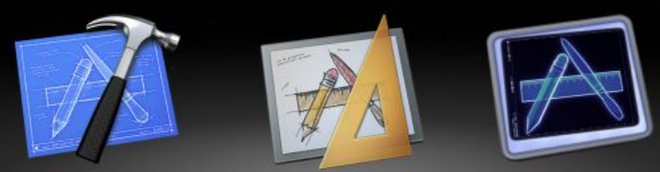


PROF. RENÊ XAVIER

DESENVOLVIMENTO PARA IOS 11 COM SWIFT 4

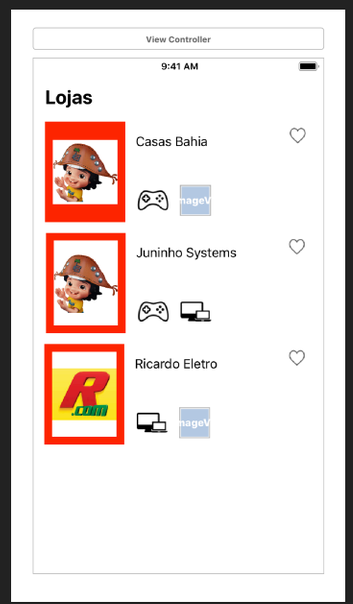
ONDE ENCONTRAR O MATERIAL?

HTTPS://GITHUB.COM/RENEFX/IOS-2018-01

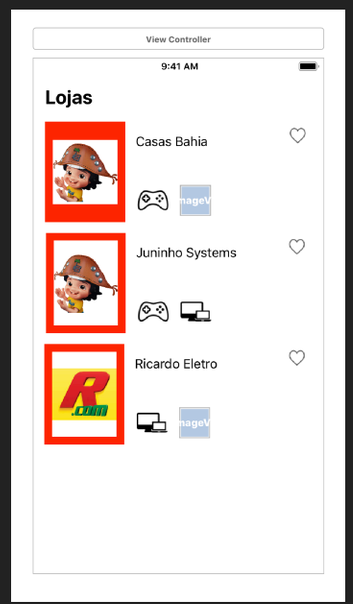
GITHUB

ALÉM DO TRADICIONAL BLACKBOARD DO IESB

CHEGA DE CRIAR VÁRIOS BAHIANINHOS

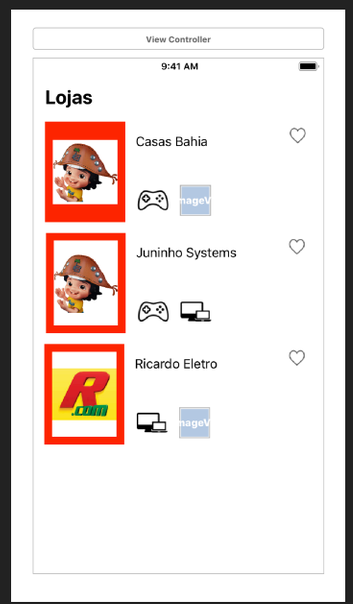


SE VAMOS COLOCAR UM SCROLL, VAMOS USAR UMA...

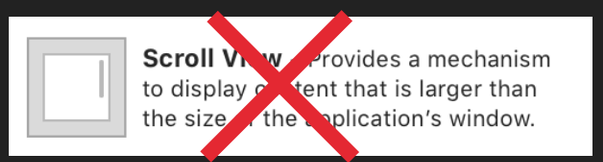


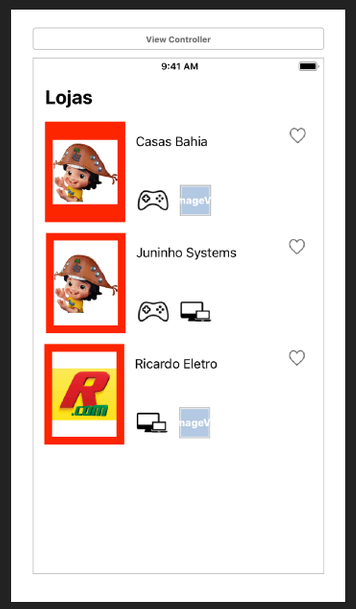
SE VAMOS COLOCAR UM SCROLL, VAMOS USAR UMA...

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.



SE VAMOS COLOCAR UM SCROLL, VAMOS USAR UMA...



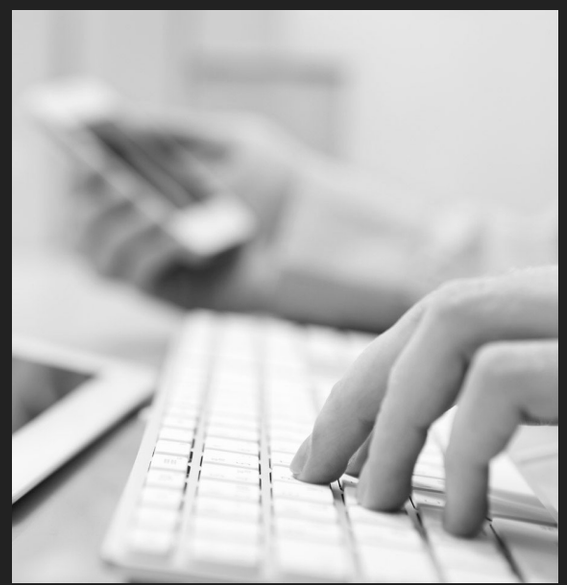


COMO UMA SCROLLVIEW DEVE SER USADA

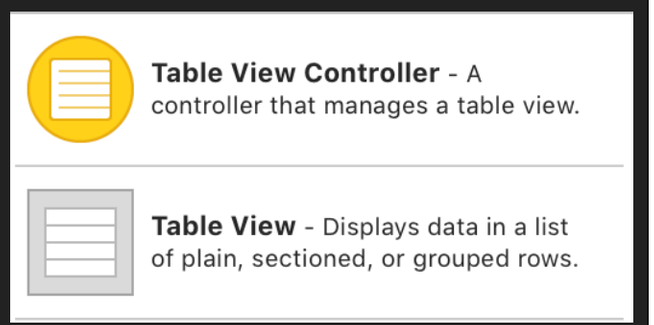
O QUE VAMOS FAZER HOJE?

AGENDA

- UITableView e UITableViewController
- Células customizadas
- Prática



TABLEVIEW CONTROLLER OU TABLEVIEW?



TABLEVIEW CONTROLLER

TABLEVIEW CONTROLLER

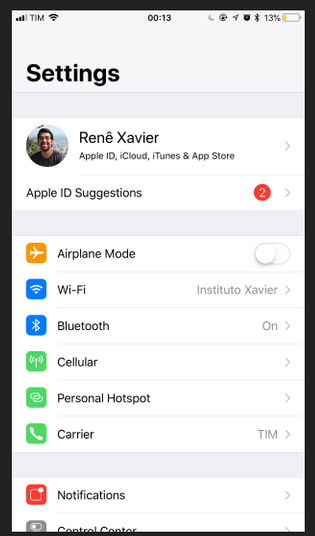
- Uma tela completa por uma lista (TableView)

TABLEVIEW CONTROLLER

- Uma tela completa por uma lista (TableView)

TABLEVIEW CONTROLLER

- Uma tela completa por uma lista (TableView)



TABLEVIEW CONTROLLER

- Vamos a sua anatomia:

TABLEVIEW CONTROLLER

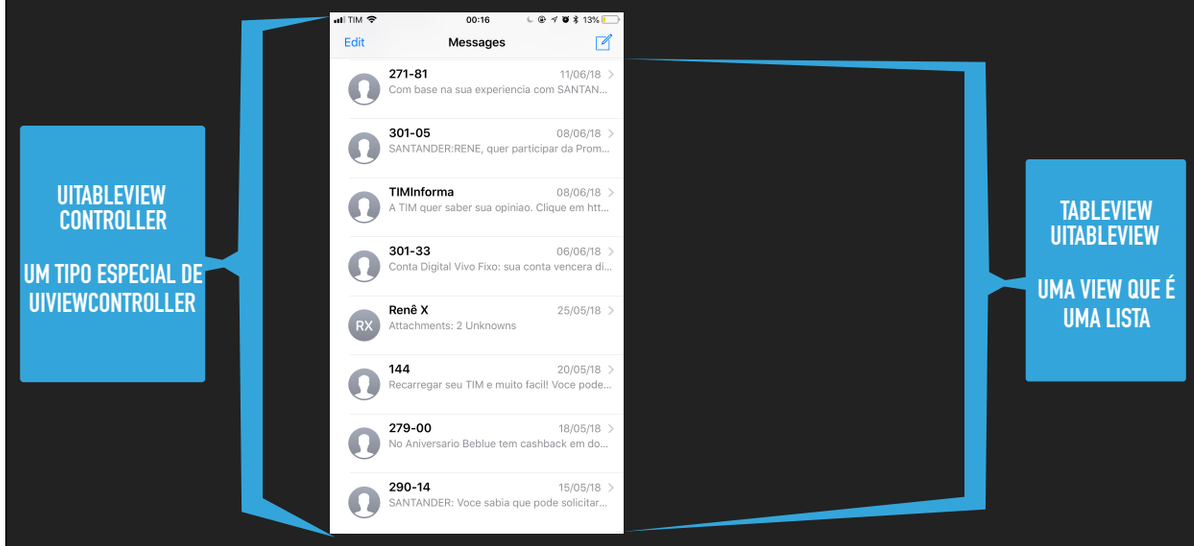
- Vamos a sua anatomia:

