

# CESAR Summer School - iOS

## TableView



# Conceito TableView e CollectionView

---

Implementar TableView

---

Popular Tabela

---

Remover Objeto

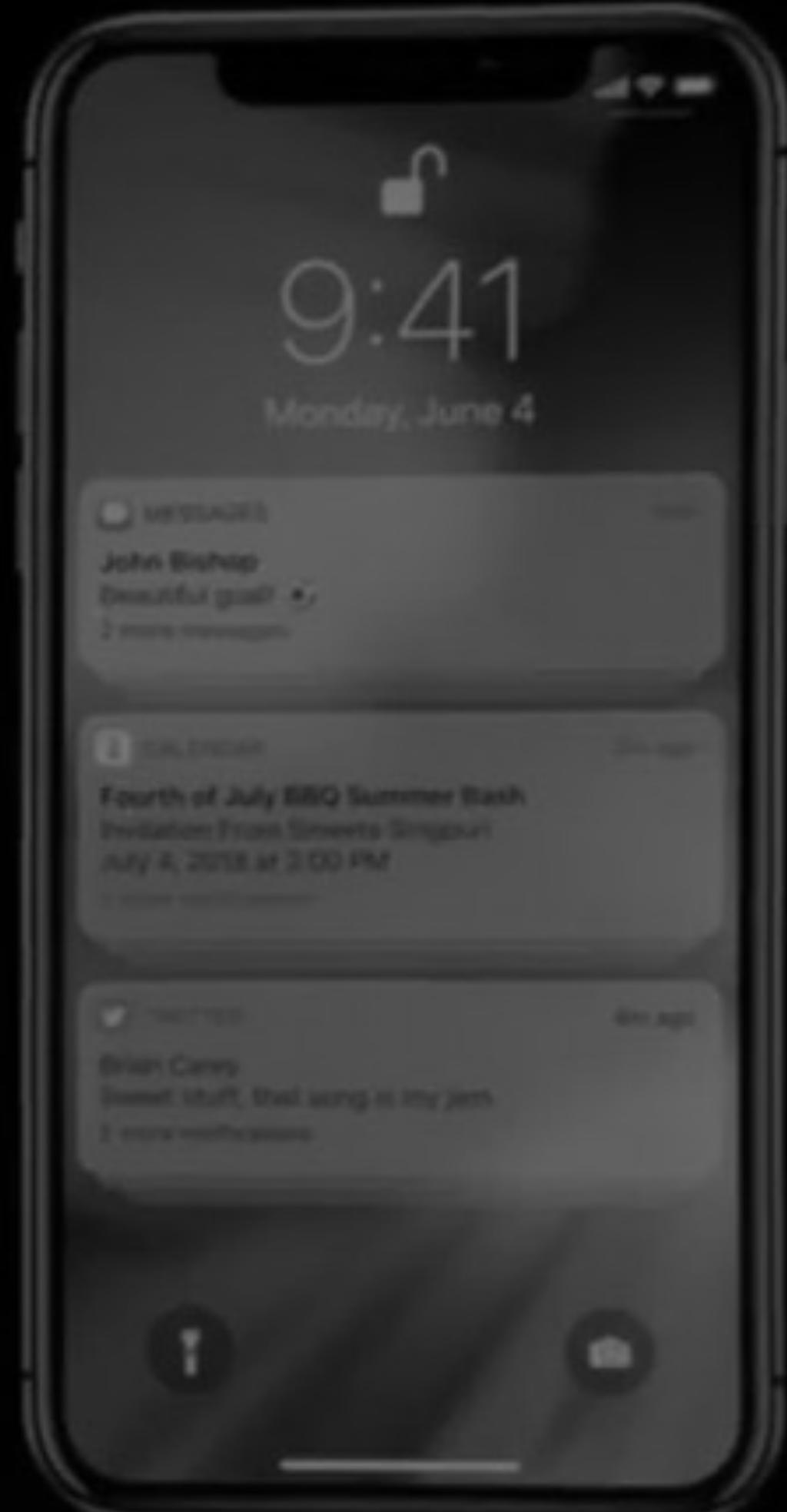
---

Inserir Objeto

---

Acesso a galeria

---

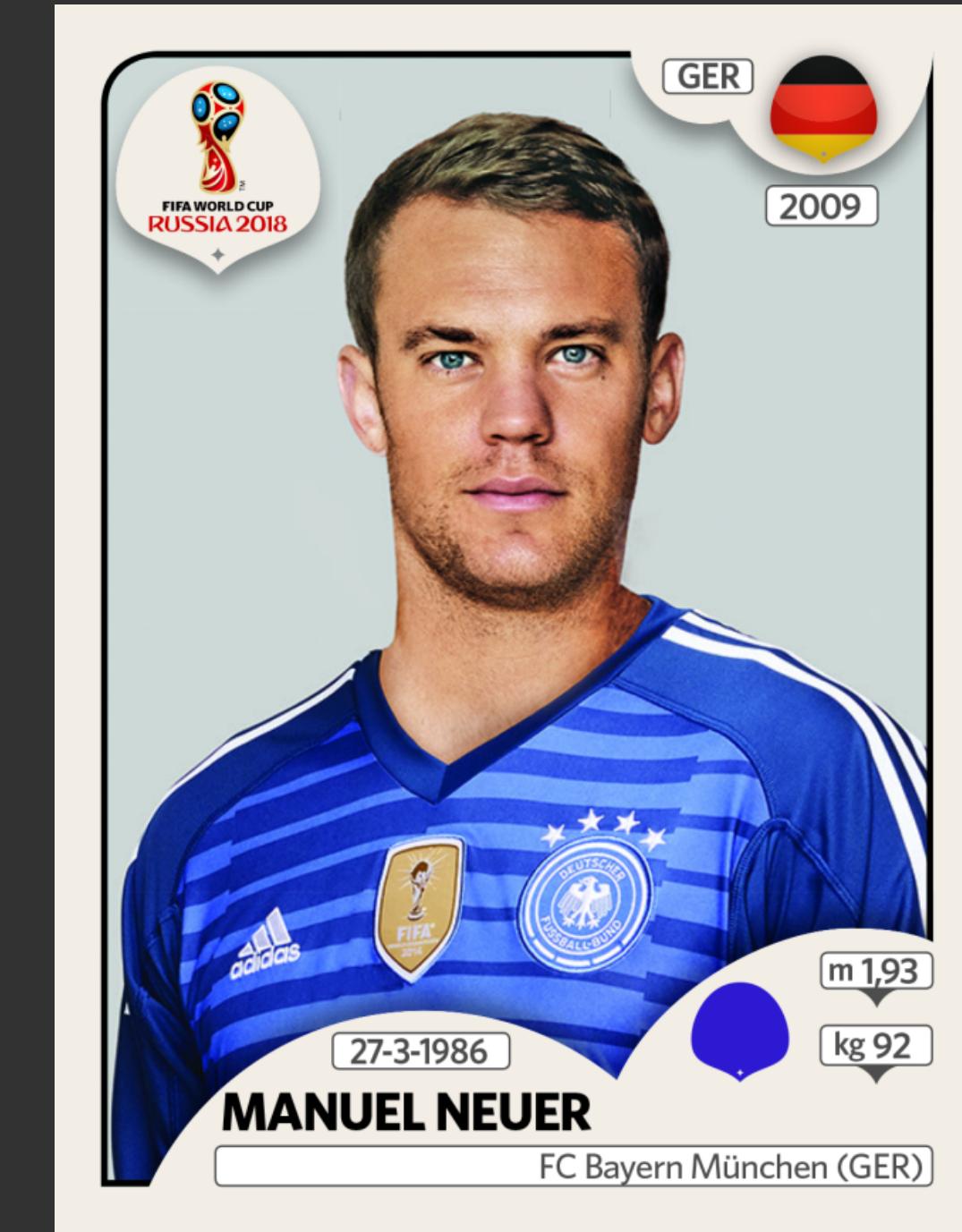
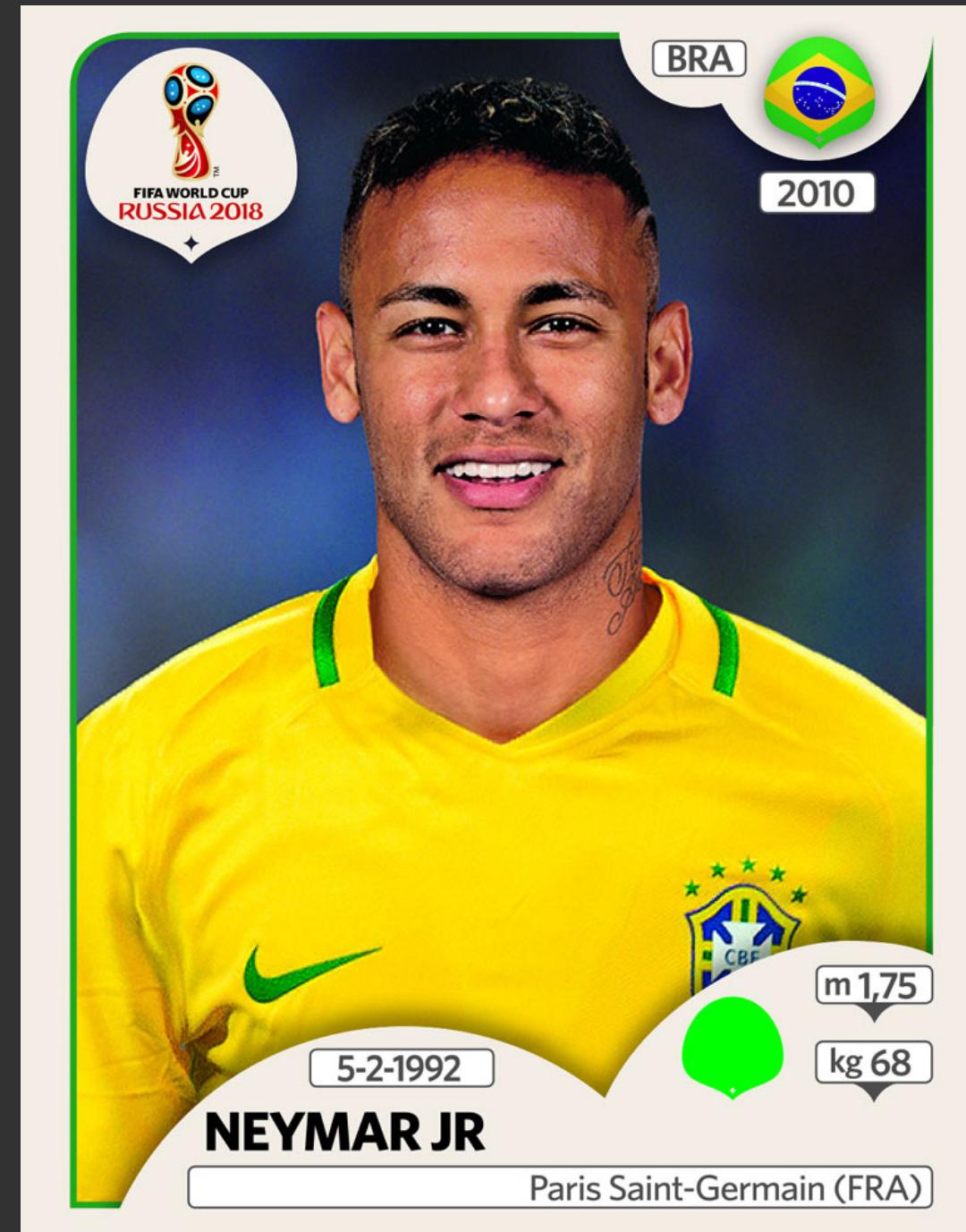


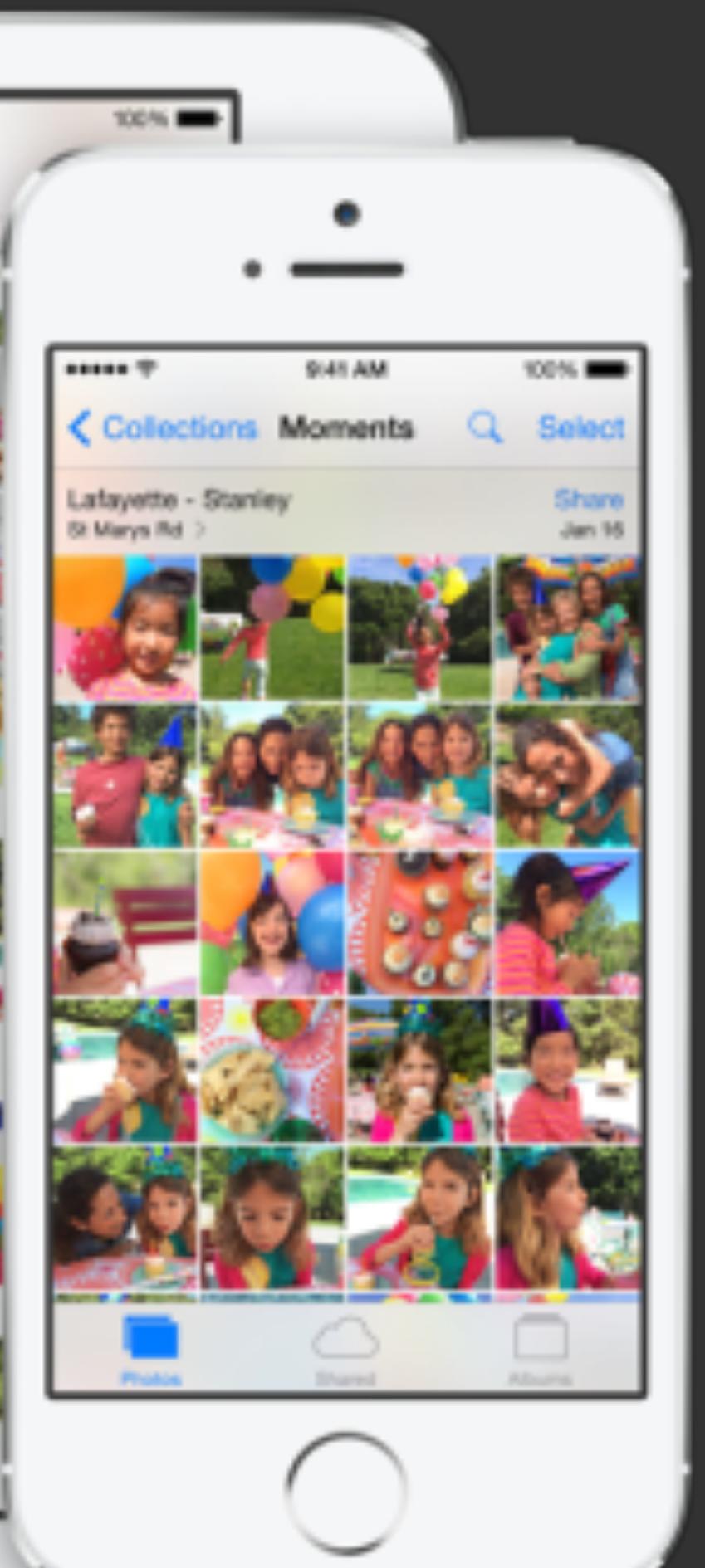
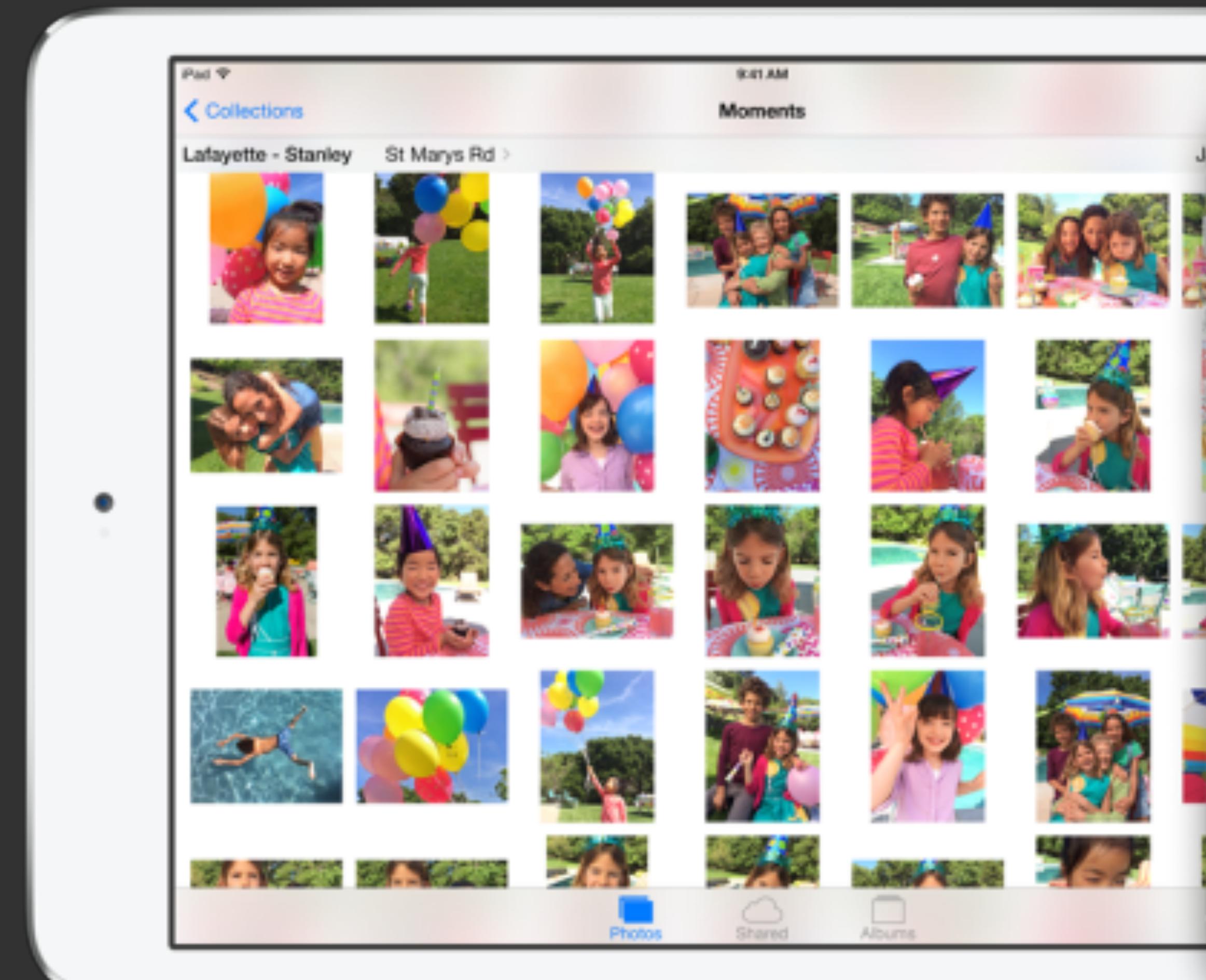
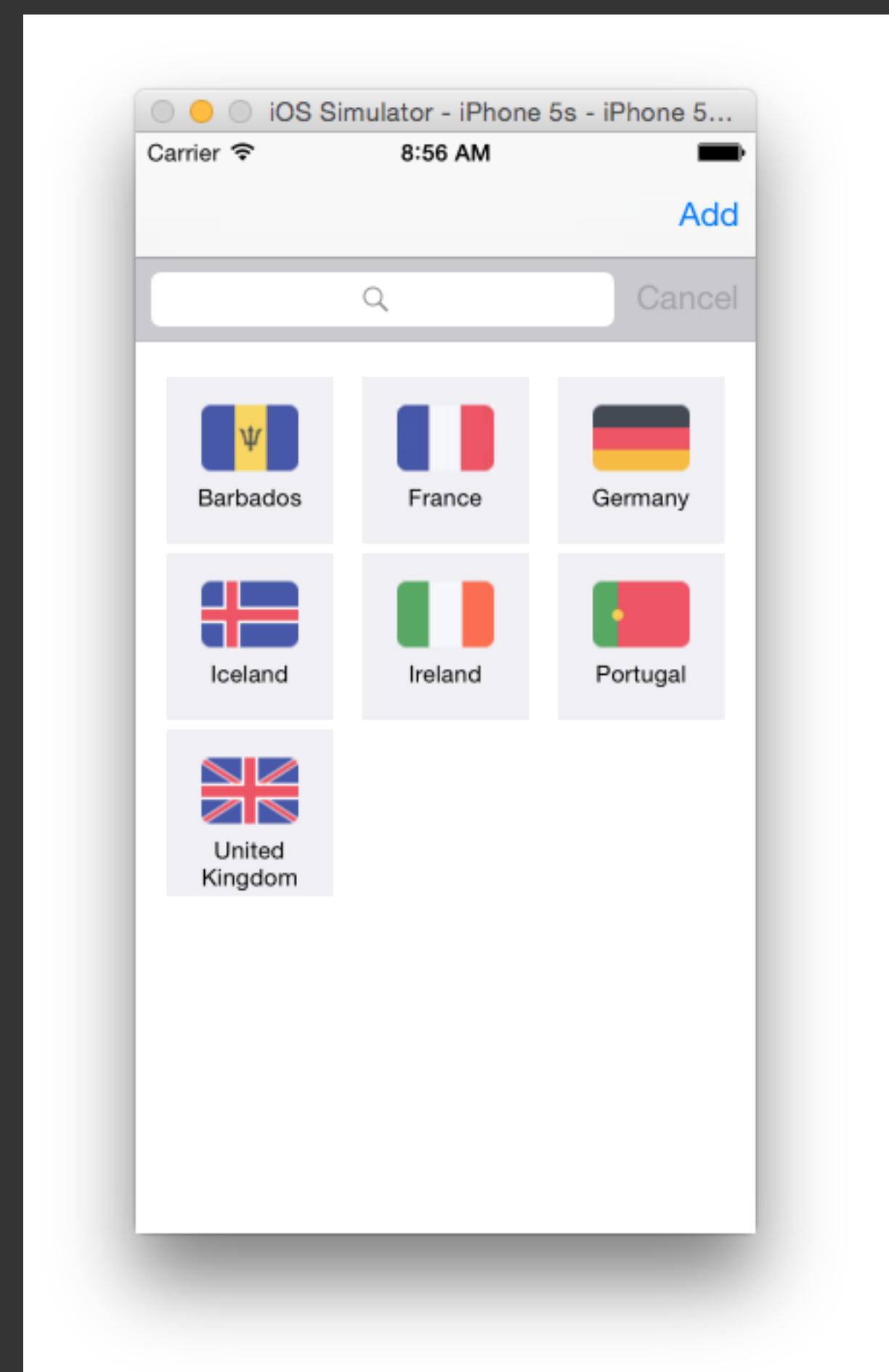
# Conceito



