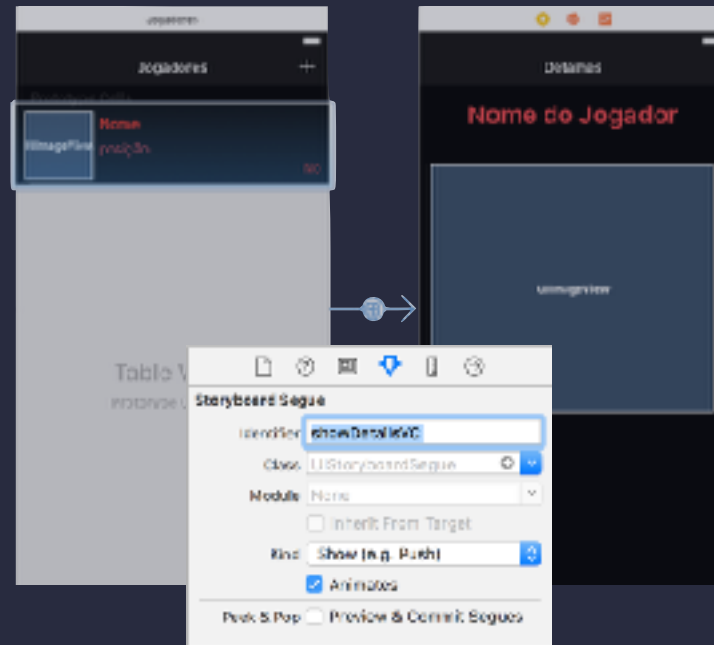


PlayerTableViewController

DetailsViewController



// 1. segue





// 2. override

```
override fun prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    super.prepare(for: segue, sender: sender)  
    guard let segueIdentifier = segue.identifier else {  
        return  
    }  
}
```



// 2. override

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    super.prepare(for: segue, sender: sender)  
    guard let segueIdentifier = segue.identifier else {  
        return  
    }  
}
```

```
switch segueIdentifier {  
  
    case "showDetailsVC":  
        guard let destinationVC = segue.destination as? DetailsViewController else {  
            fatalError("showDetalhe segue com destino de tipo inesperado")  
        }  
}
```



// 2. override

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    super.prepare(for: segue, sender: sender)  
  
    guard let segueIdentifier = segue.identifier else {  
        return  
    }  
  
    switch segueIdentifier {  
    case "showDetailsVC":  
        guard let destinationVC = segue.destination as? DetailsViewController else {  
            fatalError("showDetalhe segue com destino de tipo inesperado")  
        }  
  
        guard let selectedCell = sender as? PlayerTableViewCell else {  
            fatalError("célula selecionada é de tipo inesperado")  
        }  
  
        guard let selectedIndexPath = self.tableView.indexPath(for: selectedCell) else {  
            fatalError("célula selecionada não está sendo mostrada")  
        }  
    }  
}
```



// 2. override

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    super.prepare(for: segue, sender: sender)  
  
    guard let segueIdentifier = segue.identifier else {  
        return  
    }  
  
    switch segueIdentifier {  
    case "showDetailsVC":  
        guard let destinationVC = segue.destination as? DetailsViewController else {  
            fatalError("showDetalhe segue com destino de tipo inesperado")  
        }  
        guard let selectedCell = sender as? PlayerTableViewCell else {  
            fatalError("célula selecionada é de tipo inesperado")  
        }  
        guard let selectedIndexPath = self.tableView.indexPath(for: selectedCell) else {  
            fatalError("célula selecionada não está sendo mostrada")  
        }  
    }  
}
```