UITABLEVIEW CONTROLLER

UM TIPO ESPECIAL DE UIVIEWCONTROLLER

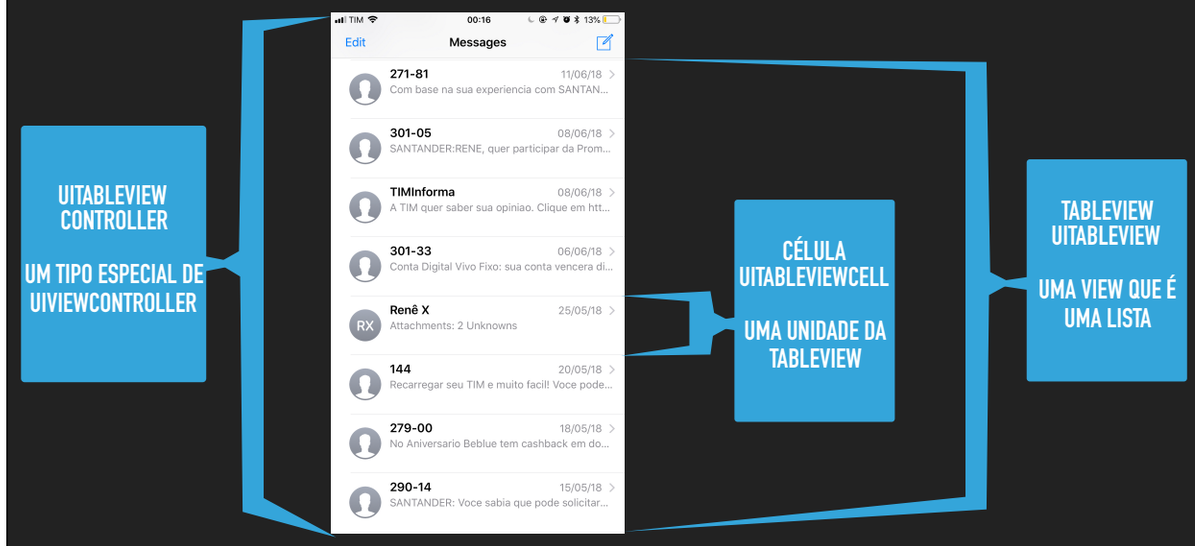
TABLEVIEW CONTROLLER

- Vamos a sua anatomia:



TABLEVIEW CONTROLLER

- Vamos a sua anatomia:



TABLEVIEW CONTROLLER

COM OU SEM SEÇÕES!?

TABLEVIEW CONTROLLER

COM OU SEM SEÇÕES!?

TABLEVIEW CONTROLLER

COM OU SEM SEÇÕES!?

SEM SEÇÕES

TABLEVIEW CONTROLLER

COM OU SEM SEÇÕES!?

SEM SEÇÕES

TABLEVIEW CONTROLLER

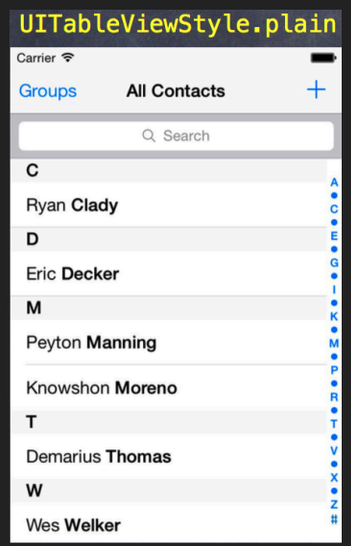
COM OU SEM SEÇÕES!?

SEM SEÇÕES

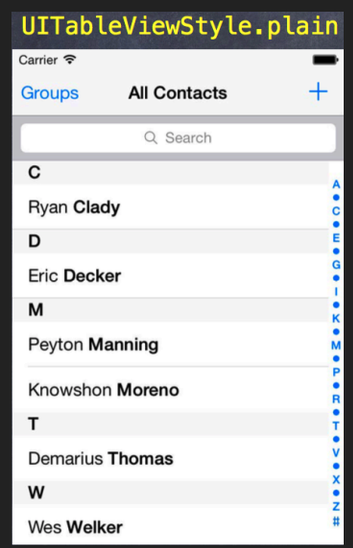
COM SEÇÕES

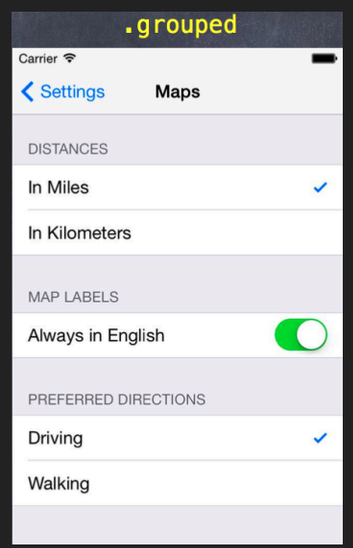
TABLEVIEW CONTROLLER

TABLEVIEW CONTROLLER



TABLEVIEW CONTROLLER



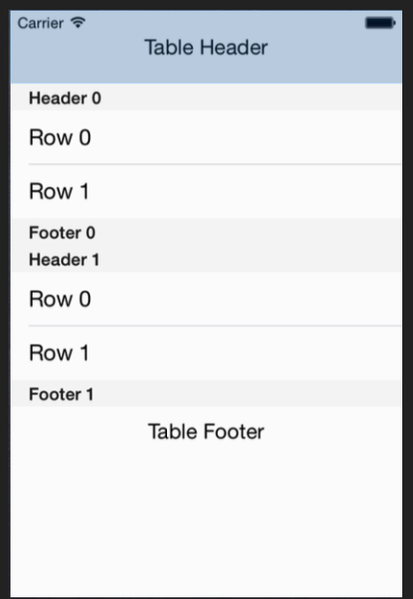


TABLEVIEW CONTROLLER

PLAIN STYLE

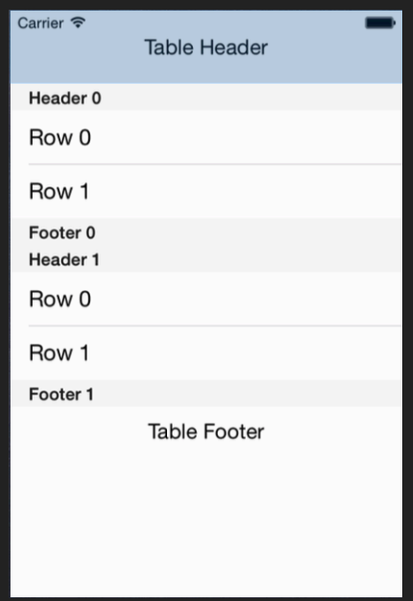
TABLEVIEW CONTROLLER

PLAIN STYLE



TABLEVIEW CONTROLLER

PLAIN STYLE

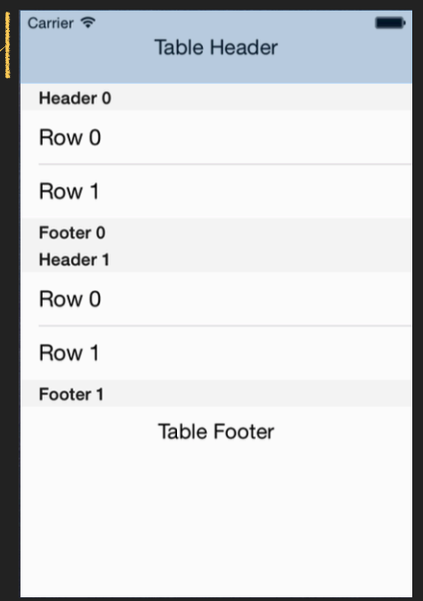



```
var tableHeaderView: UIView
```

TABLEVIEW CONTROLLER

PLAIN STYLE

CABEÇALHO DA TABELA
(TABLE HEADER)

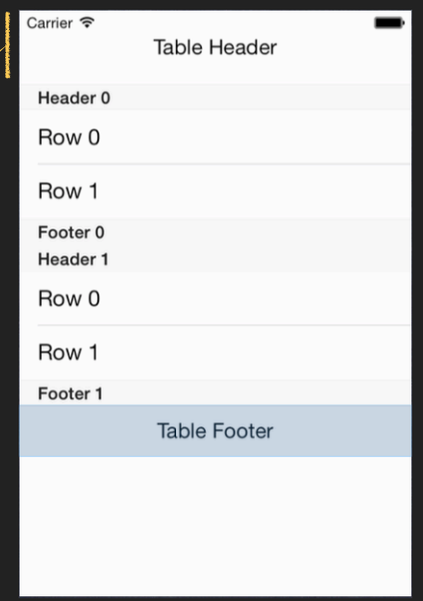



```
var tableHeaderView: UIView
```

TABLEVIEW CONTROLLER

PLAIN STYLE

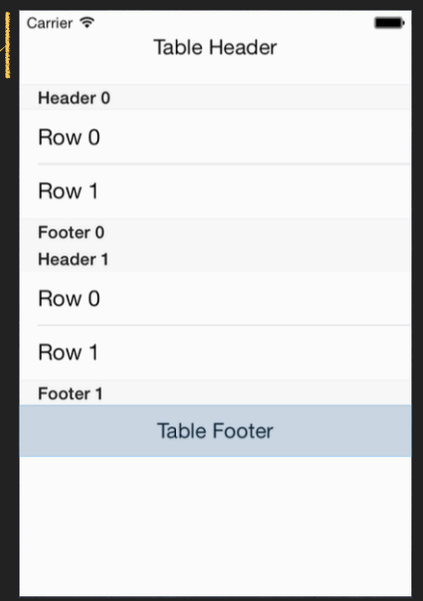
CABEÇALHO DA TABELA
(TABLE HEADER)