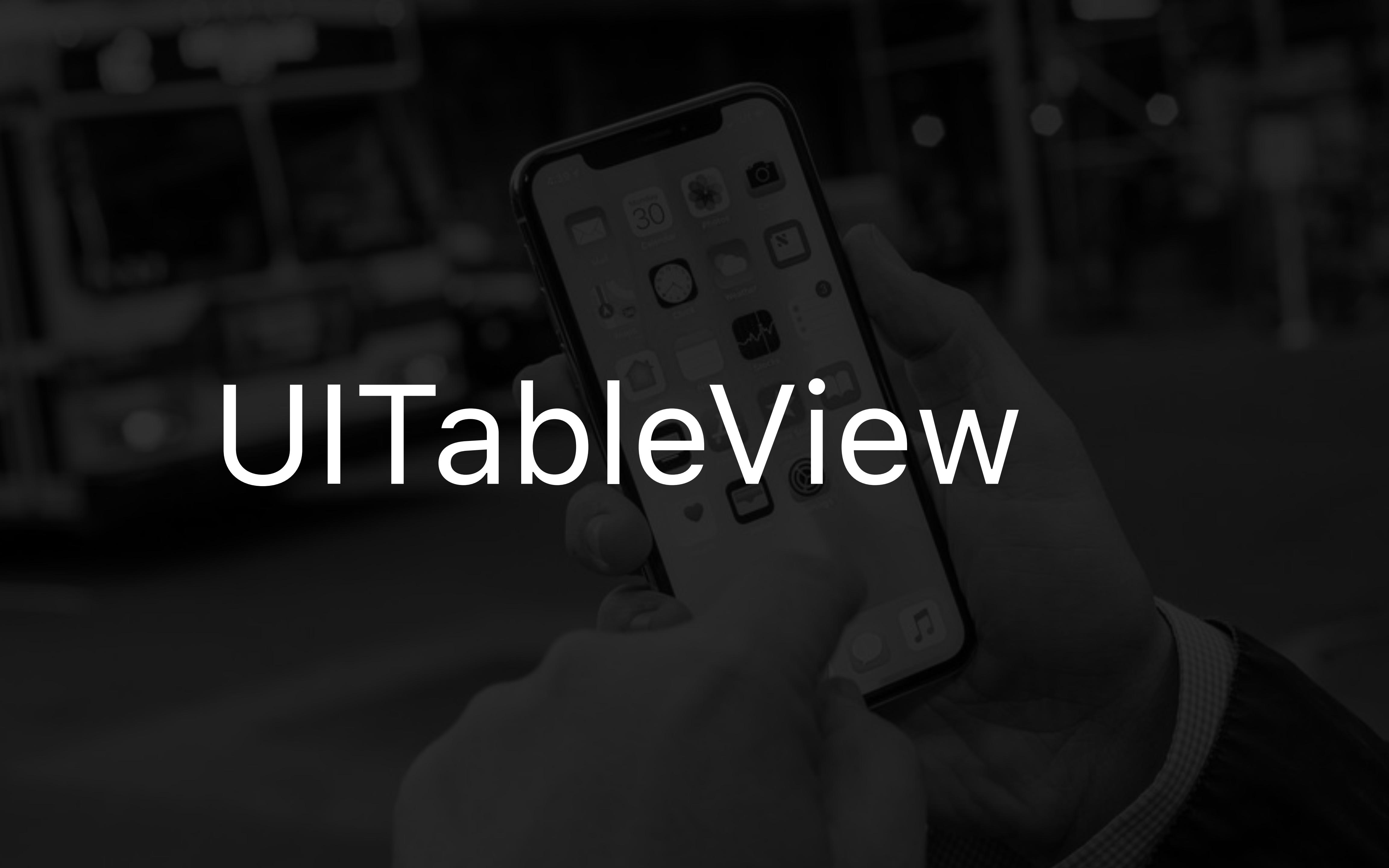
# UICollectionView





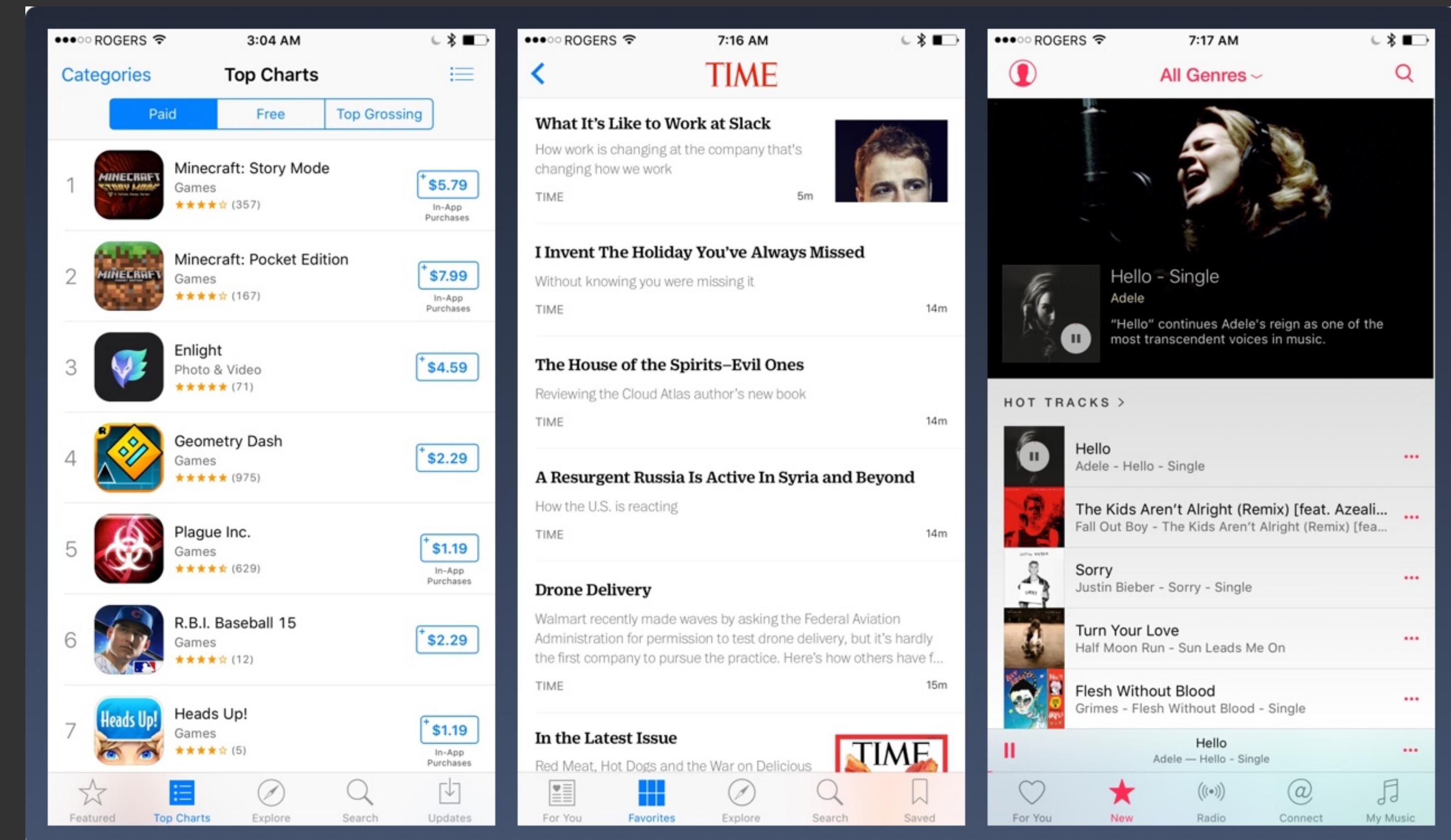


# UITableView



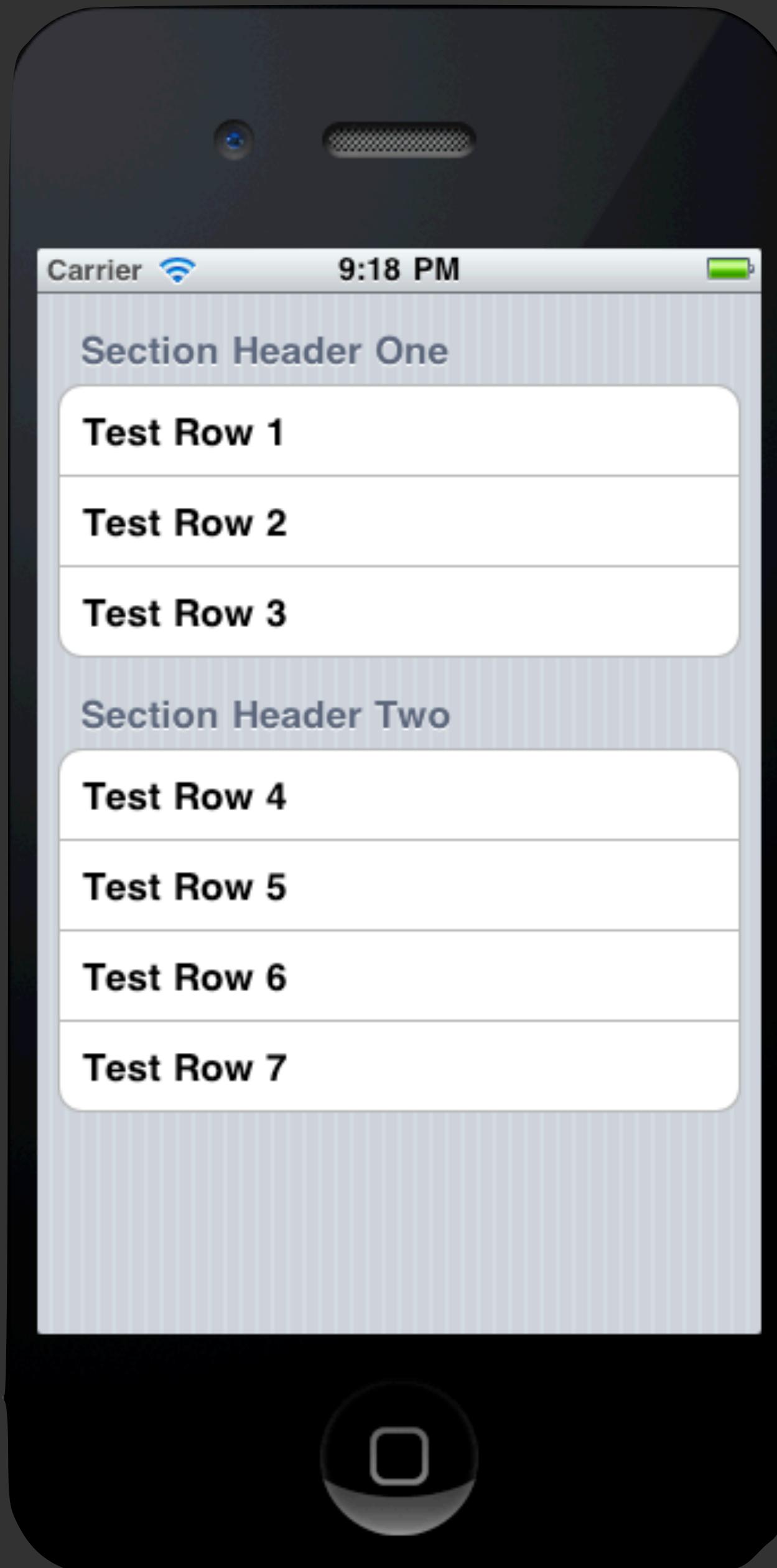
	Equipe	Pts
1	 Brasil	7
2	 Suíça	5
3	 Sérvia	3
4	 Costa Rica	1