```
let selectedPlayer = self.players[selectedIndexPath.row]  
destinationVC.currentPlayer = selectedPlayer
```



// 2. override

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    super.prepare(for: segue, sender: sender)  
  
    guard let segueIdentifier = segue.identifier else {  
        return  
    }  
  
    switch segueIdentifier {  
    case "showDetailsVC":  
        guard let destinationVC = segue.destination as? DetailsViewController else {  
            fatalError("showDetalhe segue com destino de tipo inesperado")  
        }  
        guard let selectedCell = sender as? PlayerTableViewCell else {  
            fatalError("célula selecionada é de tipo inesperado")  
        }  
        guard let selectedIndexPath = self.tableView.indexPath(for: selectedCell) else {  
            fatalError("célula selecionada não está sendo mostrada")  
        }  
        let selectedPlayer = self.players[selectedIndexPath.row]  
        destinationVC.currentPlayer = selectedPlayer  
  
    default:  
        fatalError("Identificador da segue desconhecido")  
    }  
}
```



## // 2. override

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
  
    super.prepare(for: segue, sender: sender)  
  
    guard let segueIdentifier = segue.identifier else {  
        return  
    }  
  
    switch segueIdentifier {  
    case "showDetailsVC":  
        guard let destinationVC = segue.destination as? DetailsViewController else {  
            fatalError("showDetalhe segue com destino de tipo inesperado")  
        }  
        guard let selectedCell = sender as? PlayerTableViewCell else {  
            fatalError("célula selecionada é de tipo inesperado")  
        }  
        guard let selectedIndexPath = self.tableView.indexPath(for: selectedCell) else {  
            fatalError("célula selecionada não está sendo mostrada")  
        }  
        let selectedPlayer = self.players[selectedIndexPath.row]  
        destinationVC.currentPlayer = selectedPlayer  
    default:  
        fatalError("Identificador da segue desconhecido")  
    }  
}
```



### // 3. atualizar interface

```
class DetailsViewController: UIViewController {
    var currentPlayer: Player?

    @IBOutlet weak var numberLabel: UILabel!
    @IBOutlet weak var positionLabel: UILabel!
    @IBOutlet weak var playerImage: UIImageView!
    @IBOutlet weak var nameLabel: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()