TABLEVIEW CONTROLLER

PLAIN STYLE

CABEÇALHO DA TABELA
(TABLE HEADER)



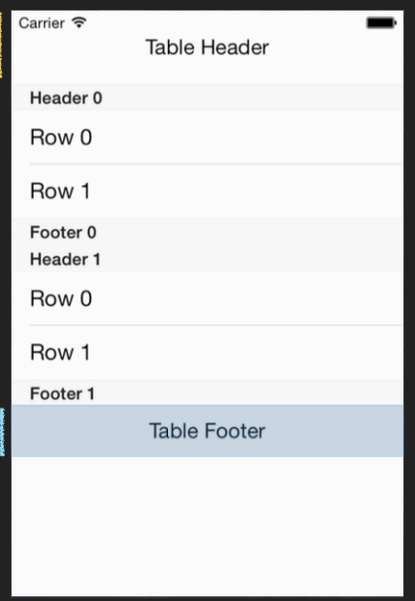

```
var tableFooterView: UIView
```

TABLEVIEW CONTROLLER

PLAIN STYLE

CABEÇALHO DA TABELA
(TABLE HEADER)

RODAPÉ DA TABELA
(TABLE FOOTER)



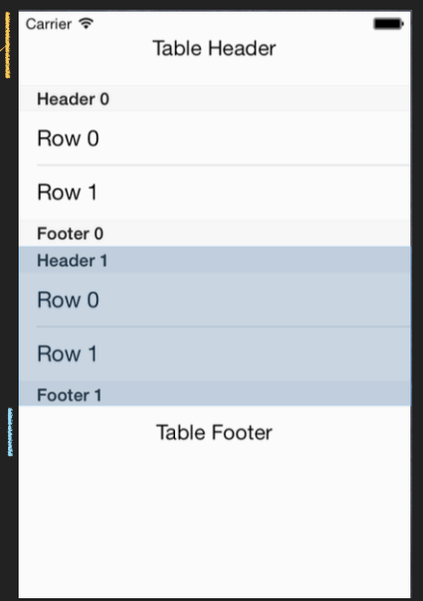

```
var tableFooterView: UIView
```

TABLEVIEW CONTROLLER

PLAIN STYLE

CABEÇALHO DA TABELA
(TABLE HEADER)

RODAPÉ DA TABELA
(TABLE FOOTER)



TABLEVIEW CONTROLLER

PLAIN STYLE

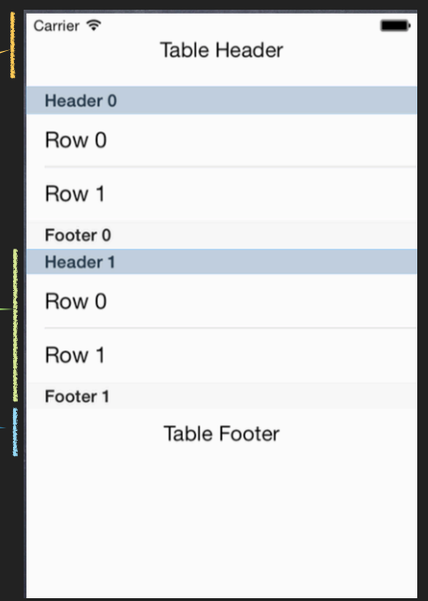
TABLEVIEW CONTROLLER

PLAIN STYLE

CABEÇALHO DA TABELA
(TABLE HEADER)

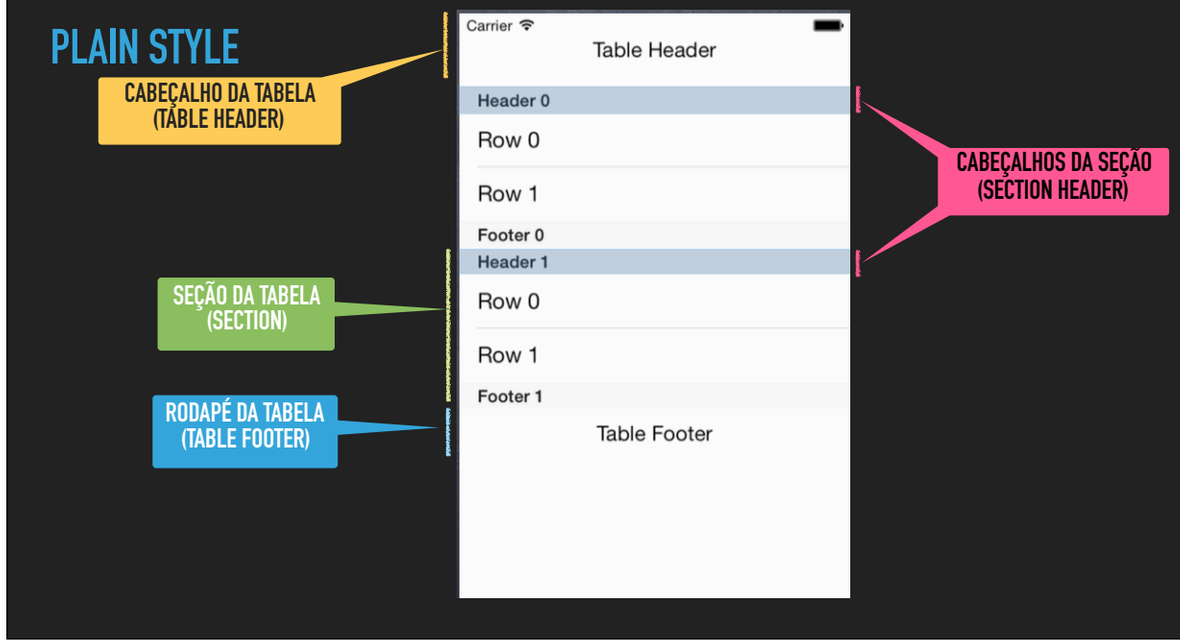
SEÇÃO DA TABELA
(SECTION)

RODAPÉ DA TABELA
(TABLE FOOTER)