- Conteúdo em lista
- Customizável
- Subclasse de UIScrollView

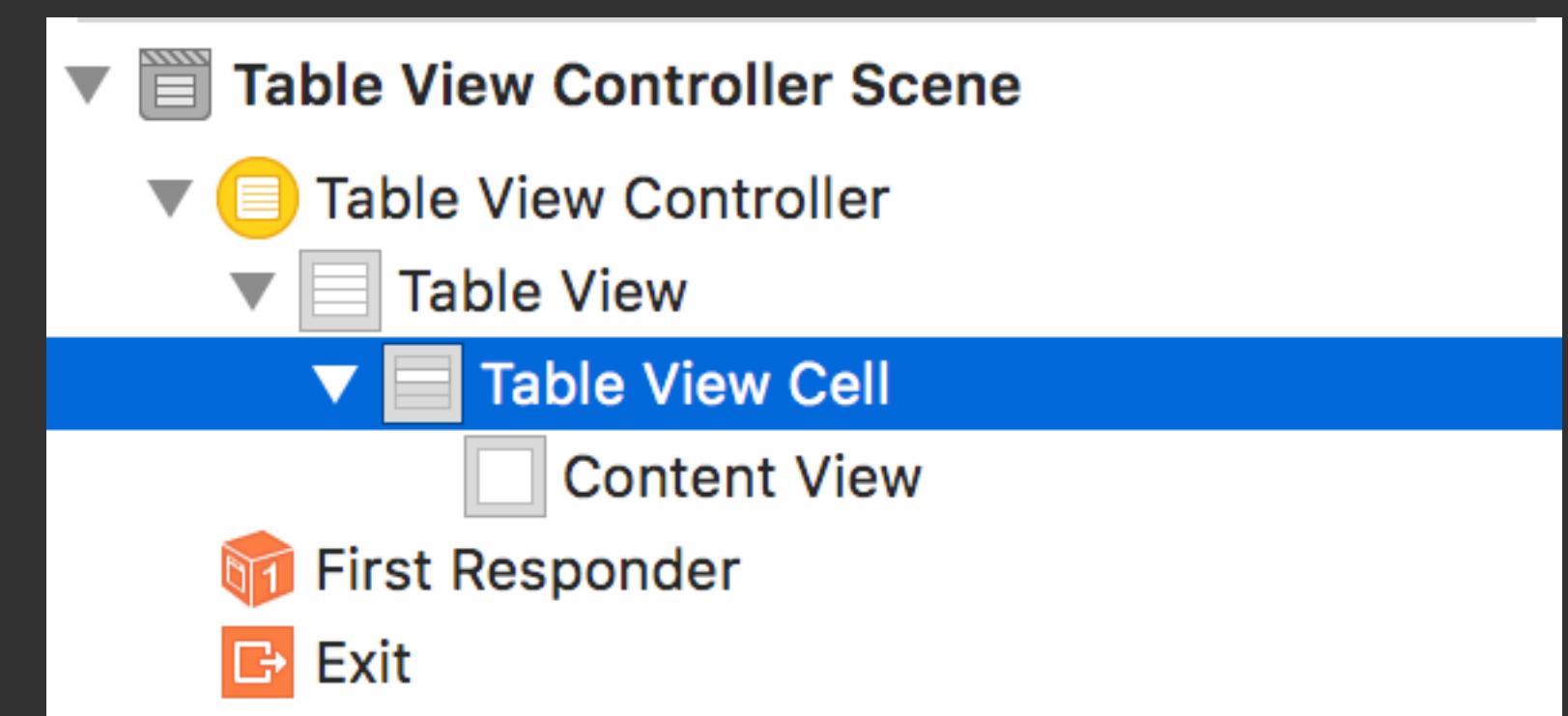
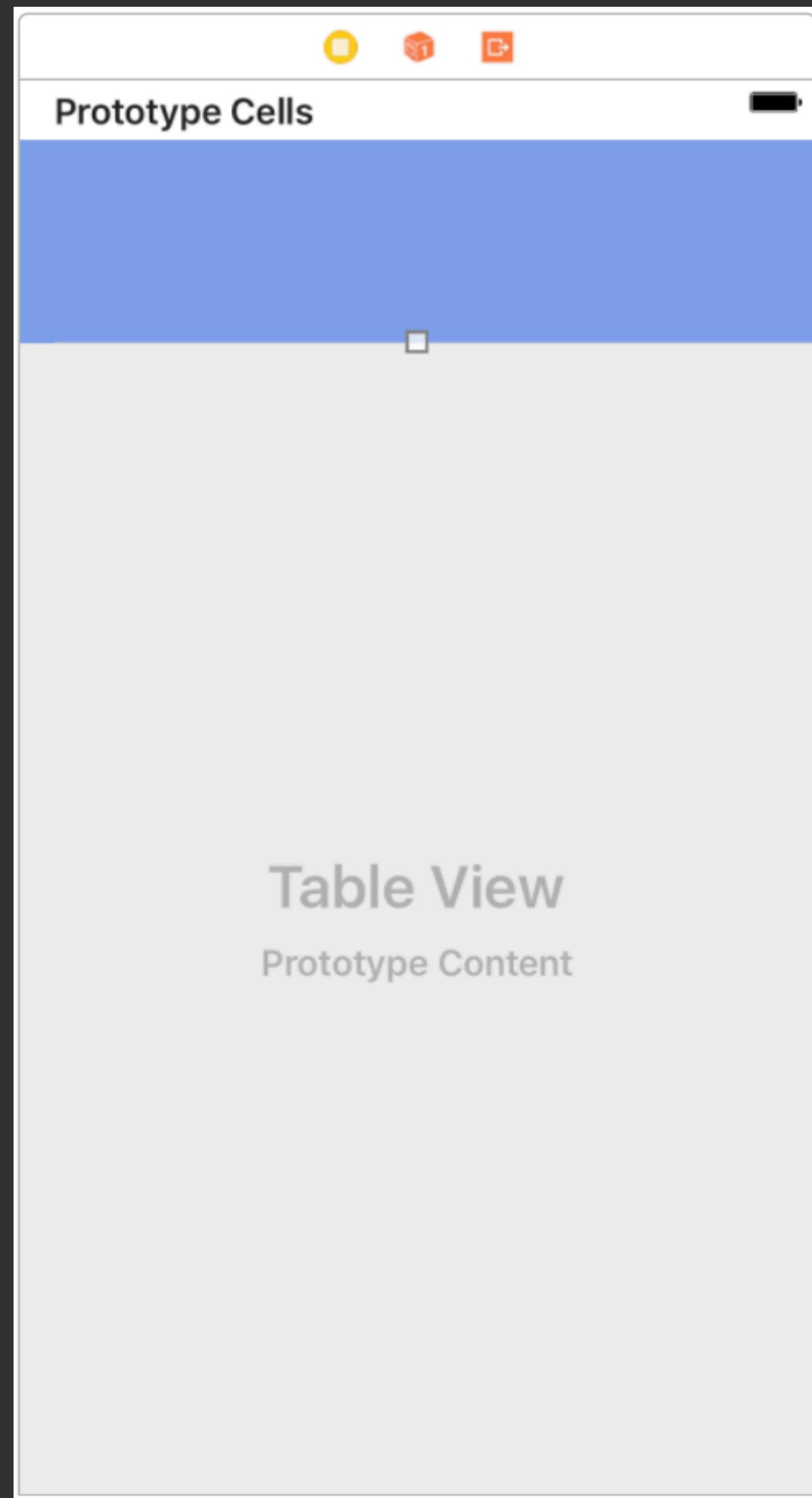
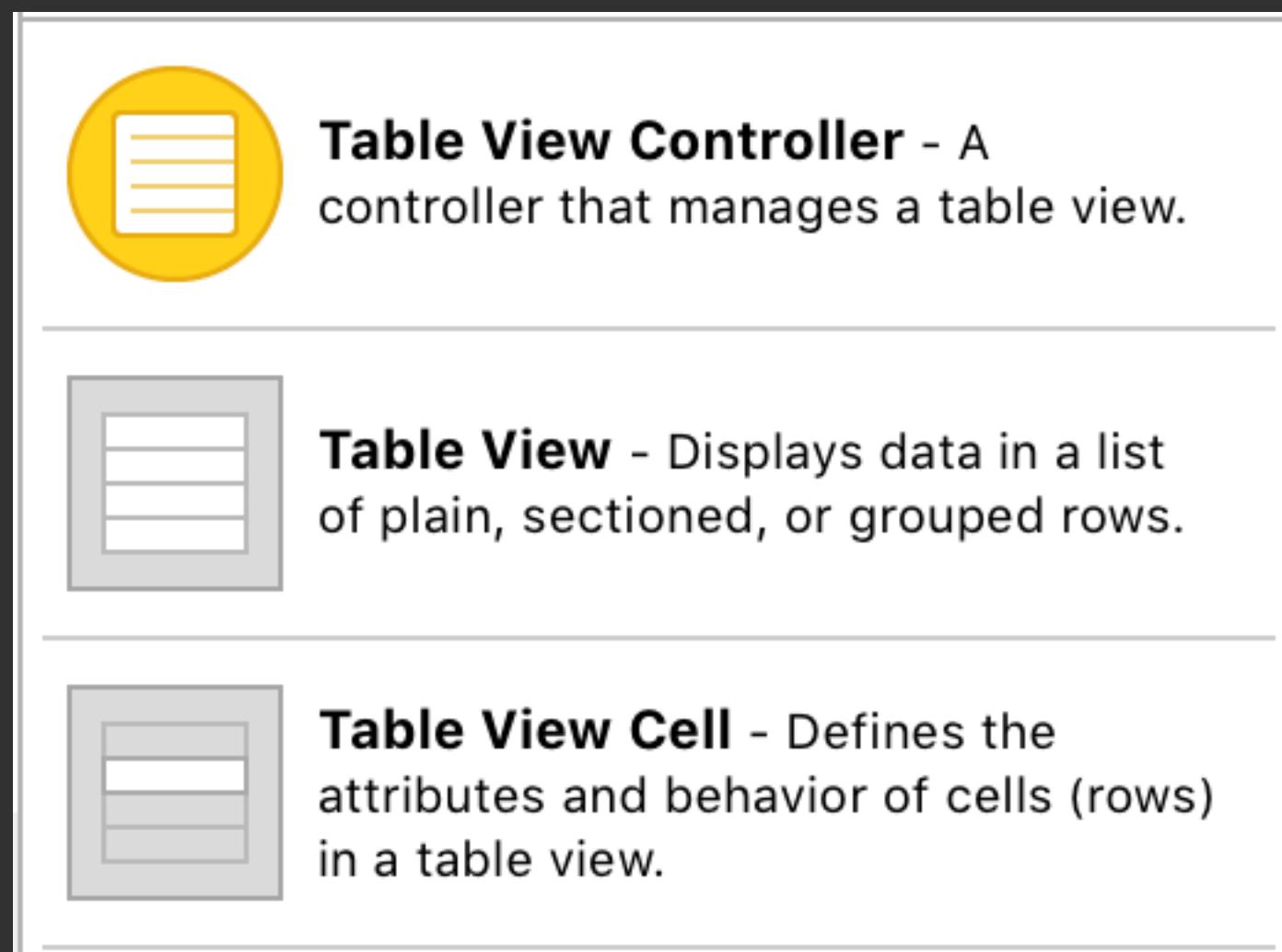


# // anatomia

- Seções
- Linhas
- Células



# // storyBoard



# // anatomia

- TableView + CollectionView



# Implementação



# Pegue imagens...

## NÃO DEMOREM NESSA ETAPA



Crie um novo projeto  
e coloque as imagens

# Crie a classe Jogador

```
import UIKit

class Jogador{

    var nome: String
    var selecao: String
    var image: UIImage
    var numero: Int

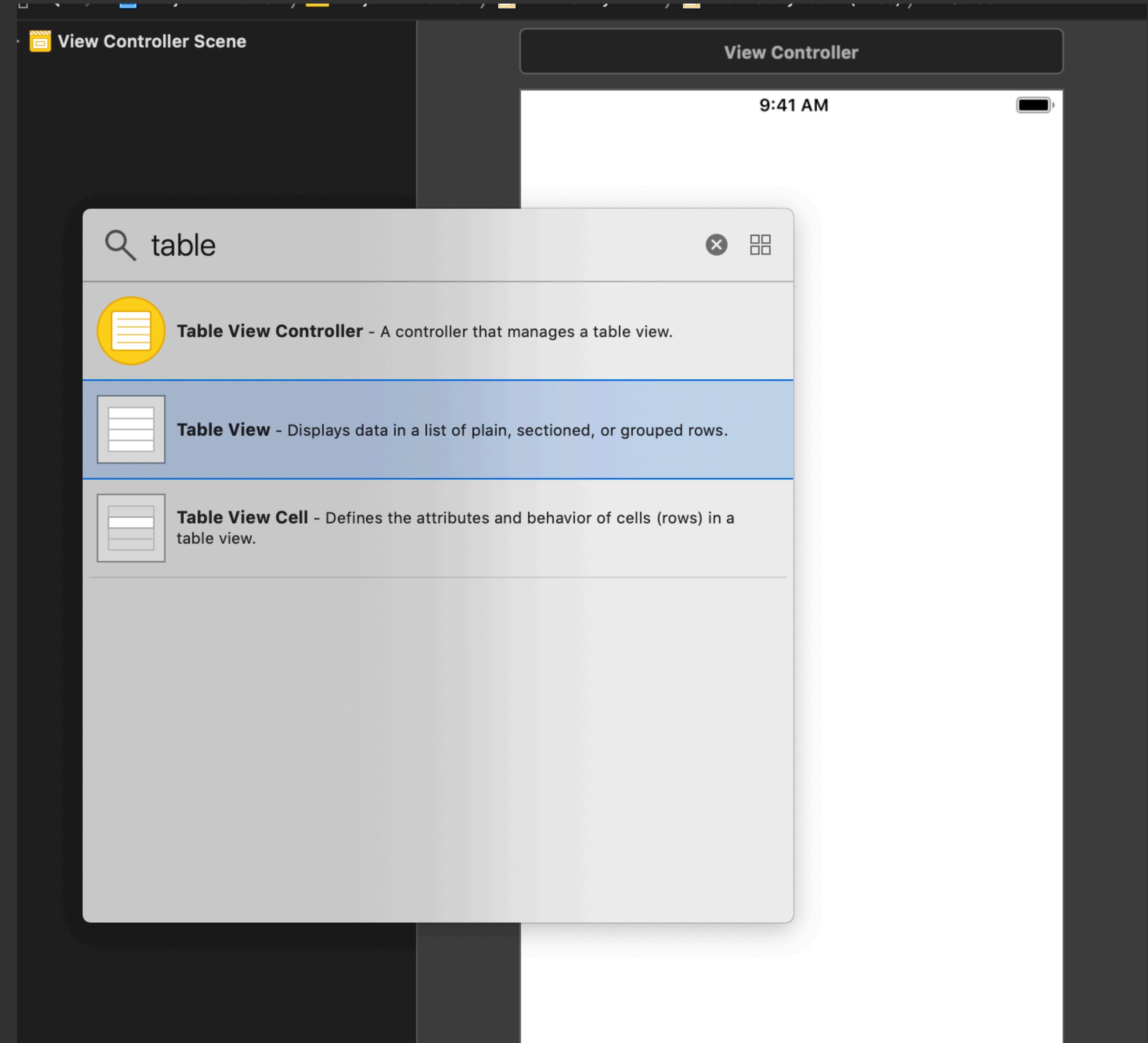
    init(nome: String, selecao: String,
         image: UIImage, numero: Int) {

        self.nome = nome
        self.selecao = selecao
        self.image = image
        self.numero = numero
    }

}
```