        if let currentPlayer = self.currentPlayer {
            nameLabel.text = currentPlayer.nome
            playerImage.image = currentPlayer.foto
            positionLabel.text = currentPlayer.posicao
            numberLabel.text = String(describing: currentPlayer.numero)
        }
    }
}
```



// resultado

Carlini 12:25 PM

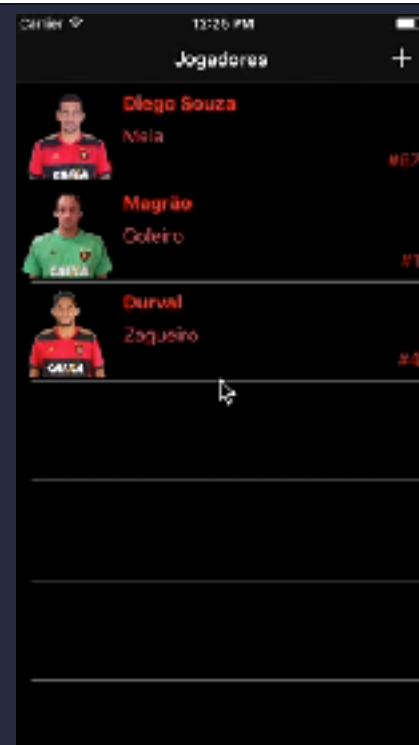
Jogadores +

	<b>Diego Souza</b> Meia	#87
	<b>Magrão</b> Goleiro	#1
	<b>Durval</b> Zagueiro	#4




**cit**



// resultado



The screenshot shows a mobile application interface with a dark background. At the top, there is a status bar with 'Carier', a signal strength icon, the time '12:25 PM', and a battery icon. Below this is a header bar with the title 'Jogadores' and a plus sign icon. The main content area displays a list of three players, each with a small profile picture, their name, position, and number. The first player is Diego Souza, a Midfielder (Meia) with number 87. The second player is Magrão, a Forward (Atacante) with number 11. The third player is Durval, a Forward (Atacante) with number 4. Below the list, there are four empty rows with horizontal lines, suggesting a scrollable list. A mouse cursor is visible over the third player's row.

	<b>Diego Souza</b> Meia	#87
	<b>Magrão</b> Atacante	#11
	<b>Durval</b> Atacante	#4

**cit**

/\*

Quais são as constraints  
do zoom?

\*/



// algoritmo

1. adicionar **Scroll View**

1.1. fixar os **4 lados** da Scroll View em relação à superview

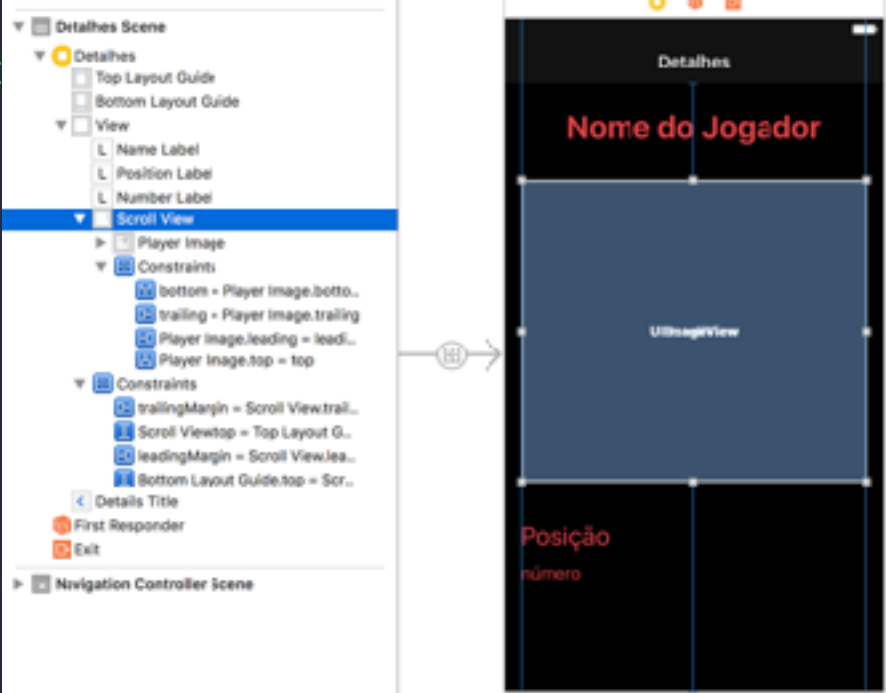
2. adicionar **Image View**

2.1. fixar os **4 lados** da Image View em relação à Scroll View

2.2. fixar **altura** e **largura**