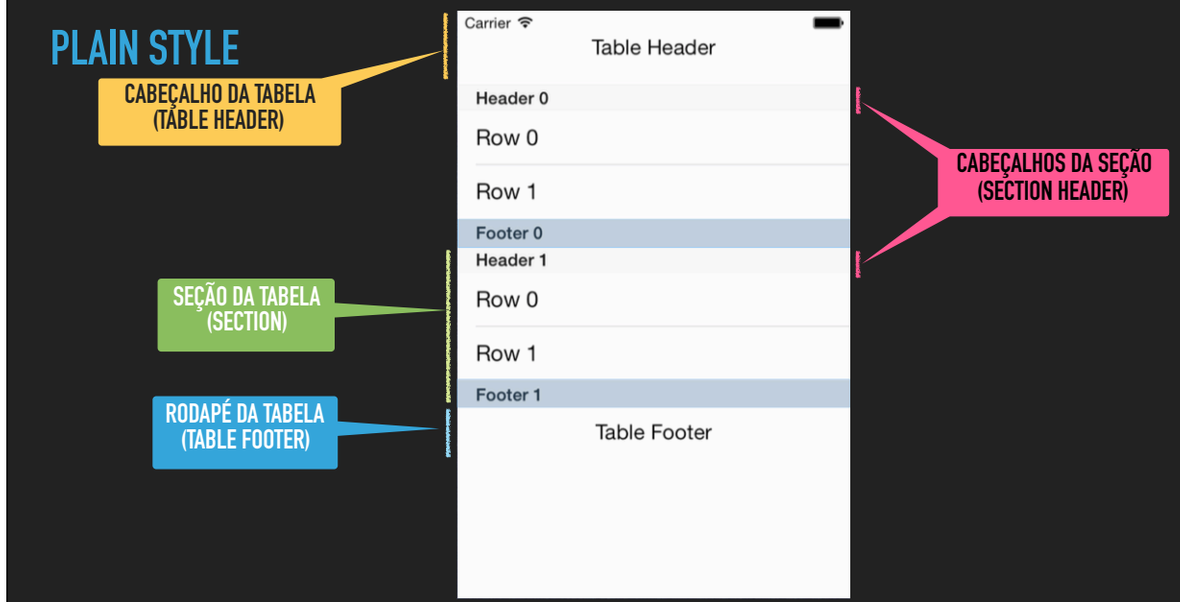
TABLEVIEW CONTROLLER

PLAIN STYLE



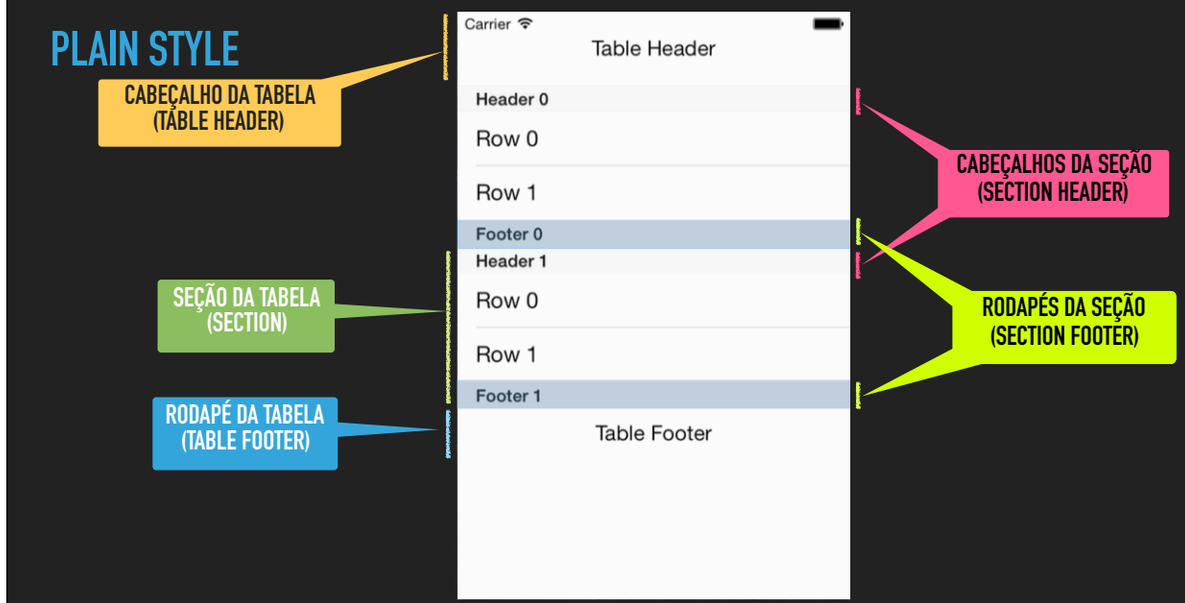
TABLEVIEW CONTROLLER

PLAIN STYLE



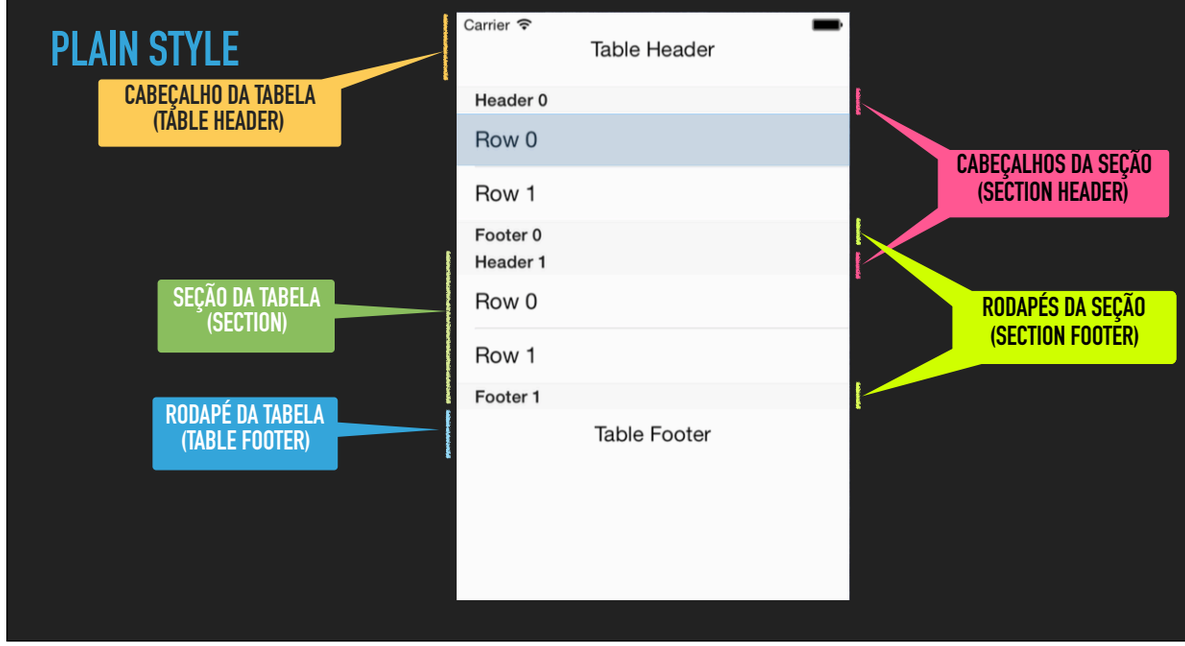
TABLEVIEW CONTROLLER

PLAIN STYLE



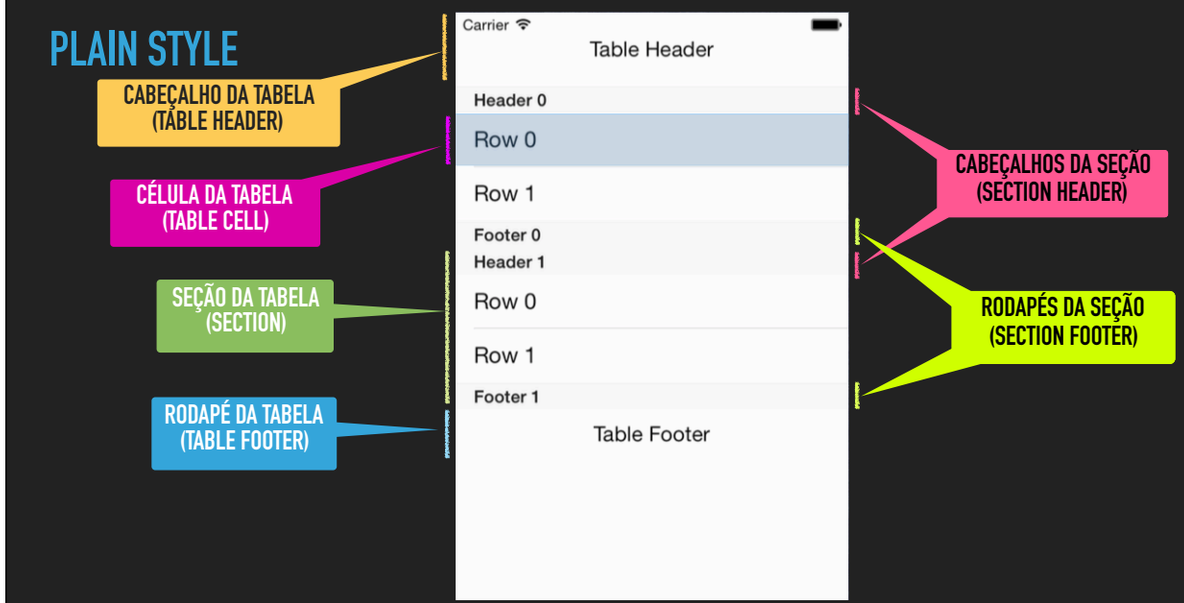
TABLEVIEW CONTROLLER

PLAIN STYLE



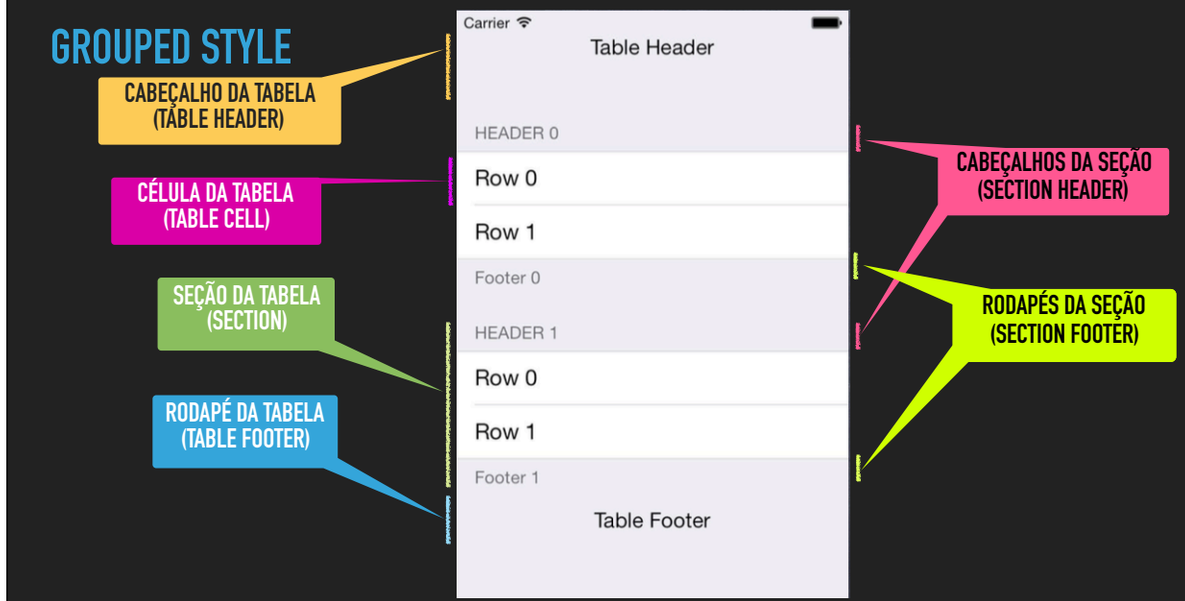
TABLEVIEW CONTROLLER

PLAIN STYLE



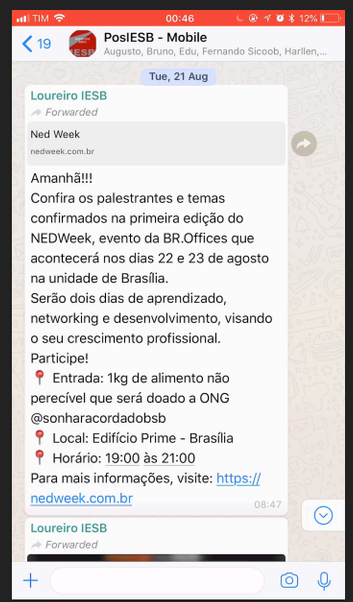
TABLEVIEW CONTROLLER

GROUPED STYLE

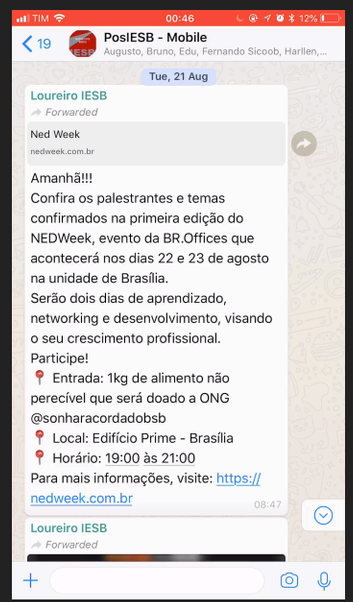


ONDE EU JÁ VI UMA TABLEVIEW CONTROLLER GROUPED?