# Crie uma TableView no storyboard



# Agora crie uma TableViewCell na TableView

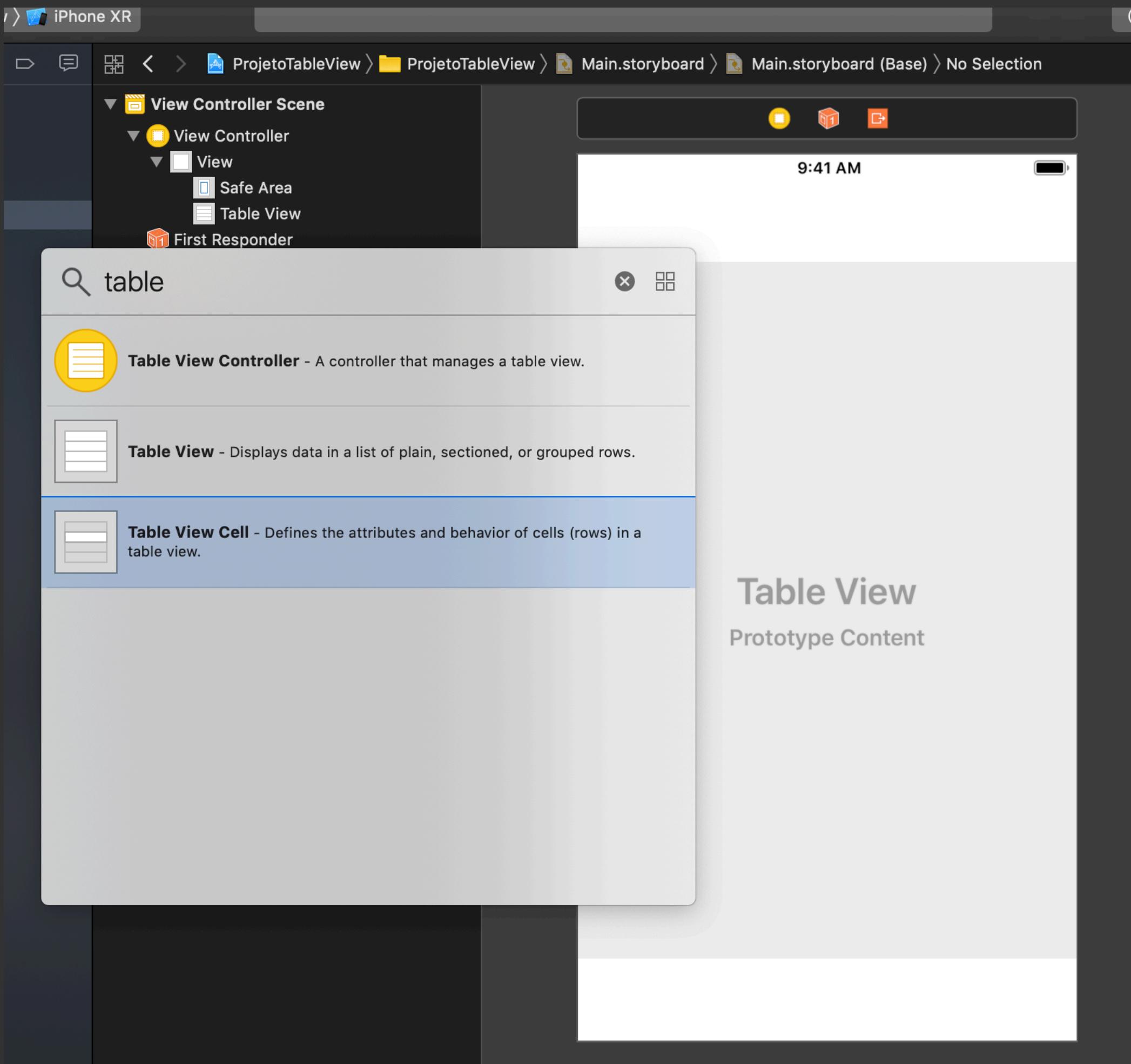
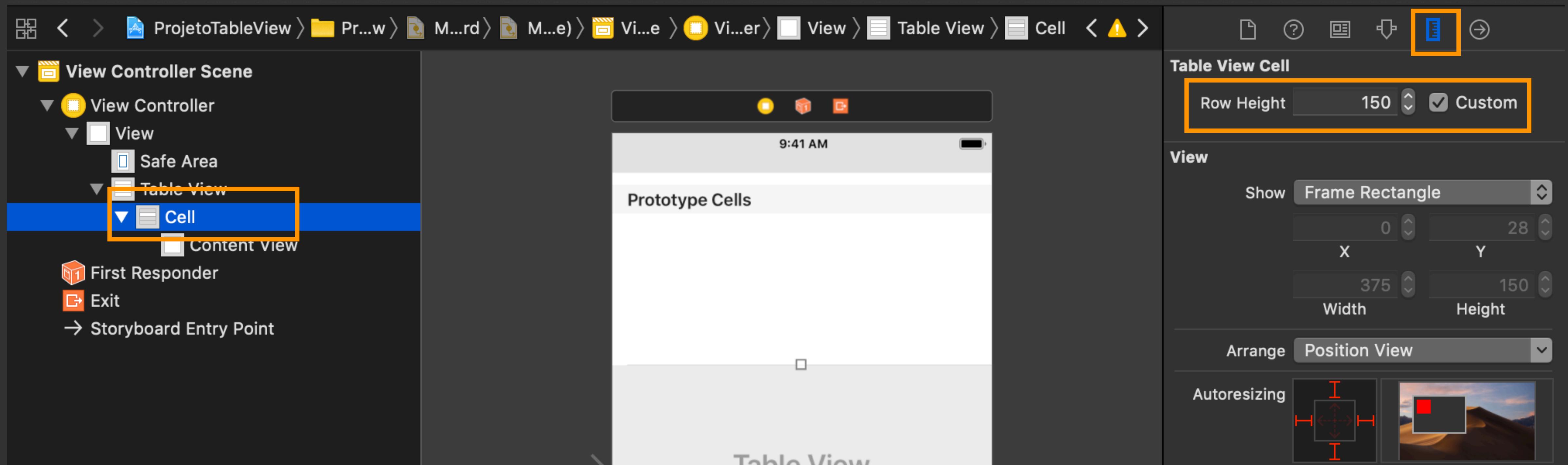


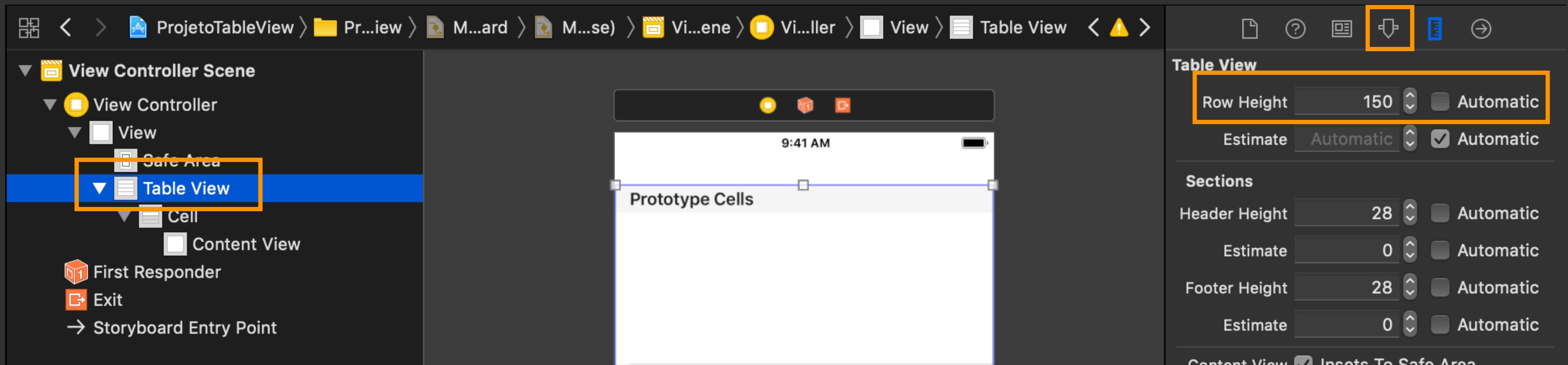
Table View  
Prototype Content

# Mudando altura da cell



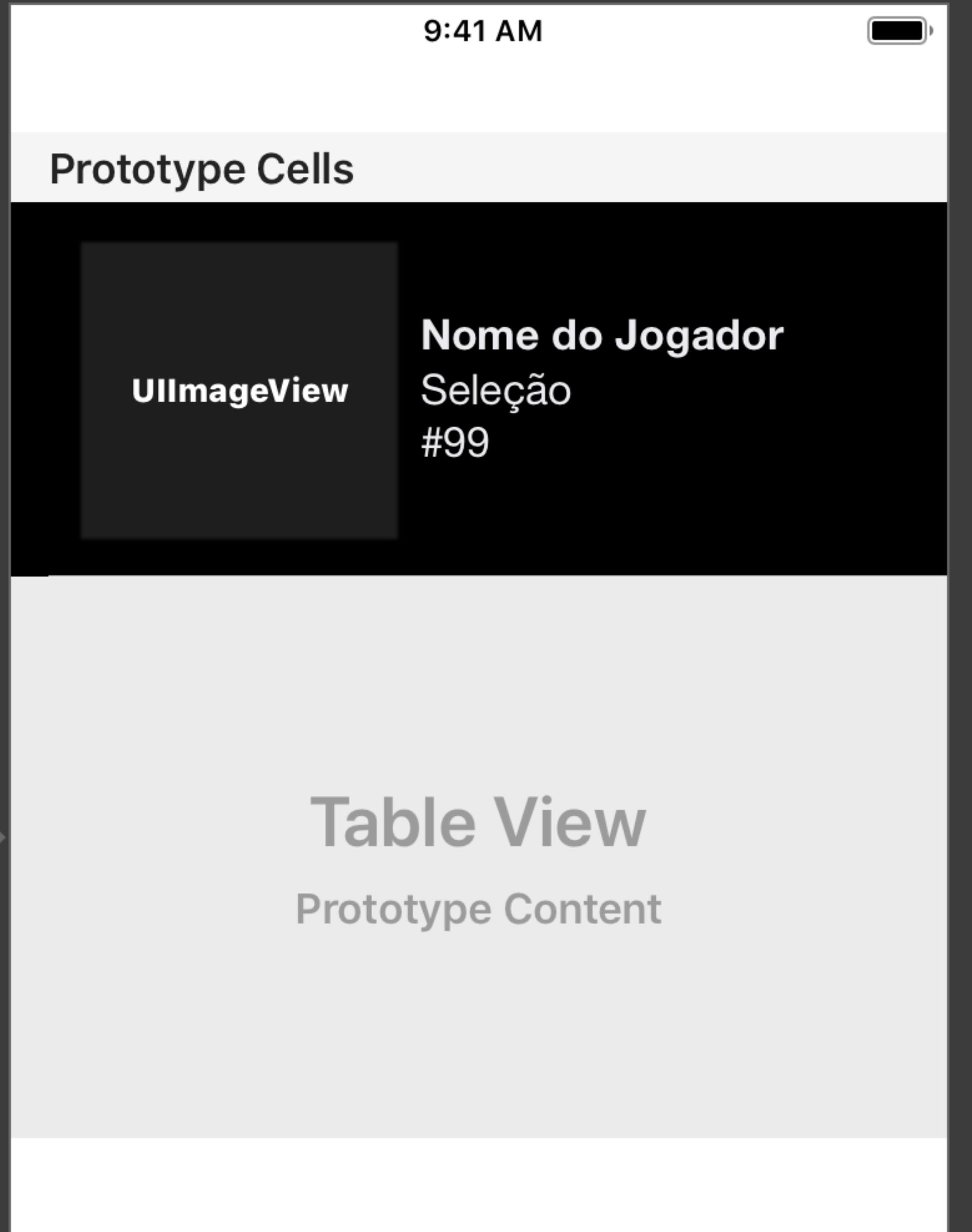
1 - Mude na TableViewCell

# Mudando altura da cell



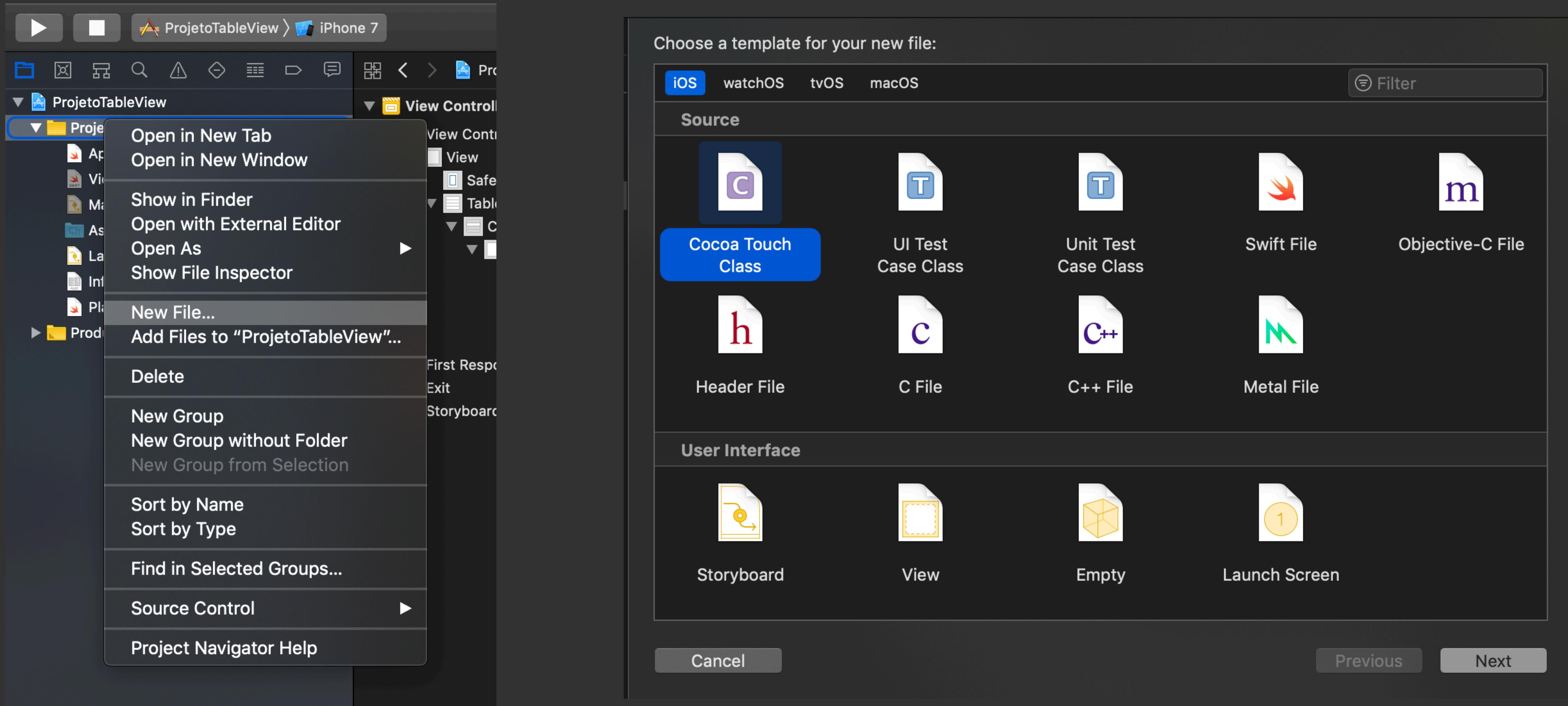
2 - Mude na TableView

# Personaliza sua cel



# UITableViewCell





```
import UIKit

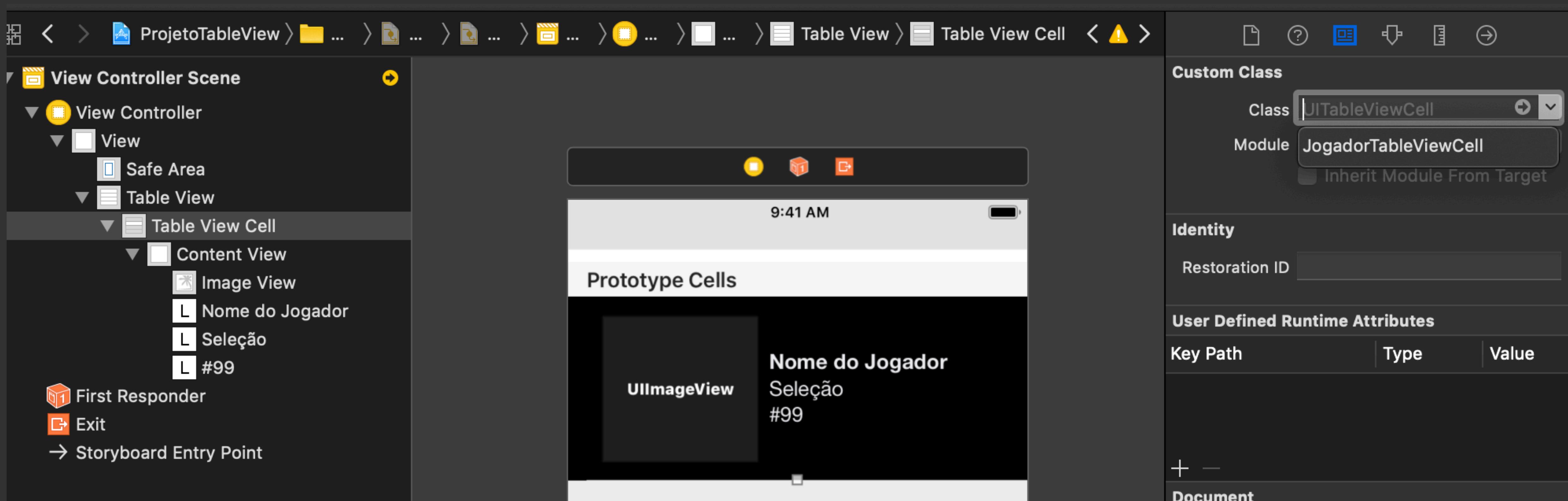
class JogadorTableViewCell: UITableViewCell {

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

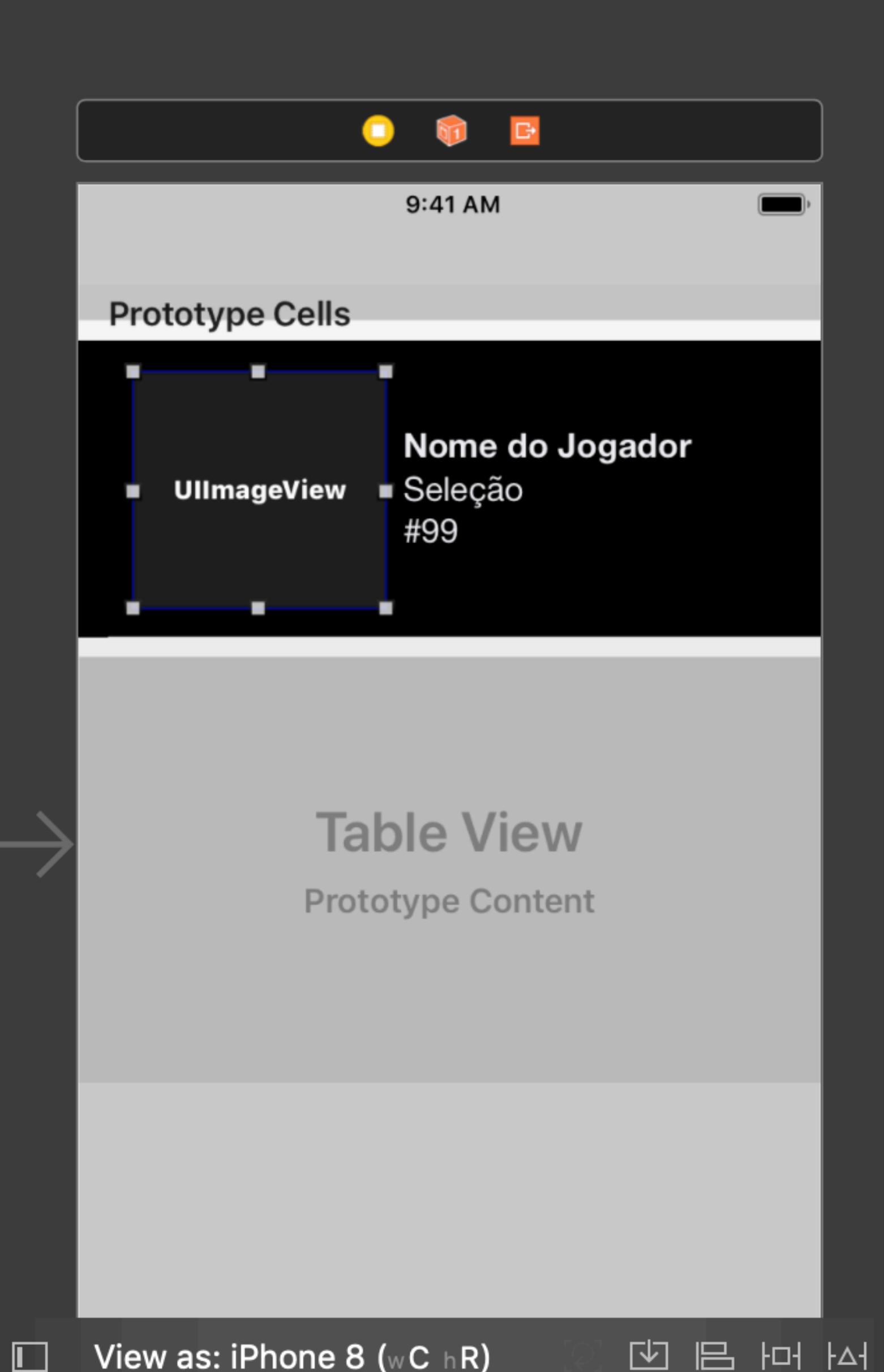
    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }
}
```