```
// constraint
```






// constraints

iPhone 7 - iOS 10.3 (14E301)

Carrier: 32 10:43 PM

Jogadoras

	<b>Diego Souza</b> Mala	#57
	<b>Magrão</b> Goleiro	#1
	<b>Durval</b> Zagueiro	#4




*cit*

// constraints

iPhone 7 - iOS 10.3 (14E301)

Carrier: 32 10:43 PM

Jogadoras

	<b>Diego Souza</b> Mala	#57
	<b>Magrão</b> Goleiro	#5
	<b>Durval</b> Zagueiro	#6

*cit*

// iOS do princípio com Swift: Básico de Auto Layout

<https://code.tutsplus.com/pt/tutorials/ios-from-scratch-with-swift-auto-layout-basics--cms-25520>



// Table View

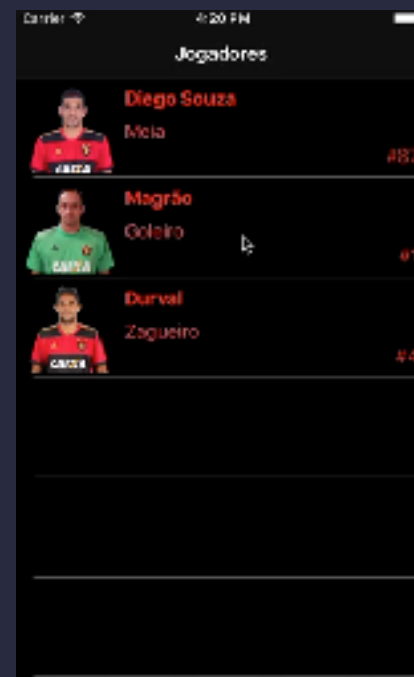







# // Edições



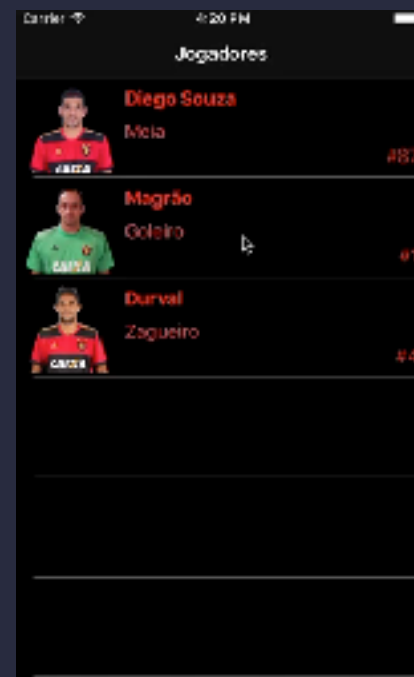
// remoção






The screenshot shows a mobile application interface with a dark background. At the top, there's a status bar with 'Carrier', '4:20 PM', and a battery icon. Below that, the title 'Jogadores' is centered. The list contains three items, each with a small player photo on the left, the player's name in red text, their position in white text, and a red number on the right. A mouse cursor is hovering over the second item.

Jogadores		
	<b>Diego Souza</b> Meia	487
	<b>Magrão</b> Goleiro	41
	<b>Durval</b> Zagueiro	44

// remoção



The screenshot shows a mobile application interface with a dark background. At the top, there is a status bar with 'Carrier', '4:20 PM', and a battery icon. Below this is a header 'Jogadores'. The main content is a list of three players, each with a small profile picture, their name, position, and a numerical value. The first player is Diego Souza (Meia) with value 487. The second player is Magrão (Goleiro) with value 41. The third player is Durval (Zagueiro) with value 44. The list is followed by three empty rows.

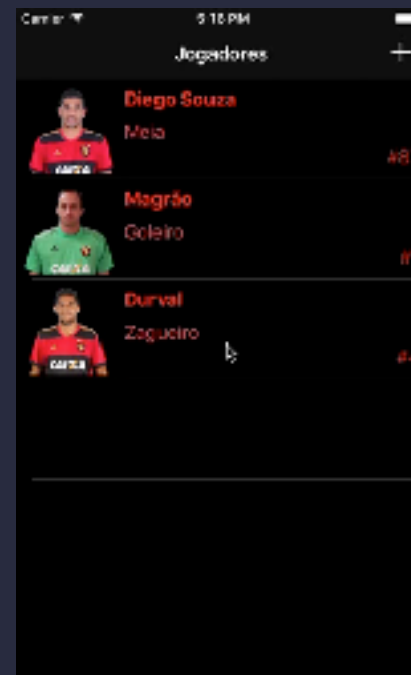
Jogadores			
	Diego Souza	Meia	487
	Magrão	Goleiro	41
	Durval	Zagueiro	44

// remoção

```
override func tableView(_ tableView: UITableView,  
    commit editingStyle: UITableViewCellEditingStyle,  
    forRowAt indexPath: IndexPath) {  
  