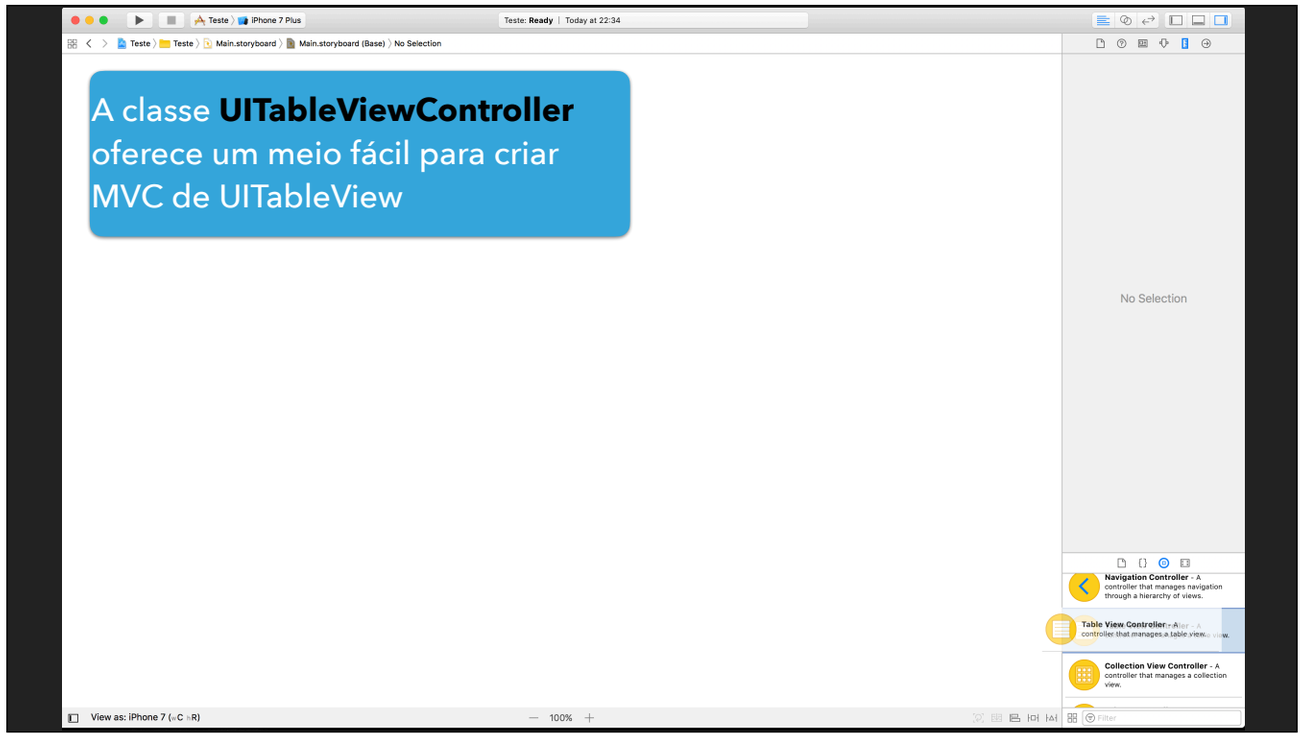
ONDE EU JÁ VI UMA TABLEVIEW CONTROLLER GROUPED?

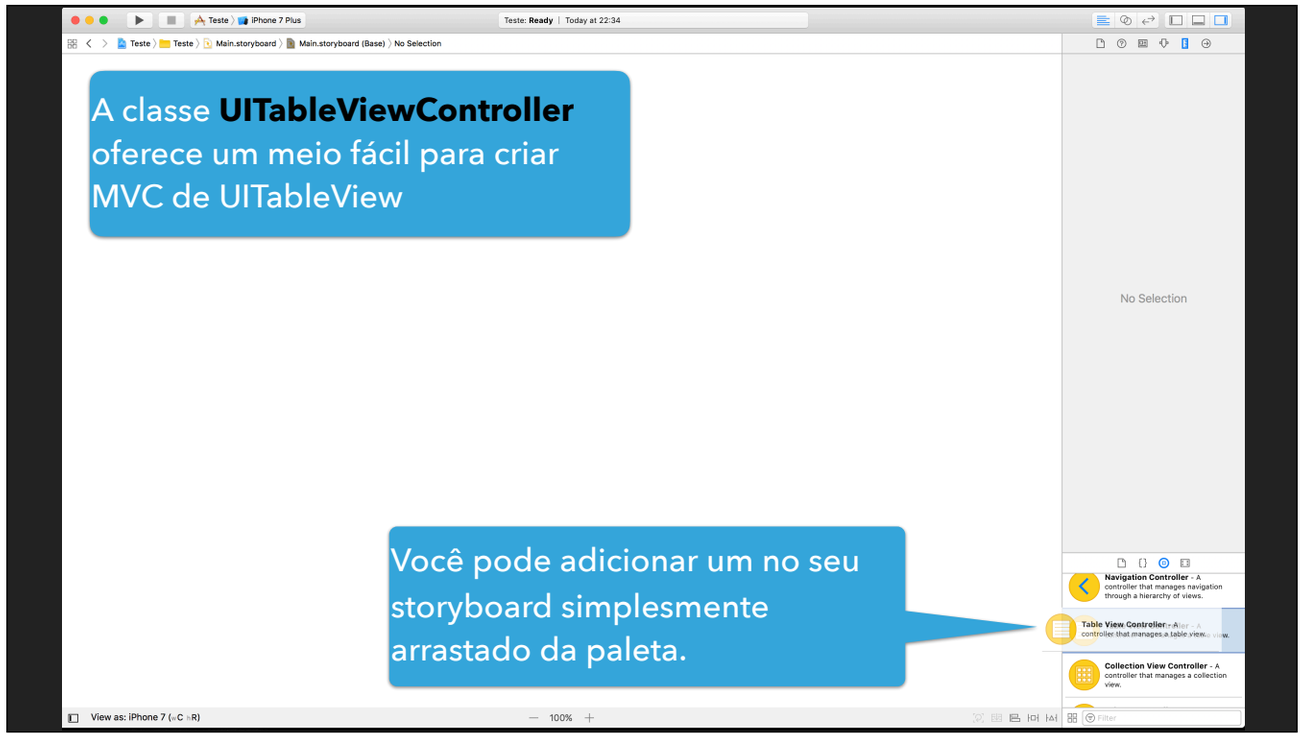


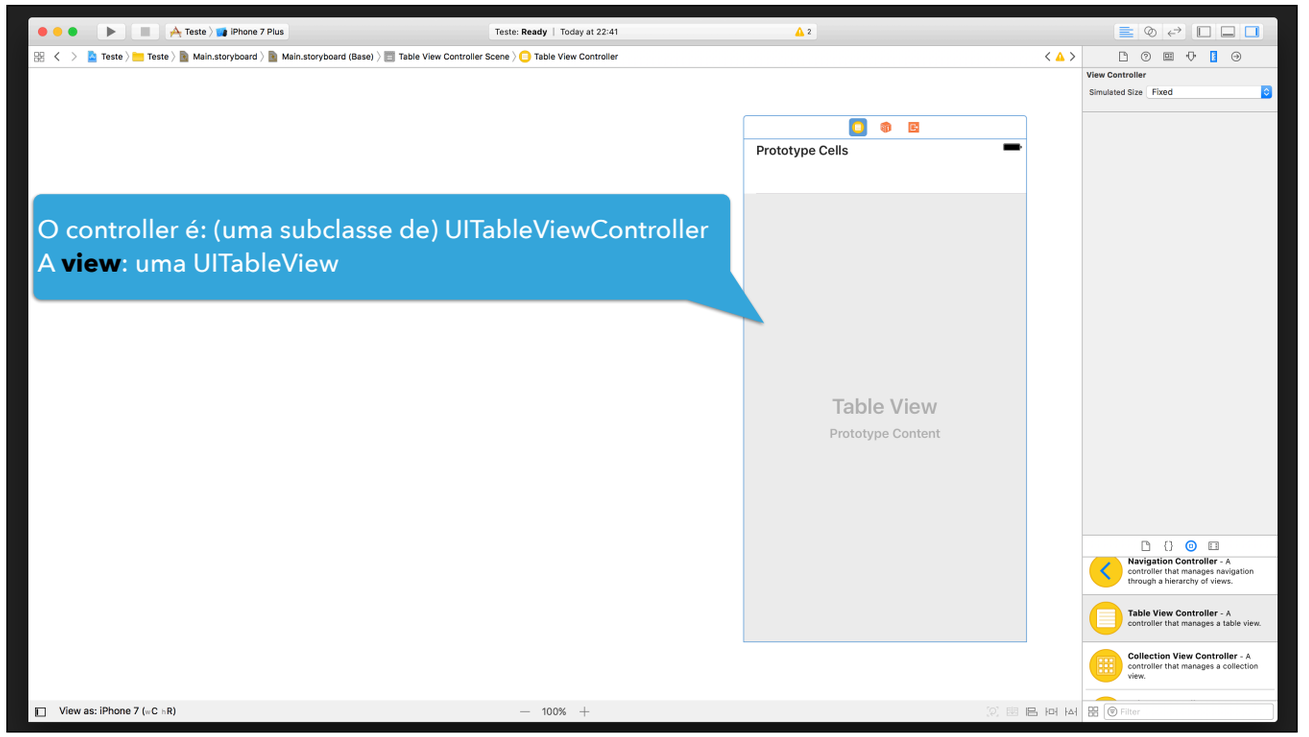
ONDE EU JÁ VI UMA TABLEVIEW CONTROLLER GROUPED?