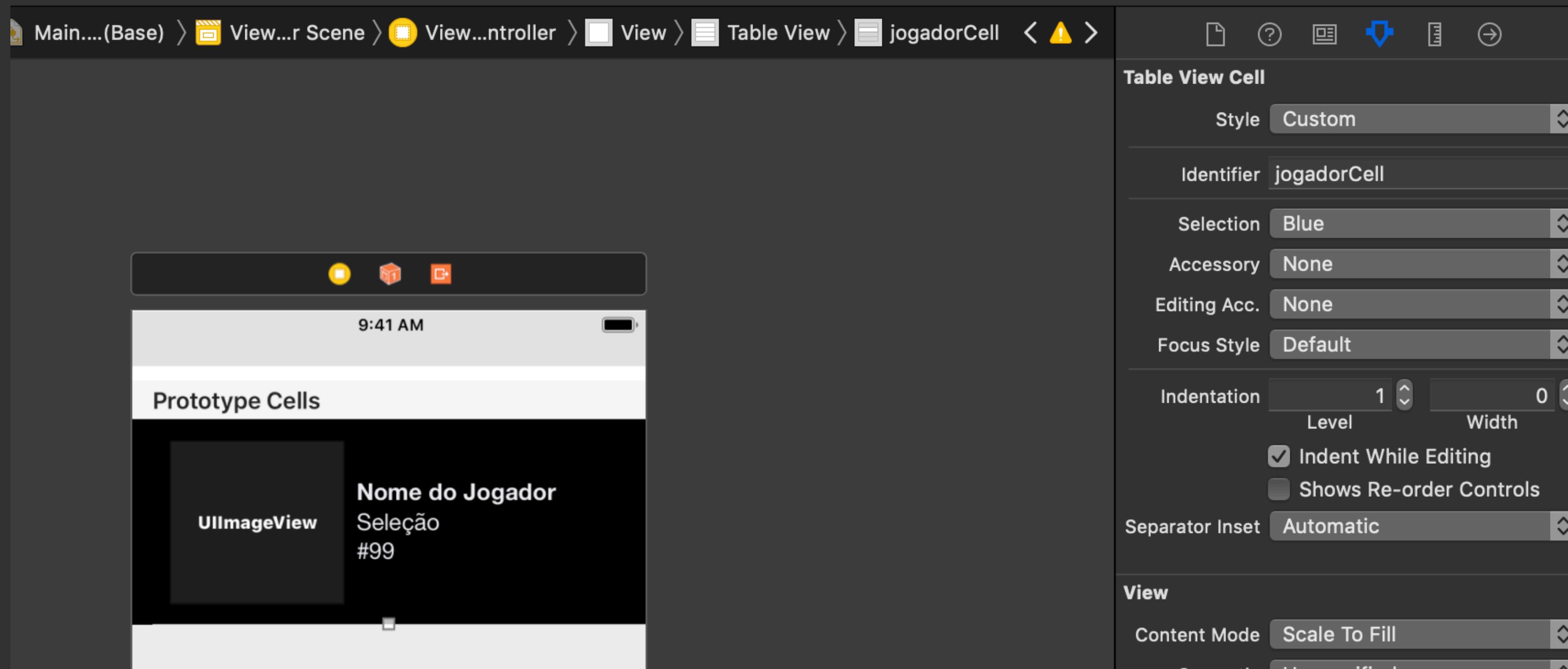
# Crie associação com a cell



# Crie os IBOutlet



# Coloque um identificador em cell

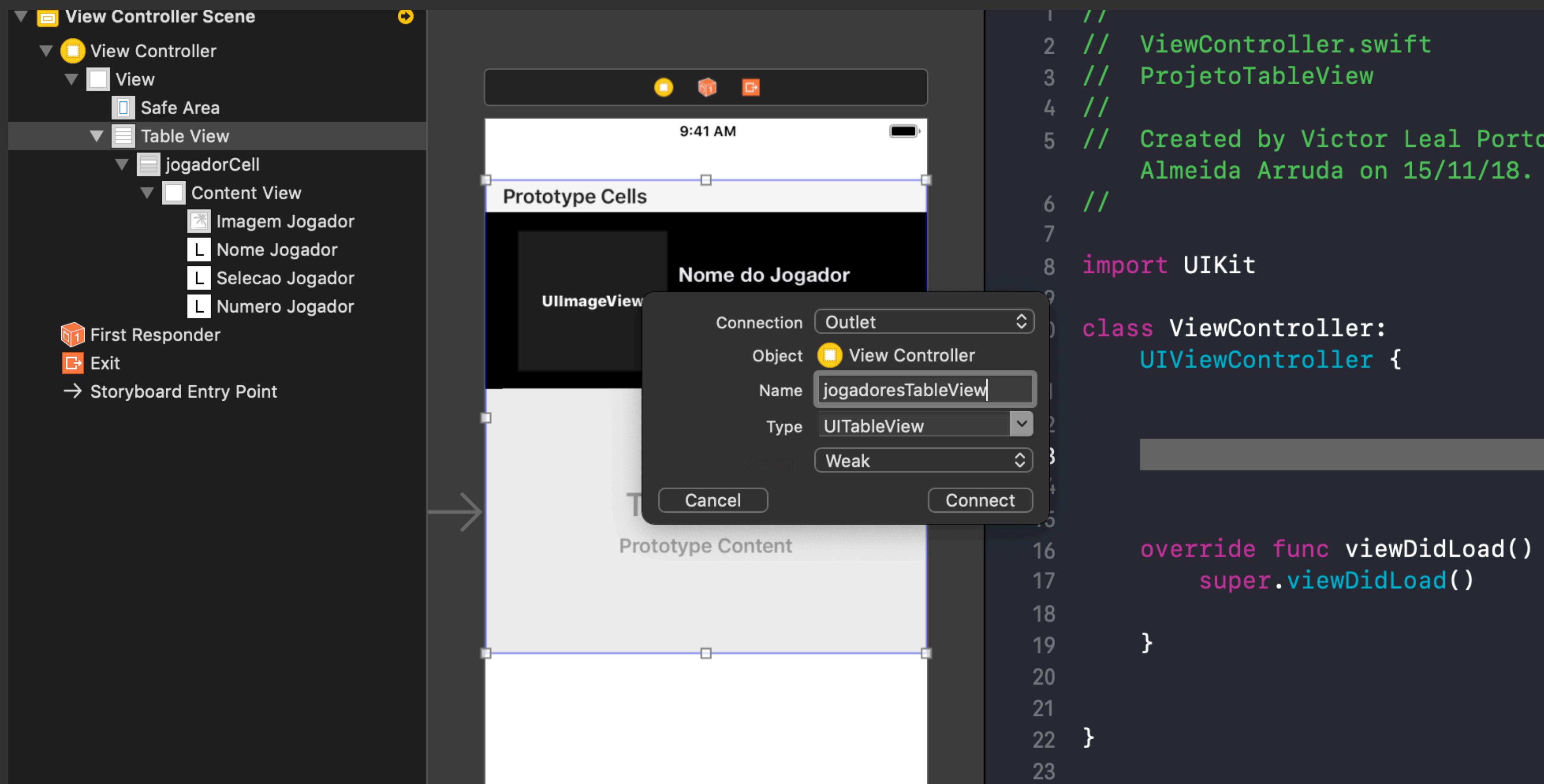


Faça um array de  
objetos Jogador

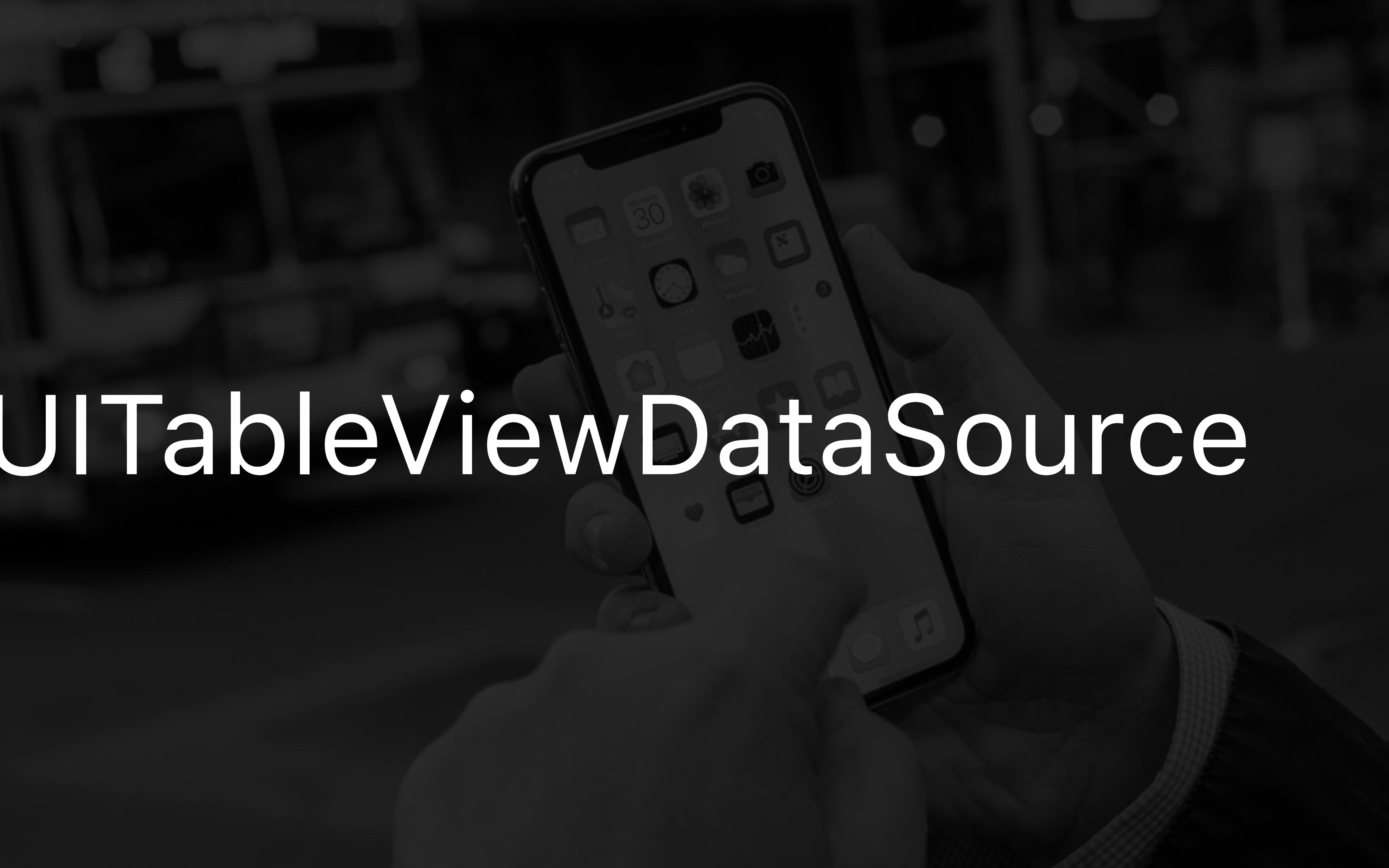
```
class ViewController: UIViewController {  
  
    let jogador1 = Jogador(nome: "Marcelo", selecao: "Brasil", image: 🏴, numero: 6)  
  
    let jogador2 = Jogador(nome: "Kane", selecao: "Inglaterra", image: 🏴, numero: 9)  
  
    let jogador3 = Jogador(nome: "Van Persie", selecao: "Holanda", image: 🇳🇱, numero: 9)  
  
    let jogador4 = Jogador(nome: "Hulk", selecao: "Brasil", image: 🏴, numero: 19)  
  
    lazy var jogadores = [jogador1, jogador2, jogador3, jogador4]  
  
    (...)
```

Voltando pra  
TableView

# Faça o IBOutlet da TableView



# UITableViewDataSource



Número de seções da tabela

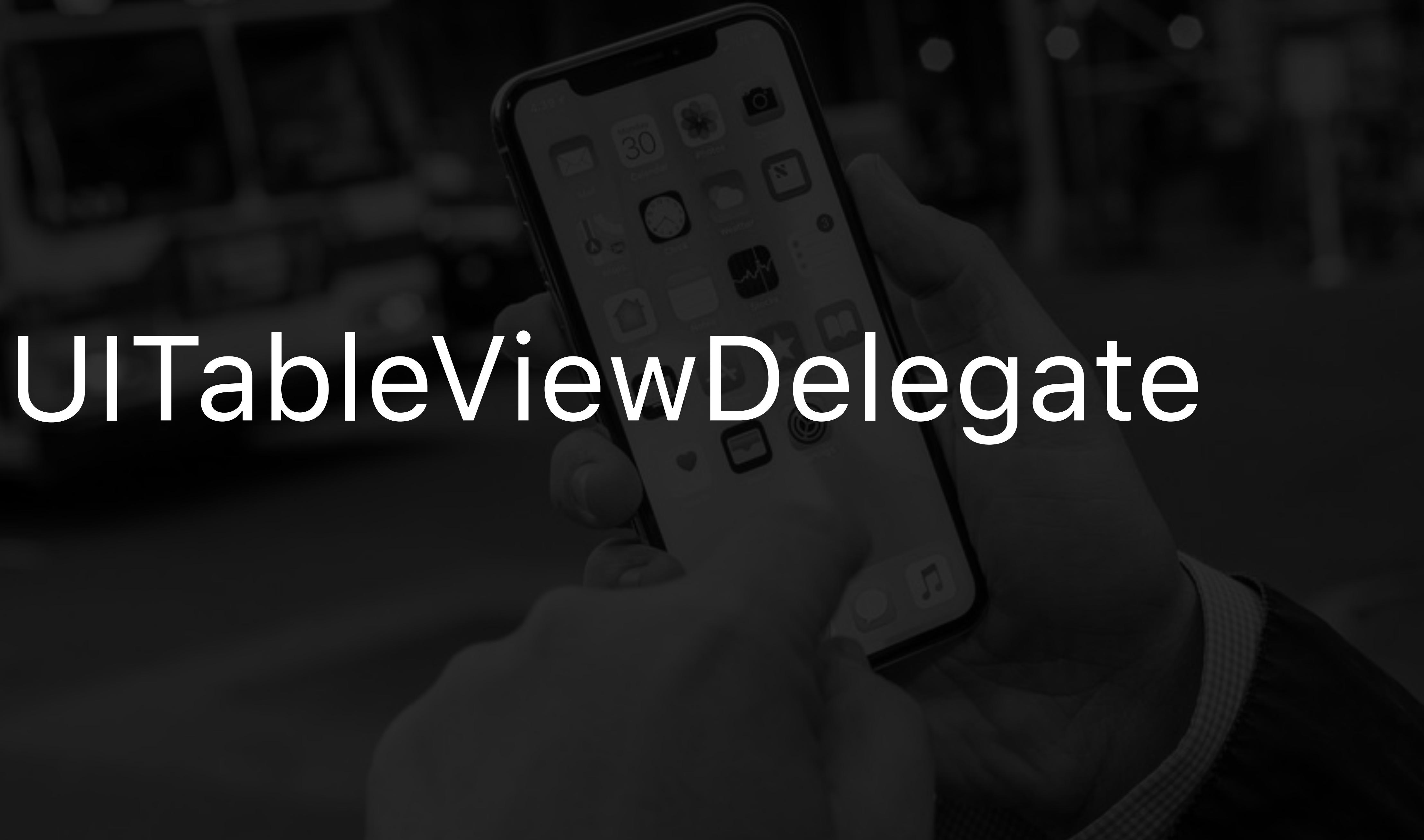
Número de linhas em uma seção

Célula a ser exibida

Título da seção

...