    if editingStyle == .delete {  
        // Delete the row from the data source  
        self.players.remove(at: indexPath.row)  
        tableView.deleteRows(at: [indexPath], with: .fade)  
    }  
}
```



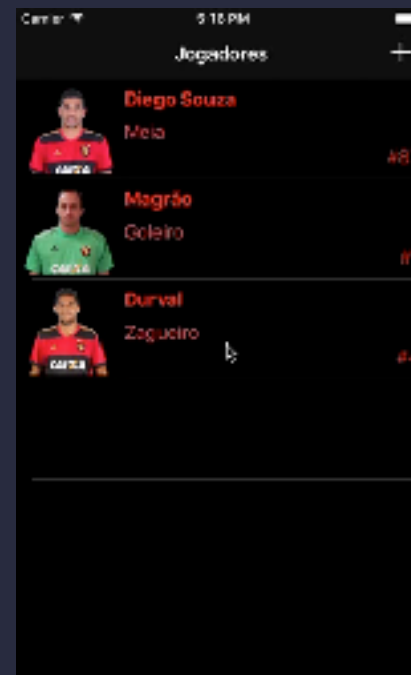
// inserção






The screenshot shows a mobile application interface with a dark background. At the top, there is a status bar with 'Carro' and a signal icon on the left, '9:10 PM' in the center, and a battery icon on the right. Below the status bar is a header with the text 'Jogadores' and a plus icon on the right. The main content area displays a list of three players, each with a small profile picture, their name, position, and a number. The players are: Diego Souza (Meia, #87), Magrão (Goleiro, #1), and Durval (Zagueiro, #4). A mouse cursor is hovering over the 'Zagueiro' position of Durval. Below the list, there is a horizontal line and a large empty space.

Player	Position	Number
Diego Souza	Meia	#87
Magrão	Goleiro	#1
Durval	Zagueiro	#4

// inserção



The screenshot shows a mobile application interface with a dark background. At the top, there is a status bar with 'Carro' and a signal icon on the left, '9:10 PM' in the center, and a battery icon on the right. Below the status bar is a header with the text 'Jogadores' and a plus icon on the right. The main content is a list of three players, each with a small profile picture on the left, their name in bold red text, their position in regular white text, and a number in red text on the right. The players are: Diego Souza (Meia, #87), Magrão (Goleiro, #1), and Durval (Zagueiro, #4). A mouse cursor is pointing at the 'Zagueiro' position of Durval. Below the list is a horizontal line, and the bottom of the screen is empty.

Jogadores		
	<b>Diego Souza</b> Meia	#87
	<b>Magrão</b> Goleiro	#1
	<b>Durval</b> Zagueiro	#4

// inserção

```
@IBAction func addPlayer(_ sender: Any) {  
    let newPath = IndexPath(row: players.count, section: 0)  
  
    let andre = Player(nome: "André",  
                        posicao: "Atacante",  
                        numero: 90,  
                        foto: #imageLiteral(resourceName: "andre"))  
  
    players.append(andre)  
  
    self.tableView.insertRows(at: [newIndexPath], with: .automatic)  
}
```

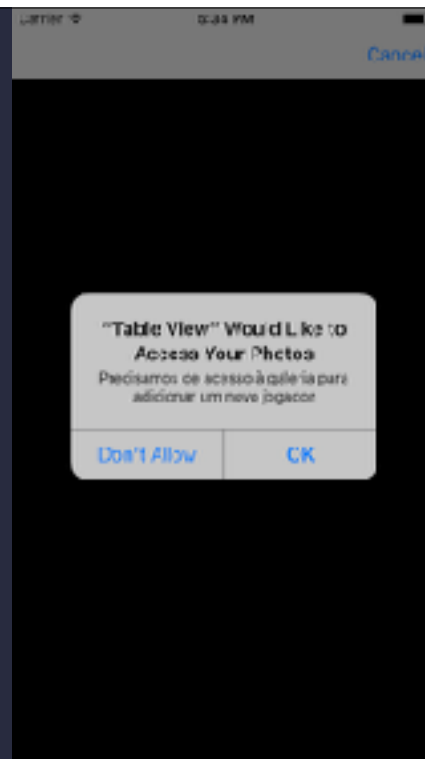


// Accesso à galleria





// permissão



*cit*

// info.plist

Key	Type	Value
Information Property List	Dictionary	(15 items)
Localization native development region	String	en
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Status bar is initially hidden	Boolean	NO
Privacy - Photo Library Usage Description	String	Precisamos de acesso à galeria para adicionar um novo jogador
Supported interface orientations	Array	(3 items)

// abrindo galeria