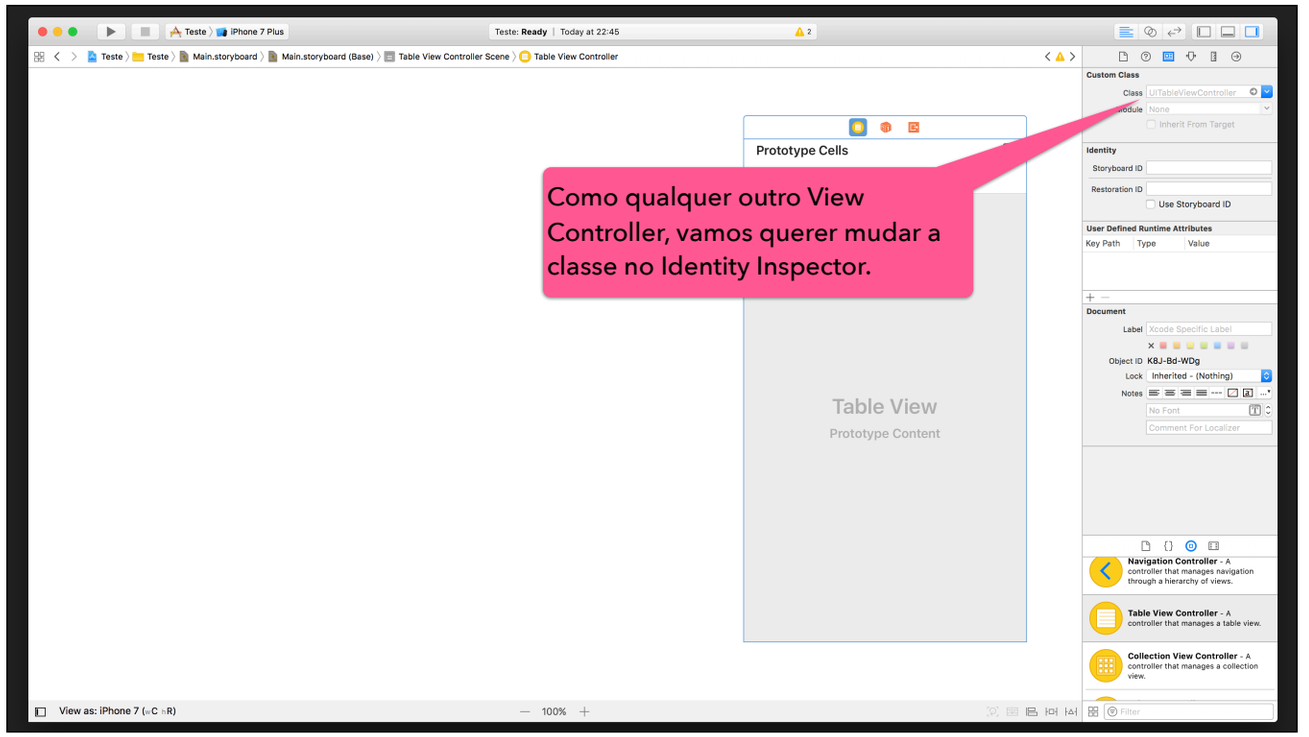


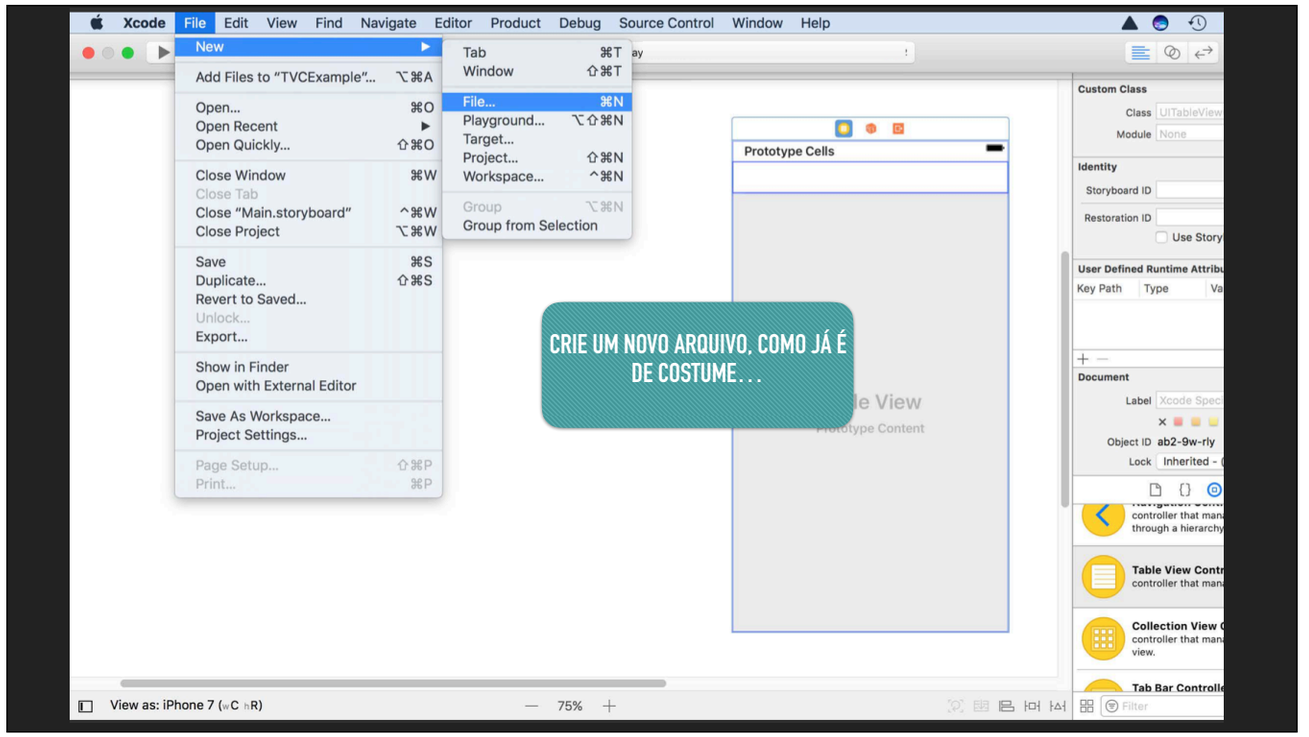
A classe **UITableViewController** oferece um meio fácil para criar MVC de UITableView