# UITableViewDelegate



O datasource dá informações sobre os dados.  
O delegate é responsável por acessar os dados e  
entregá-los para a view.

# Coloque os protocolos

```
class ViewController: UIViewController, UITableViewDataSource,  
UITableViewDelegate {  
(...)
```

```
func tableView(_ tableView: UITableView,  
 numberOfRowsInSection section: Int) -> Int {
```

<#code#>

```
}
```

```
func tableView(_ tableView: UITableView,  
numberOfRowsInSection section: Int) -> Int {  
  
    return jogadores.count  
  
}
```

```
func tableView(_ tableView: UITableView,  
cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
  
<#code#>  
  
}
```

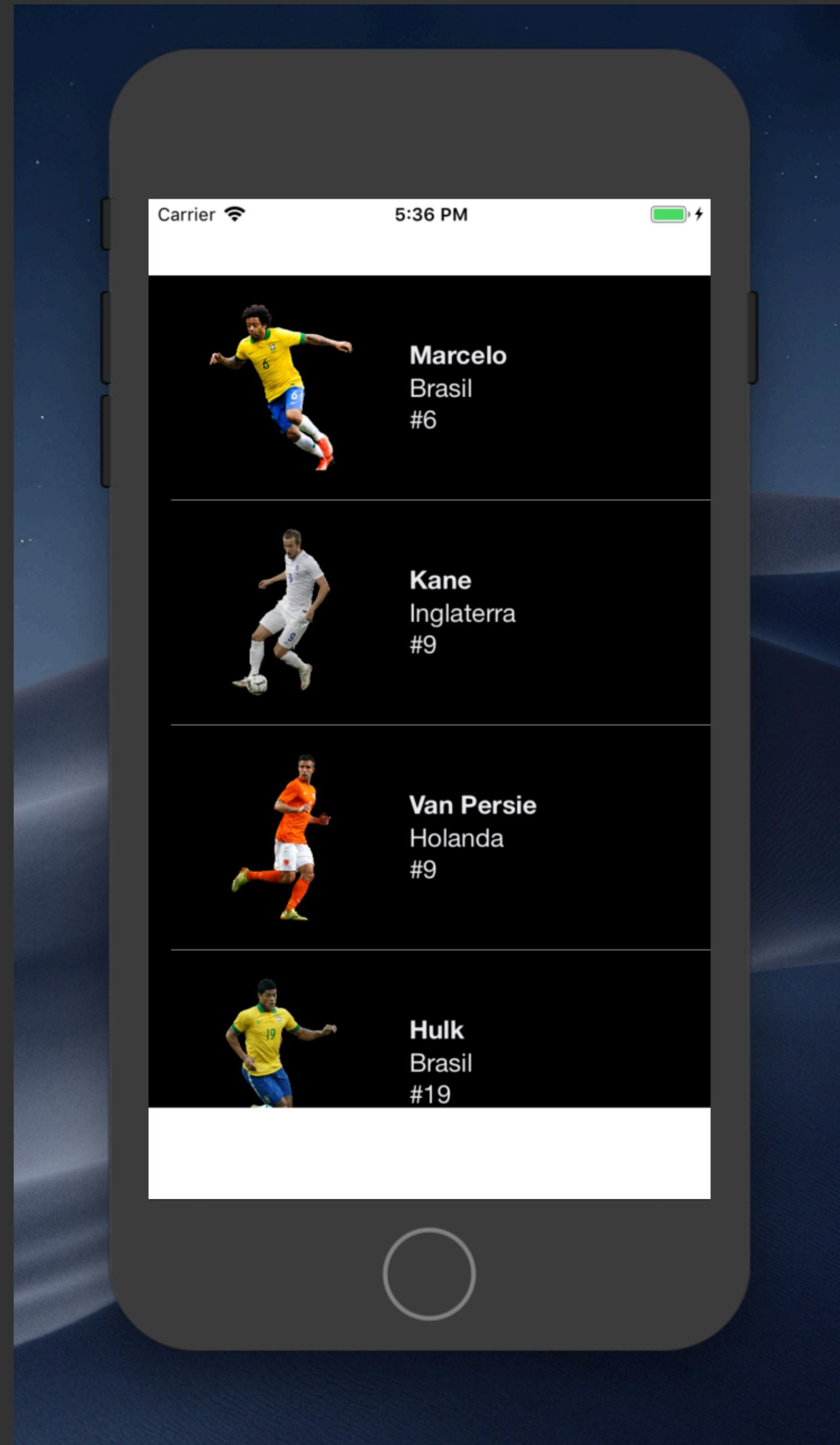
`NSIndexPath.row`  
`NSIndexPath.section`

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    let cell = jogadoresTableView.dequeueReusableCell(withIdentifier: "jogadorCell") as!  
JogadorTableViewCell  
  
    cell.nomeJogador.text = jogadores[indexPath.row].nome  
    cell.selecaoJogador.text = jogadores[indexPath.row].selecao  
    cell.numeroJogador.text = "#" + String(jogadores[indexPath.row].numero)  
    cell.imagemJogador.image = jogadores[indexPath.row].image  
  
    return cell  
}
```

Falta assinar o  
contrato

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    jogadoresTableView.dataSource = self  
    jogadoresTableView.delegate = self  
  
}
```

# Run

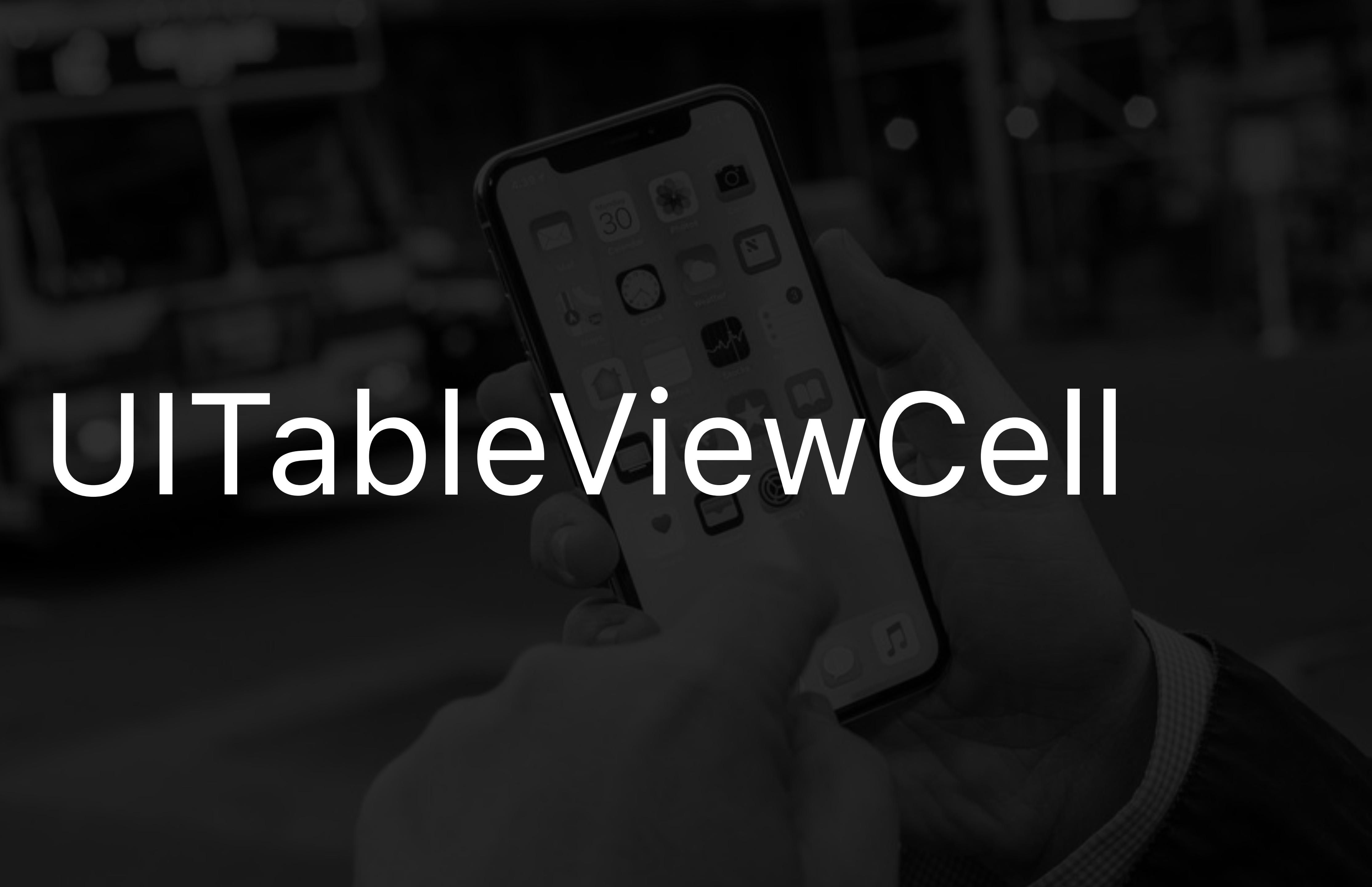


<https://developer.apple.com/documentation/uikit/uitableviewdatasource>

# Dados

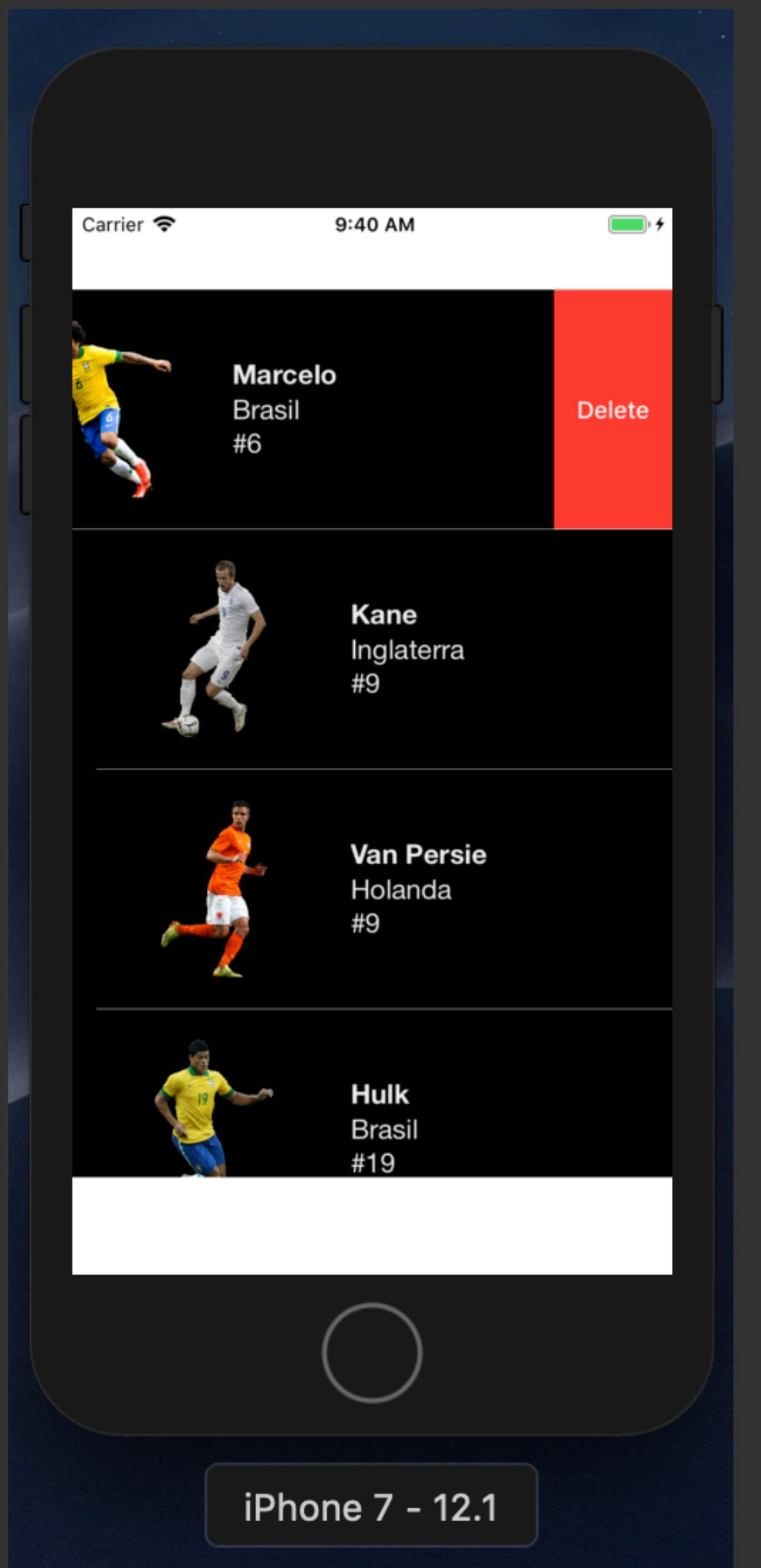


# UITableViewCell



# Remover





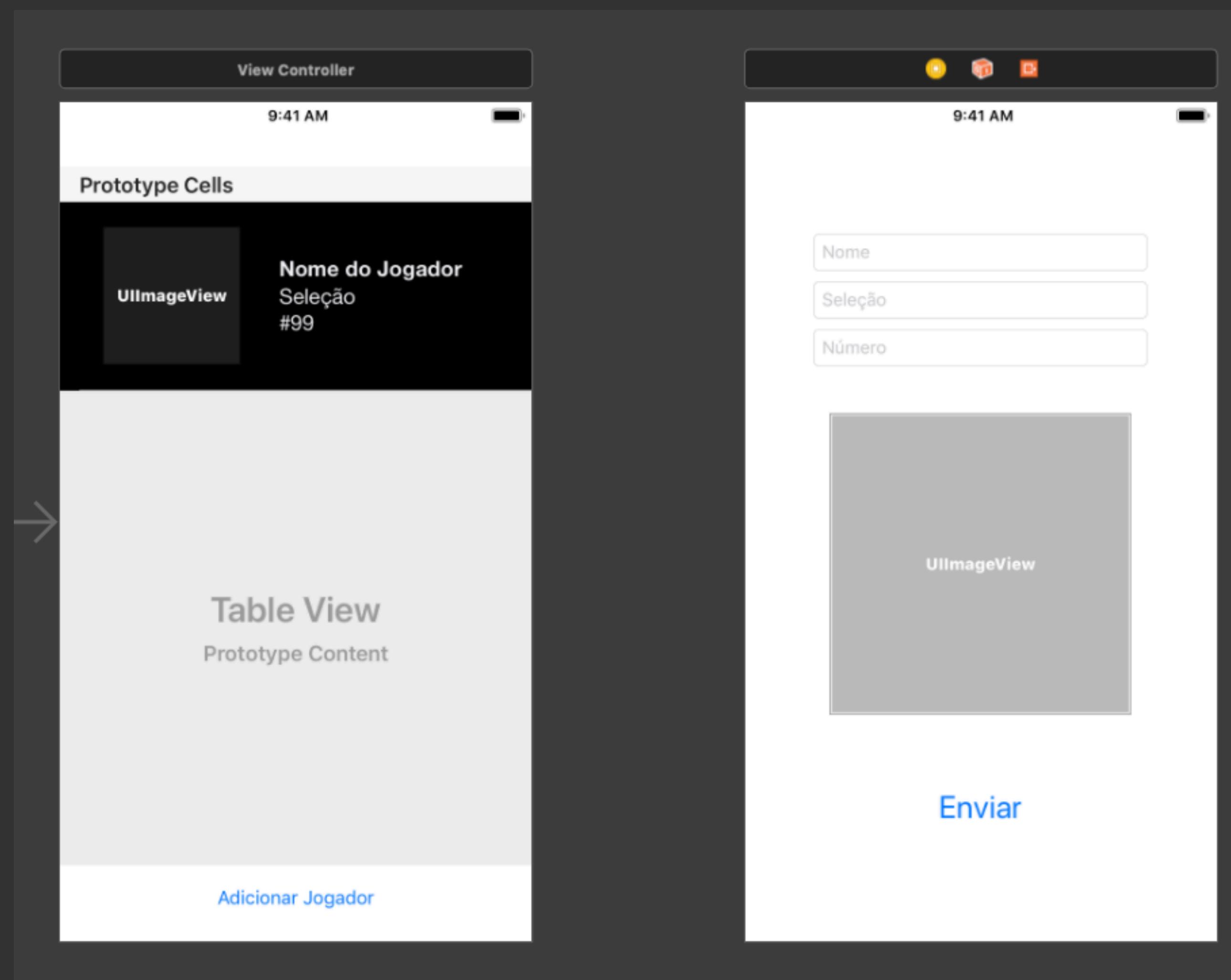
```
func tableView(_ tableView: UITableView,  
commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {  
  
    if editingStyle == .delete {  
        self.jogadores.remove(at: indexPath.row)  
        tableView.deleteRows(at: [indexPath], with: .fade)  
    }  
}
```