```
if UIImagePickerController.isSourceTypeAvailable(UIImagePickerControllerSourceType.photoLibrary) {  
    let imagePicker = UIImagePickerController()  
  
    imagePicker.delegate = self  
  
    imagePicker.sourceType = UIImagePickerControllerSourceType.photoLibrary  
  
    imagePicker.allowsEditing = true  
    self.present(imagePicker, animated: true, completion: nil)  
}
```



```
// capturando imagem selecionada
extension PlayerTableViewController: UIImagePickerControllerDelegate ,
    UINavigationControllerDelegate {

    func imagePickerController(_ picker: UIImagePickerController,
        didFinishPickingMediaWithInfo info: [String : Any]) {

        if let pickedImage = info[UIImagePickerControllerOriginalImage] as? UIImage {

            let imagemSelecionada = pickedImage
            // adicionar na tabela
        }
        picker.dismiss(animated: true, completion: nil)
    }
}
```



# // Exercício



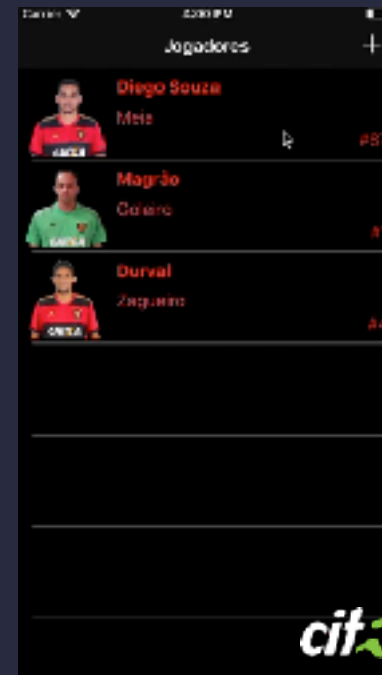
## // Exercício 14

### Lista de coisas II

1. ponha um botão de **adicionar**, que leva o usuário para uma tela onde ele escreverá um **nome**, e uma **foto** para um novo item a ser **inserido** na tabela.
2. permita que o usuário possa **deletar** um item da tabela

## // Extra

3. permita que o usuário possa **editar** um item da tabela



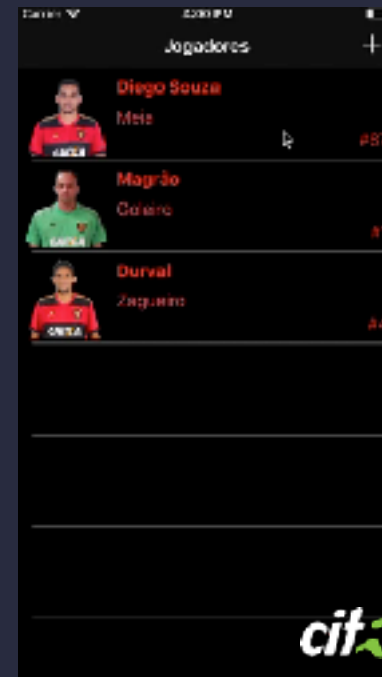
## // Exercício 14

### Lista de coisas II

1. ponha um botão de **adicionar**, que leva o usuário para uma tela onde ele escreverá um **nome**, e uma **foto** para um novo item a ser **inserido** na tabela tabela.
2. permita que o usuário possa **deletar** um item da tabela

## // Extra

3. permita que o usuário possa **editar** um item da tabela



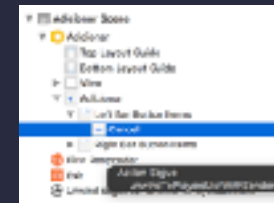
// unwind





```
// unwind
```

1. fazer método no view controller de destino
  1. `@IBAction func nomedafunc(sender: UIStoryboardSegue) {}`
2. No storyboard ligar (segurando control) a origem ao **Exit**
2. Selecionar nomedafunc



# DÚVIDAS



*cit*