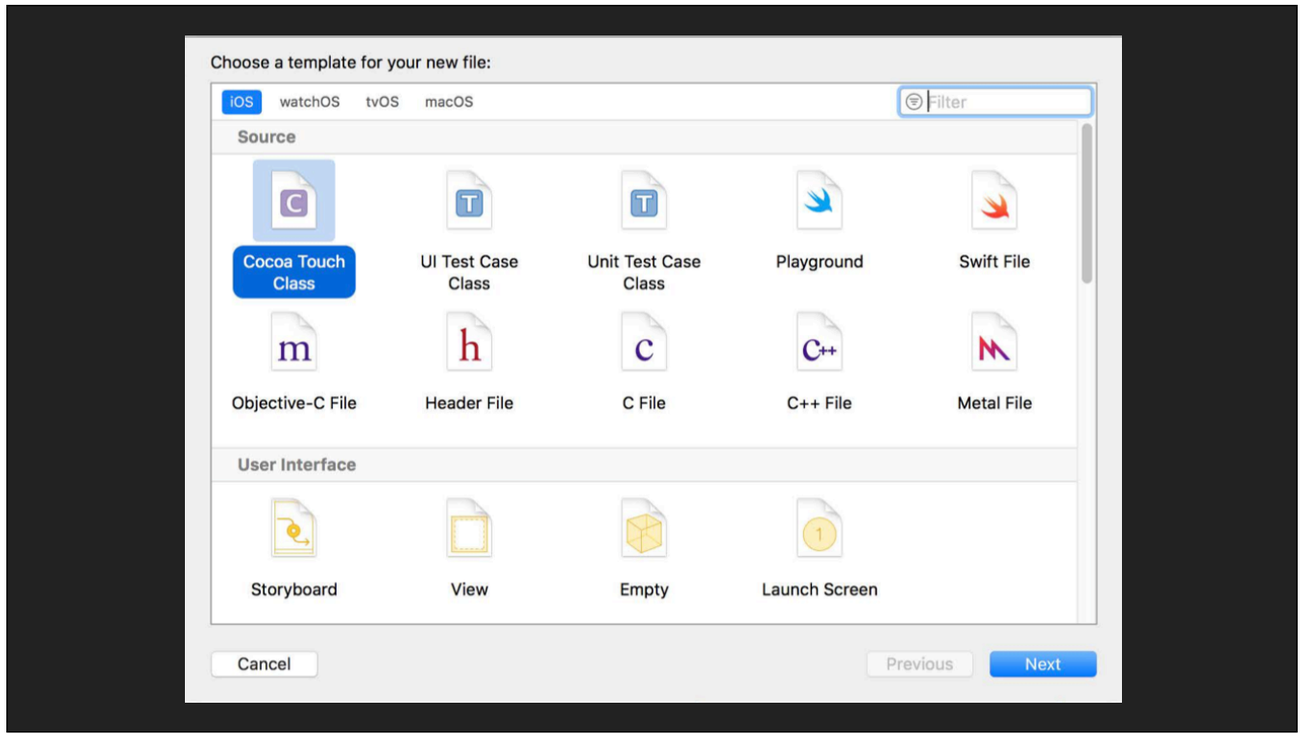

Teste 〉 iPhone 7 Plus
Teste: Ready | Today at 22:34
Teste 〉 Teste 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
A classe **UITableViewController** oferece um meio fácil para criar MVC de UITableView
No Selection
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
View as: iPhone 7 (wC hR)
100%
Filter

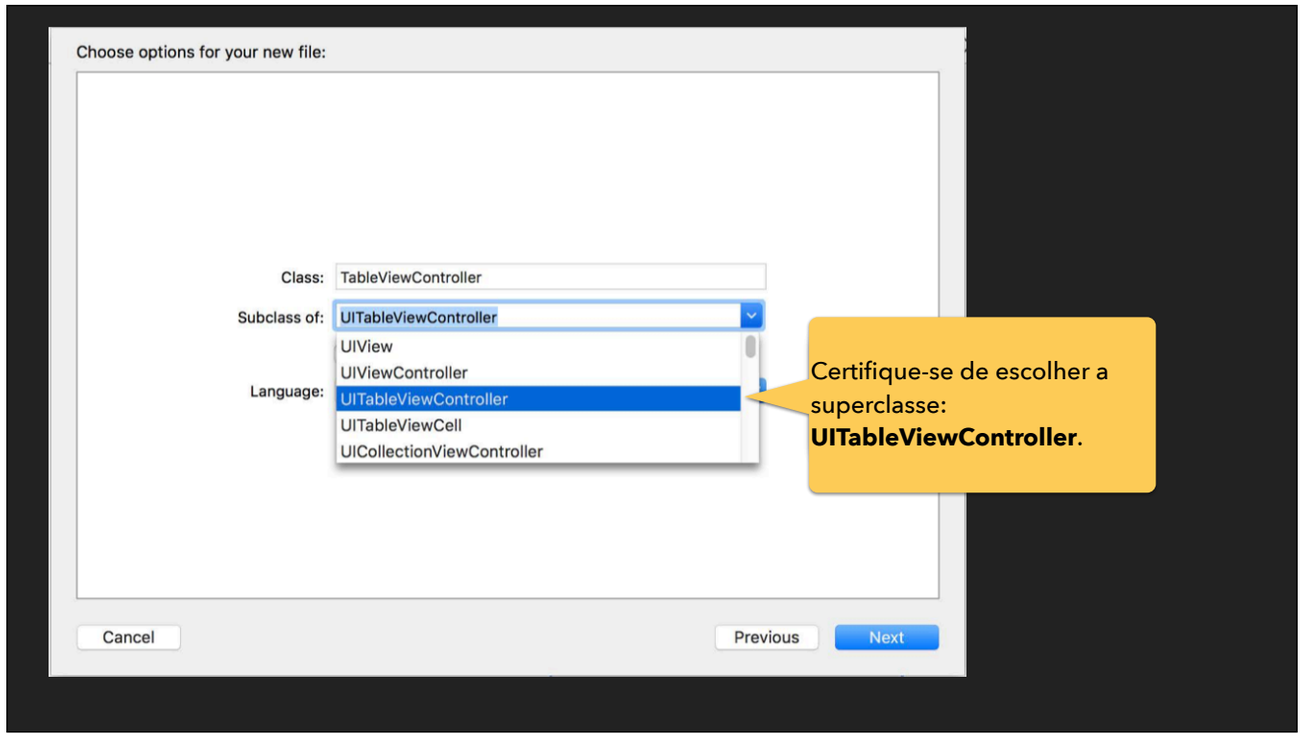

A classe **UITableViewController** oferece um meio fácil para criar MVC de UITableView
Você pode adicionar um no seu storyboard simplesmente arrastado da paleta.
No Selection
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.

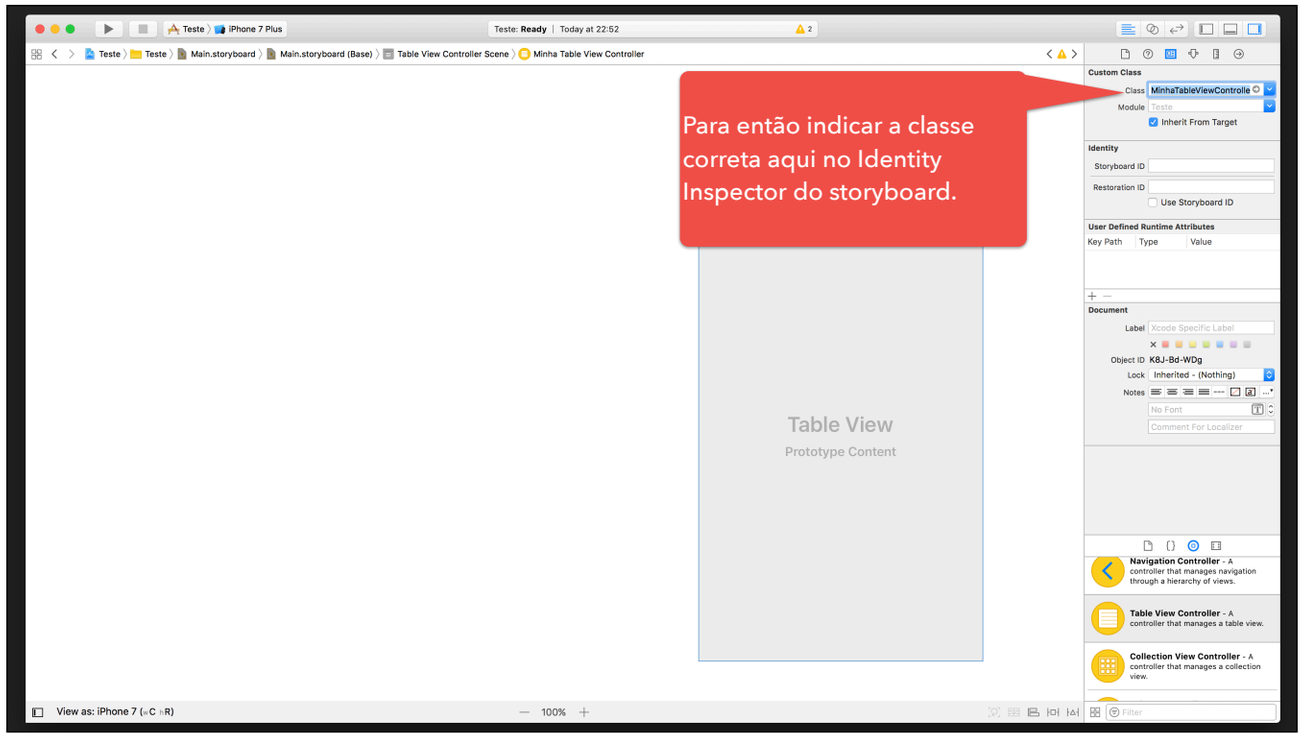

Teste
iPhone 7 Plus
Teste: Ready | Today at 22:41
Teste
Teste
Main.storyboard
Main.storyboard (Base)
Table View Controller Scene
Table View Controller
O controller é: (uma subclasse de) UITableViewController
A **view**: uma UITableView
Prototype Cells
Table View
Prototype Content
View Controller
Simulated Size
Fixed
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
View as: iPhone 7
100%