# Inserir

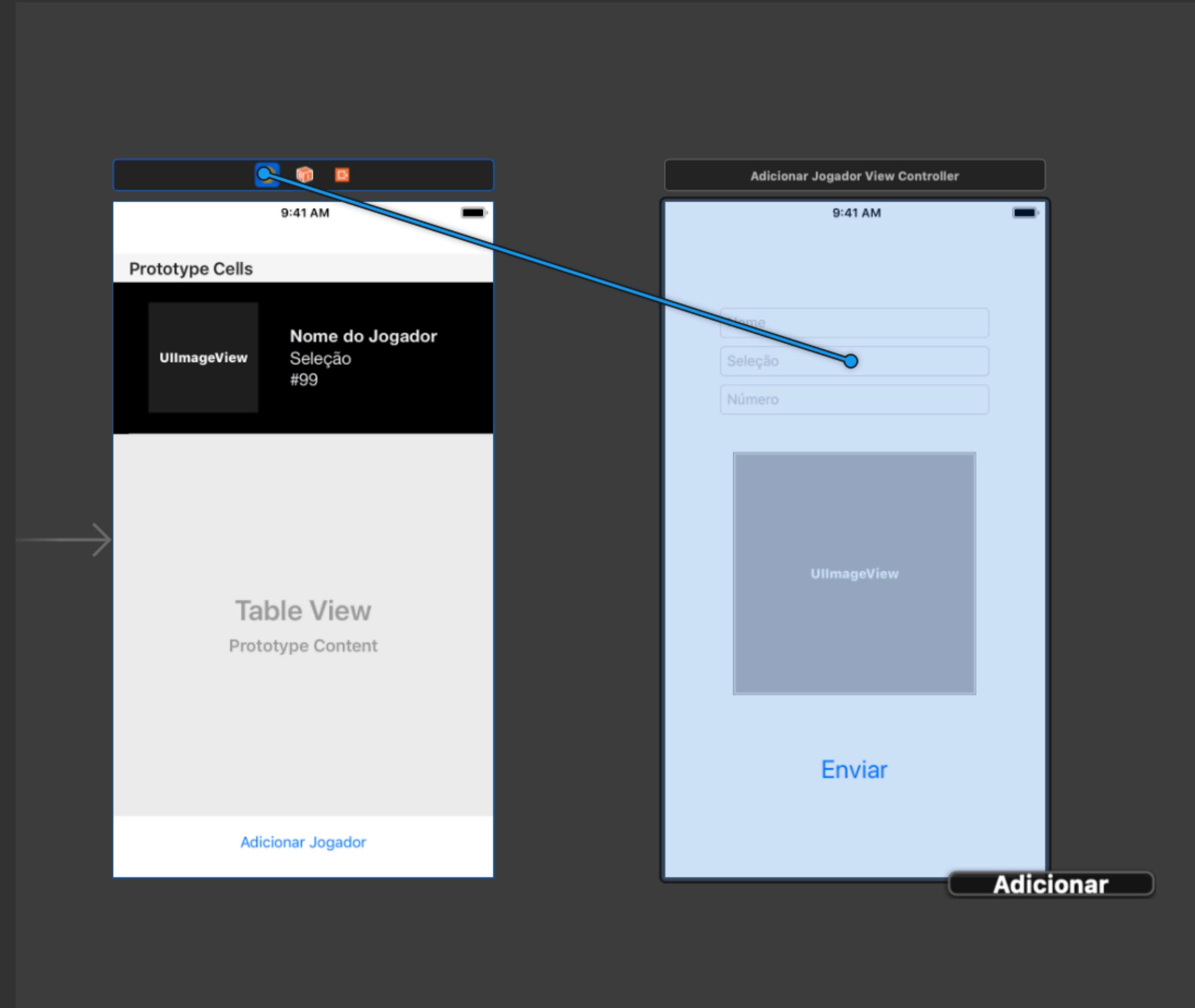


```
func addPlayer (jogador: Jogador){  
    let newIndexPath = IndexPath(row: jogadores.count, section: 0)  
    jogadores.append(jogador)  
    self.jogadoresTableView.insertRows(at: [newIndexPath], with: .automatic)  
}
```

# Coloque um botão pra adicionar jogador e crie uma nova tela



# Crie a ligação entre as telas e coloque um identificador



# Lembre-se de criar a ViewController e as conexões da nova tela

```
import UIKit

class AdicionarJogadorViewController: UIViewController {

    @IBOutlet weak var nome: UITextField!
    @IBOutlet weak var selecao: UITextField!
    @IBOutlet weak var numero: UITextField!
    @IBOutlet weak var imagem: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

    }

    @IBAction func enviar(_ sender: UIButton) {

    }

}
```

Precisamos atualizar a TableView  
estando em outra tela

```
class AdicionarJogadorViewController: UIViewController {

    var viewControllerPrincipal: ViewController!

    @IBOutlet weak var nome: UITextField!
    @IBOutlet weak var selecao: UITextField!
    @IBOutlet weak var numero: UITextField!
    @IBOutlet weak var imagem: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

    }

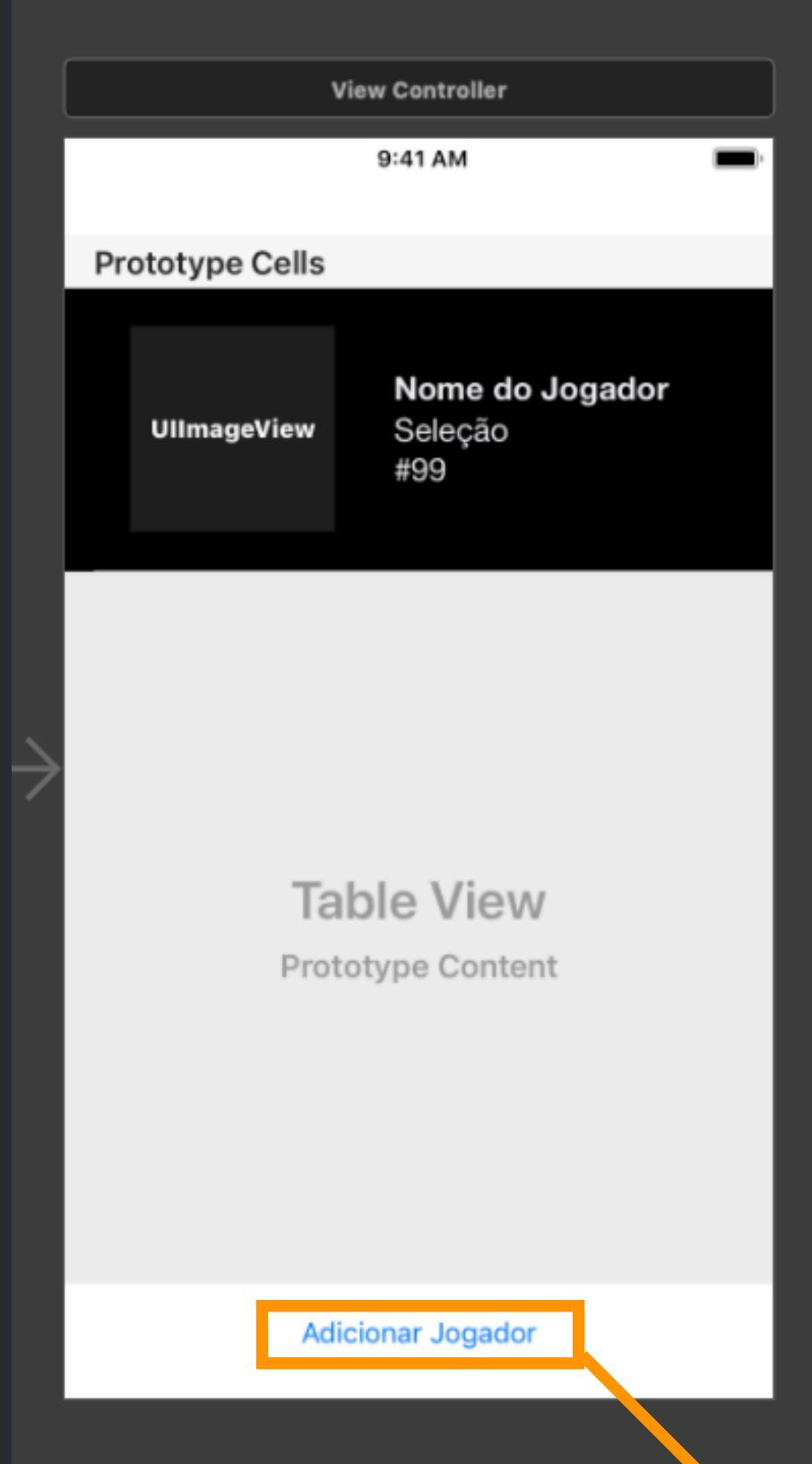
    @IBAction func enviar(_ sender: UIButton) {

        let nomeJ = self.nome.text!
        let selecaoJ = self.selecao.text!
        let numeroJ = Int(self.numero.text!)!
        let imageJ = >

        let jogador = Jogador(nome: nomeJ, selecao: selecaoJ, image: imageJ, numero: numeroJ)

        viewControllerPrincipal.addPlayer(jogador: jogador)
    }
}
```

Voltaremos pra ViewController principal

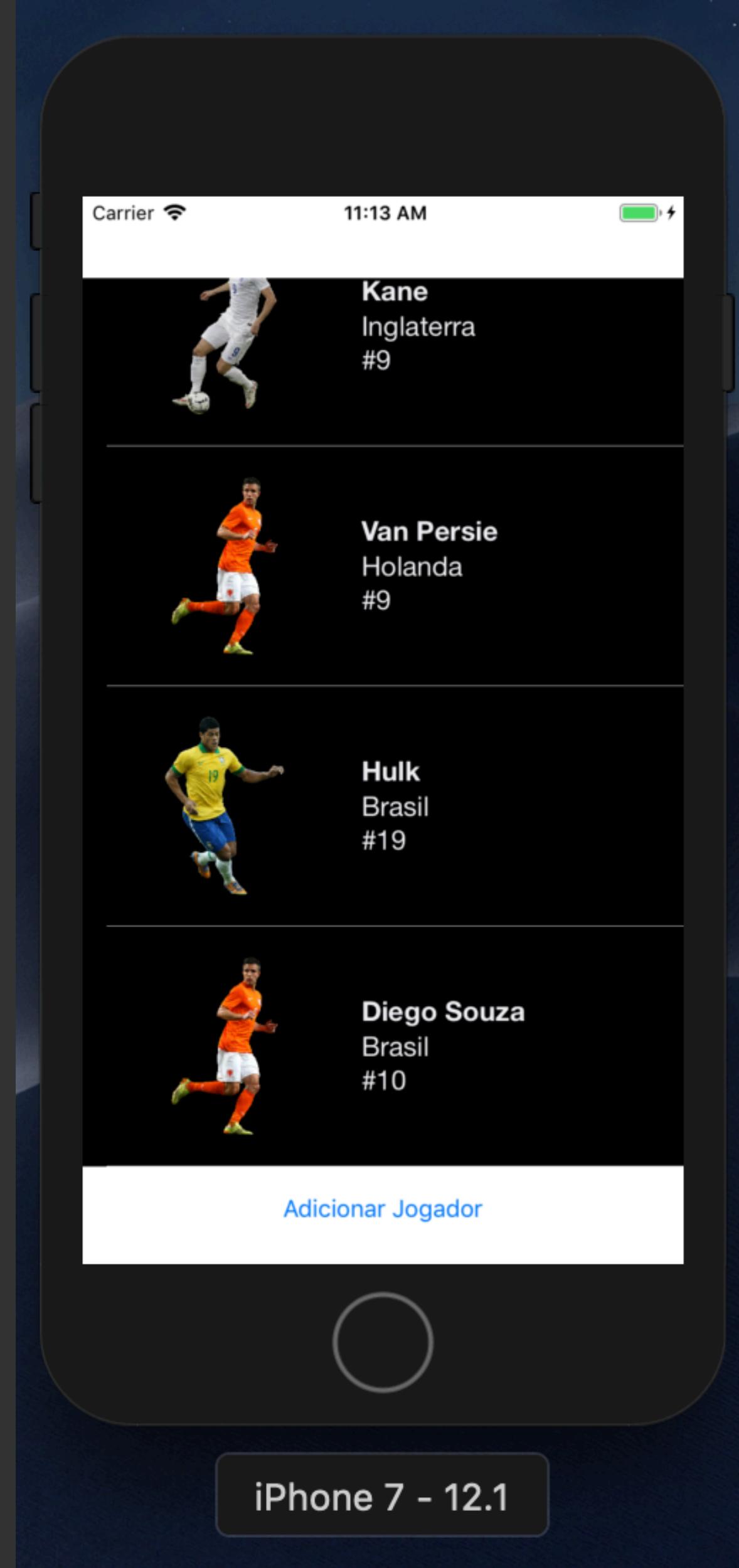
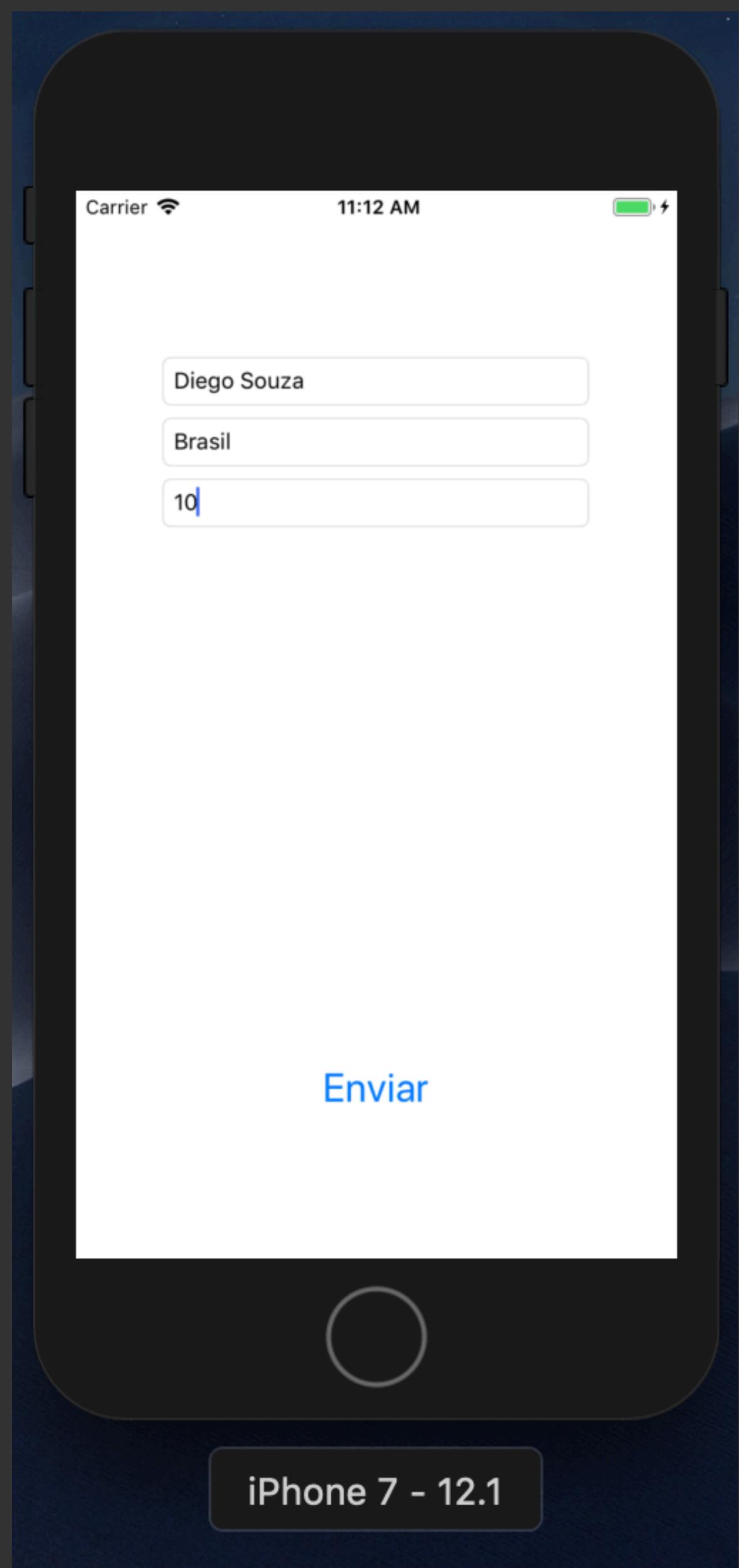


# Crie o IBAction do botão Adicionar Jogador

```
@IBAction func adicionarJogador(_ sender: UIButton) {  
    performSegue(withIdentifier: "paraTelaAdicionar", sender: nil)  
}
```

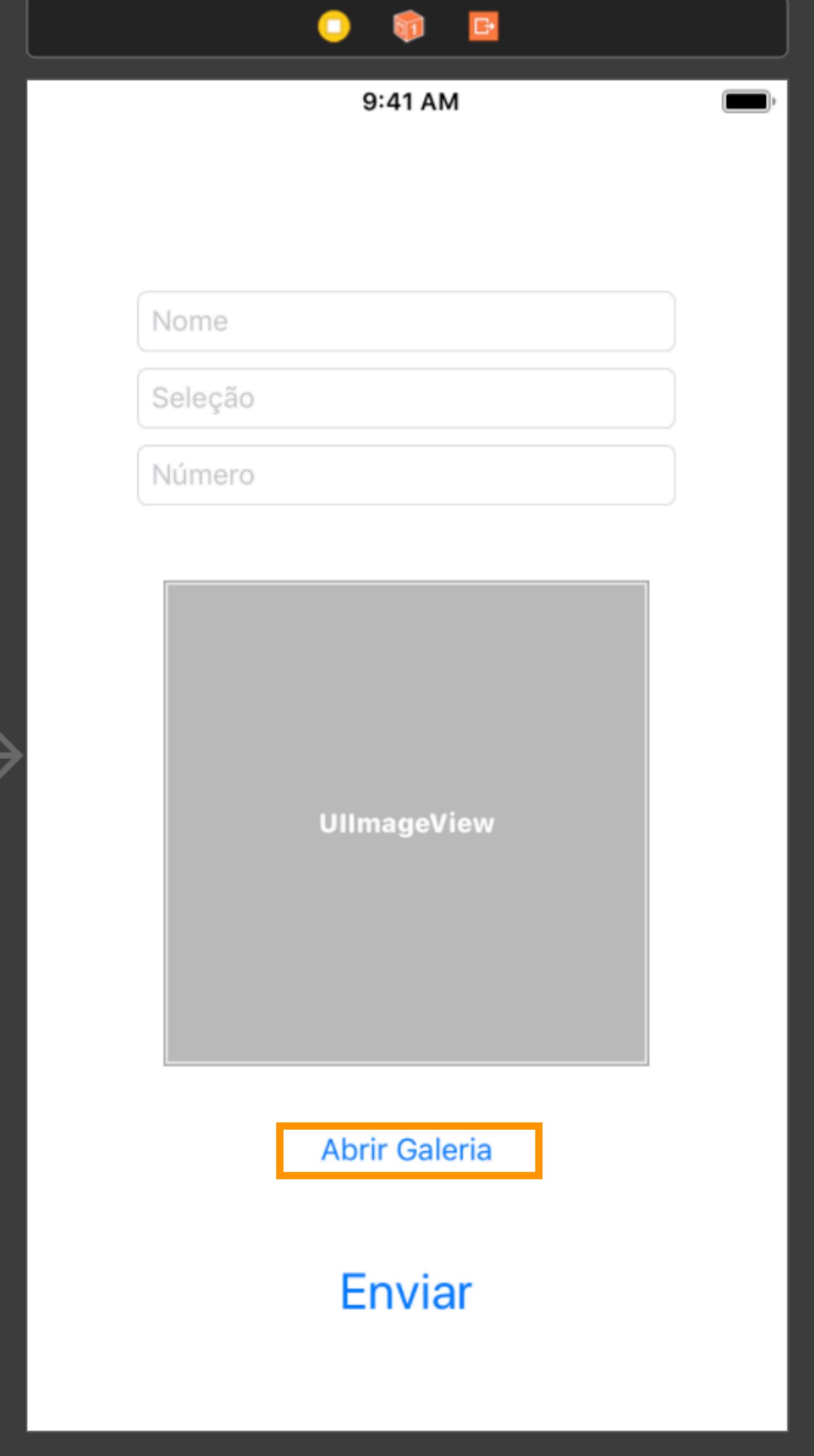
# Inicialize AdicionarJogadorViewController

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if segue.identifier == "paraTelaAdicionar"{  
        if let vcDestino = segue.destination as? AdicionarJogadorViewController{  
            vcDestino.viewControllerPrincipal = self  
        }  
    }  
}
```



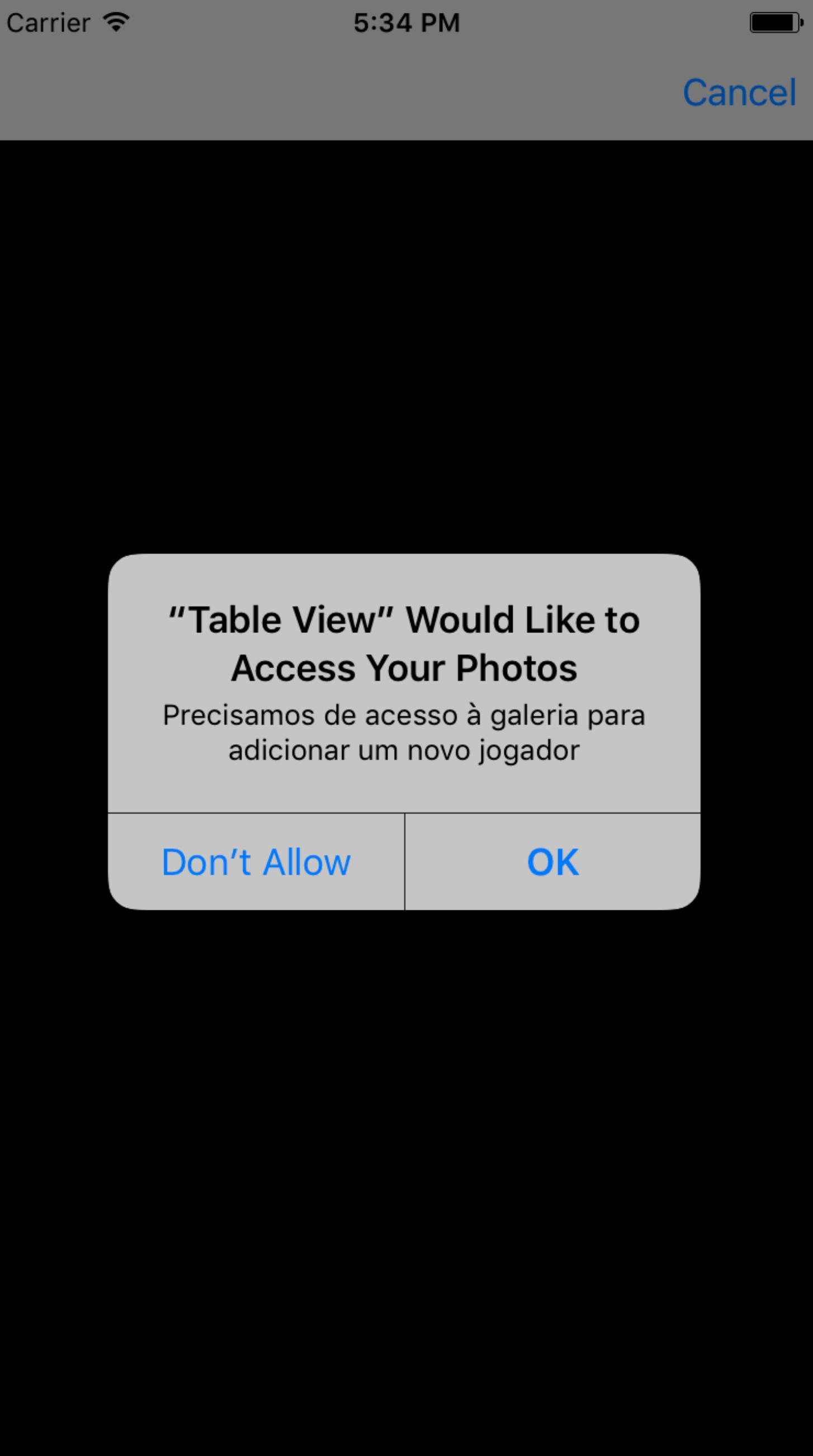
# Acesso à Galeria

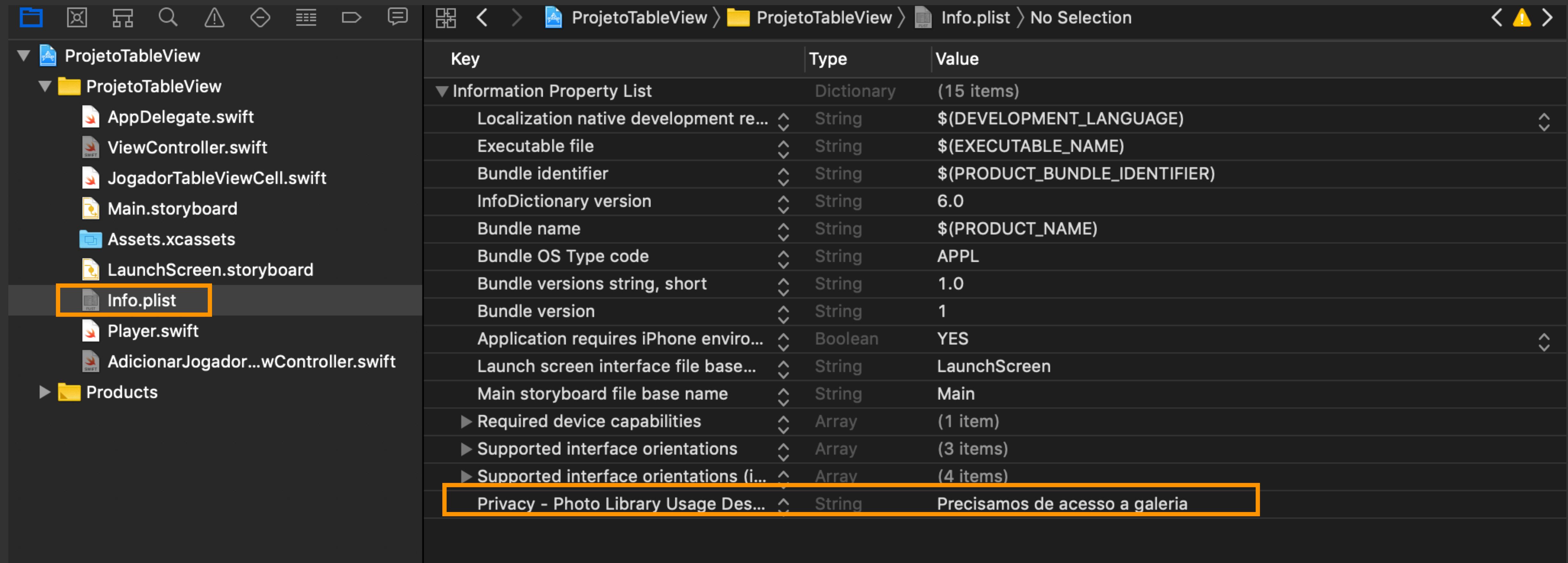




```
@IBAction func abrirGaleria(_ sender: Any) {  
}
```

# Permissão





The screenshot shows the Xcode interface with the Project TableView selected. The left sidebar shows files like AppDelegate.swift, ViewController.swift, and Info.plist. The Info.plist file is open in the main editor. A specific entry, 'Privacy - Photo Library Usage Description', is highlighted with an orange border.

Key	Type	Value
Localization native development region	String	\$(DEVELOPMENT_LANGUAGE)
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)
Supported interface orientations	Array	(3 items)
Supported interface orientations (iPhone)	Array	(4 items)
Privacy - Photo Library Usage Description	String	Precisamos de acesso a galeria

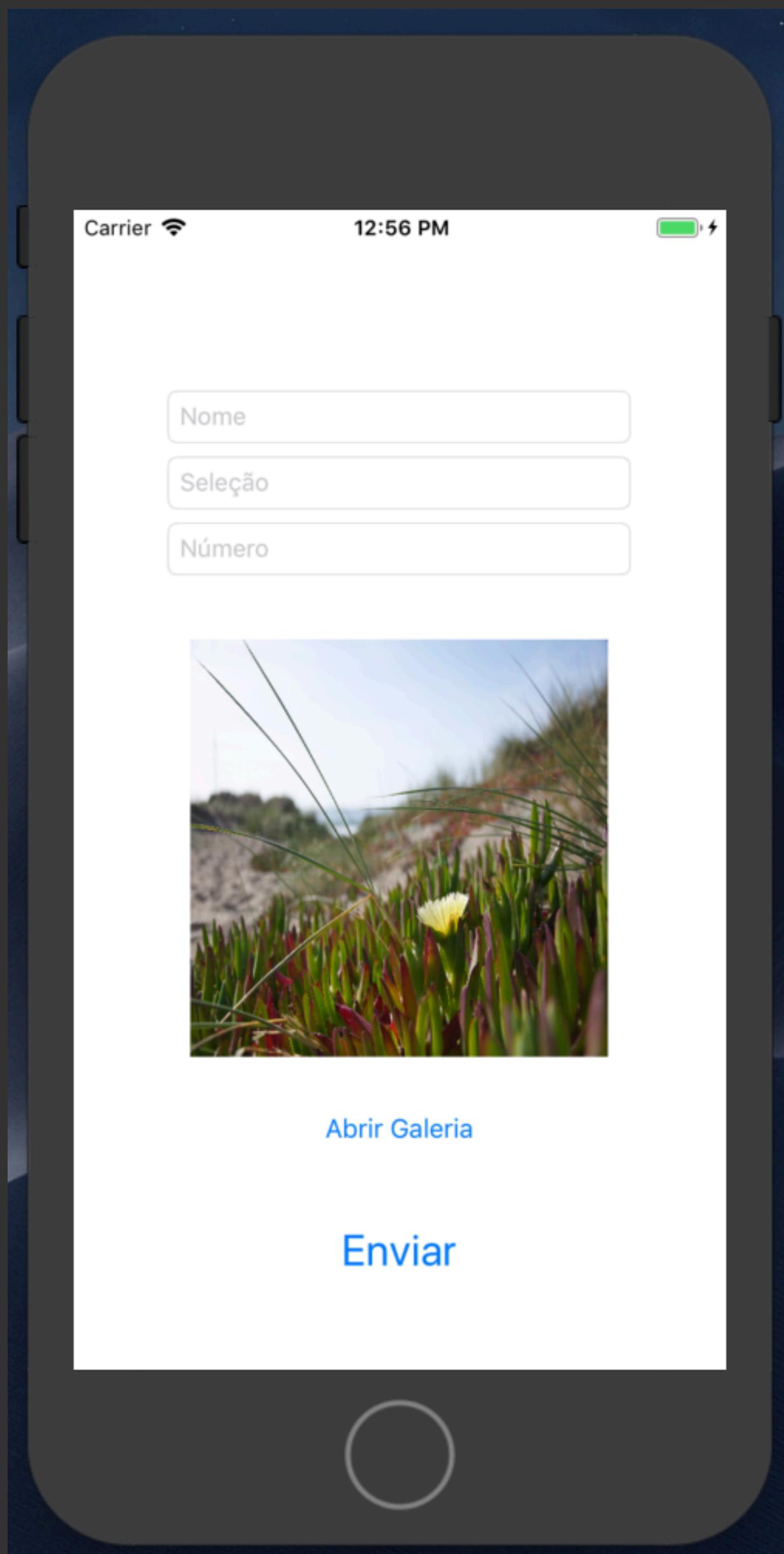
# Modifique o info.plist

**Adicione os protocolos:**  
**UIImagePickerControllerDelegate**  
**UINavigationControllerDelegate**

```
class AdicionarJogadorViewController: UIViewController,  
UIImagePickerControllerDelegate, UINavigationControllerDelegate {  
(...)
```

```
@IBAction func abrirGaleria(_ sender: Any) {  
    if UIImagePickerController.isSourceTypeAvailable(.photoLibrary) {  
        let imagemGaleria = UIImagePickerController()  
        imagemGaleria.delegate = self  
        imagemGaleria.sourceType = .photoLibrary;  
        imagemGaleria.allowsEditing = true  
        self.present(imagemGaleria, animated: true, completion: nil)  
    }  
}
```

```
func imagePickerController(_ picker: UIImagePickerController,  
                         didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {  
  
    if let imagem = info[UIImagePickerController.InfoKey.originalImage] as? UIImage{  
        self.imagem.image = imagem  
    }  
  
    picker.dismiss(animated: true, completion: nil)  
}
```



# Ajuste a função de enviar

```
@IBAction func enviar(_ sender: UIButton) {  
  
    let nomeJ = self.nome.text!  
    let selecaoJ = self.selecao.text!  
    let numeroJ = Int(self.numero.text!)!  
    let imageJ = imagem.image!  
  
    let jogador = Jogador(nome: nomeJ, selecao: selecaoJ, image: imageJ, numero: numeroJ)  
  
    viewControllerPrincipal.addPlayer(jogador: jogador)  
  
    self.dismiss(animated: true)  
}
```