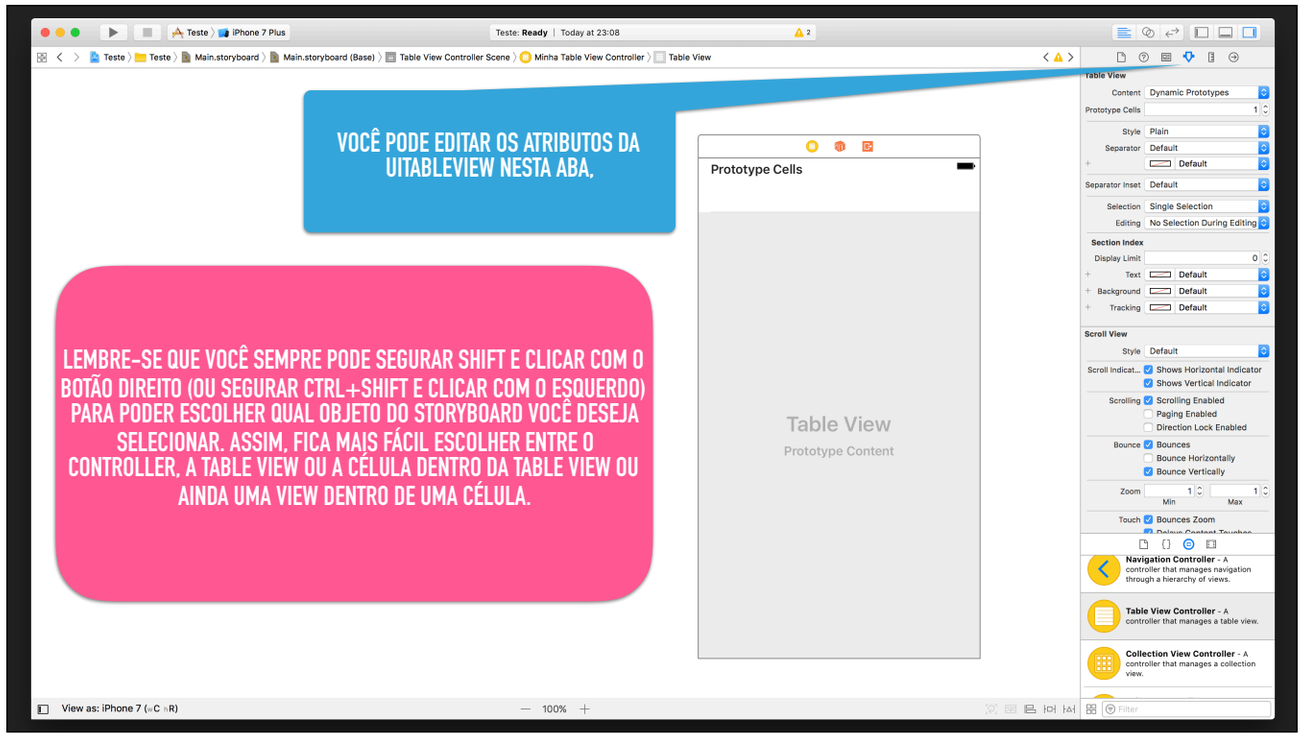

Como qualquer outro View Controller, vamos querer mudar a classe no Identity Inspector.
Prototype Cells
Table View
Prototype Content
Custom Class
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.

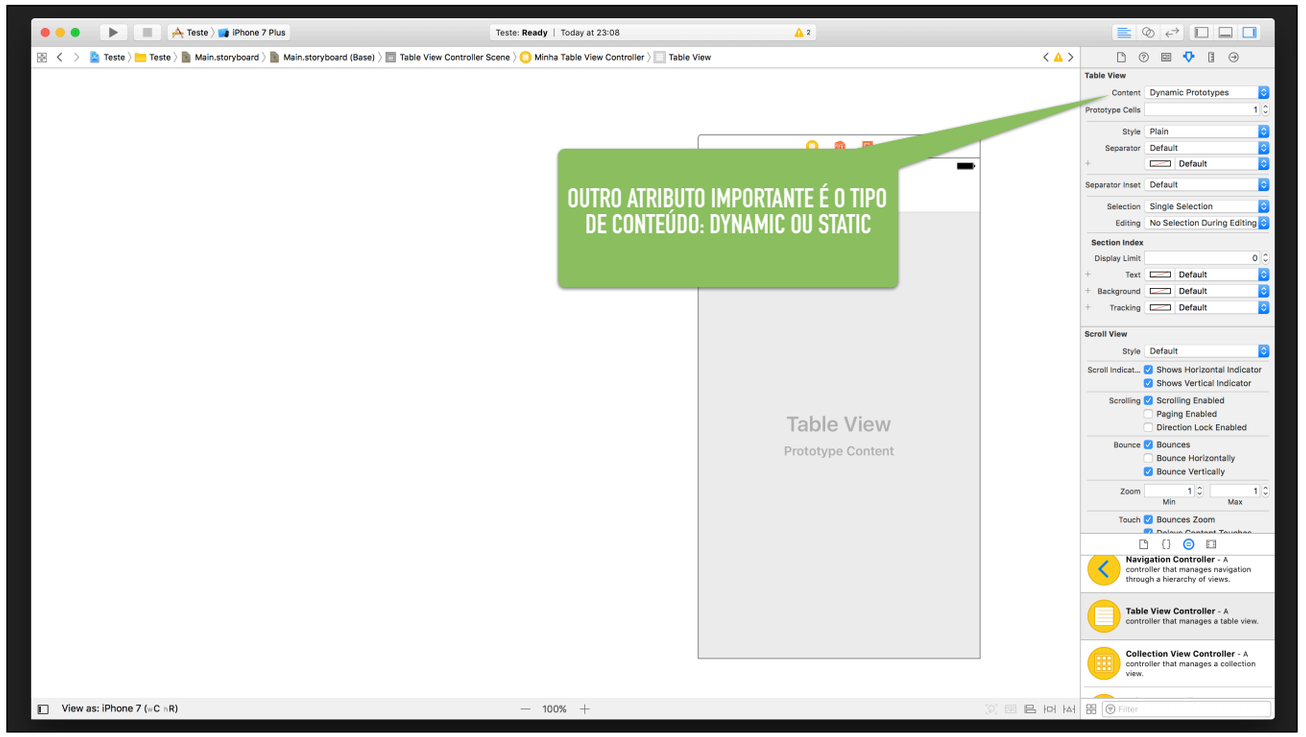

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
New
Add Files to "TVCExample"... ⌥⌘A
Open... ⌘O
Open Recent
Open Quickly... ⇧⌘O
Close Window ⌘W
Close Tab
Close "Main.storyboard" ^⌘W
Close Project ⌥⌘W
Save ⌘S
Duplicate... ⇧⌘S
Revert to Saved...
Unlock...
Export...
Show in Finder
Open with External Editor
Save As Workspace...
Project Settings...
Page Setup... ⇧⌘P
Print... ⌘P
Tab ⌘T
Window ⇧⌘T
File... ⌘N
Playground... ⌥⇧⌘N
Target...
Project... ⇧⌘N
Workspace... ^⌘N
Group ⌥⌘N
Group from Selection
Prototype Cells
CRIE UM NOVO ARQUIVO, COMO JÁ É DE COSTUME...
Custom Class
Identity
Storyboard ID
Restoration ID
User Defined Runtime Attributes
Key Path
Type
Document
Object ID ab2-9w-rly
Table View Controller
Collection View Controller
Tab Bar Controller
View as: iPhone 7 (wC hR)
75%


Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Playground
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
User Interface
Storyboard
View
Empty
Launch Screen
Cancel
Previous
Next


Choose options for your new file:
Class: TableViewController
Subclass of: UITableViewController
UIView
UIViewController
Language:
UITableViewController
UITableViewCell
UICollectionViewController
Certifique-se de escolher a superclasse: **UITableViewController**.
Cancel
Previous
Next


Para então indicar a classe correta aqui no Identity Inspector do storyboard.
Table View
Prototype Content
Custom Class
Class MinhaTableViewControlle
Module Teste
Inherit From Target
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path Type Value
Document
Label Xcode Specific Label
Object ID K8J-Bd-WDg
Lock Inherited - (Nothing)
Notes
No Font
Comment For Localizer
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
View as: iPhone 7 (wC hR)
100%


VOCÊ PODE EDITAR OS ATRIBUTOS DA UITABLEVIEW NESTA ABA.
LEMBRE-SE QUE VOCÊ SEMPRE PODE SEGURAR SHIFT E CLICAR COM O BOTÃO DIREITO (OU SEGURAR CTRL+SHIFT E CLICAR COM O ESQUERDO) PARA PODER ESCOLHER QUAL OBJETO DO STORYBOARD VOCÊ DESEJA SELECIONAR. ASSIM, FICA MAIS FÁCIL ESCOLHER ENTRE O CONTROLLER, A TABLE VIEW OU A CÉLULA DENTRO DA TABLE VIEW OU AINDA UMA VIEW DENTRO DE UMA CÉLULA.
Prototype Cells
Table View
Prototype Content
Table View
Content Dynamic Prototypes
Prototype Cells 1
Style Plain
Separator Default
Default
Separator Inset Default
Selection Single Selection
Editing No Selection During Editing
Section Index
Display Limit 0
Text Default
Background Default
Tracking Default
Scroll View
Style Default
Scroll Indicat... Shows Horizontal Indicator
Shows Vertical Indicator
Scrolling Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Bounce Bounces
Bounce Horizontally
Bounce Vertically
Zoom 1 1
Min Max
Touch Bounces Zoom
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
View as: iPhone 7
100%


OUTRO ATRIBUTO IMPORTANTE É O TIPO DE CONTEÚDO: DYNAMIC OU STATIC
Table View
Prototype Content
Table View
Content Dynamic Prototypes
Prototype Cells 1
Style Plain
Separator Default
Default
Separator Inset Default
Selection Single Selection
Editing No Selection During Editing
Section Index
Display Limit 0
Text Default
Background Default
Tracking Default
Scroll View
Style Default
Shows Horizontal Indicator
Shows Vertical Indicator
Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Bounces
Bounce Horizontally
Bounce Vertically
Zoom Min 1 Max 1
Bounces Zoom
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.

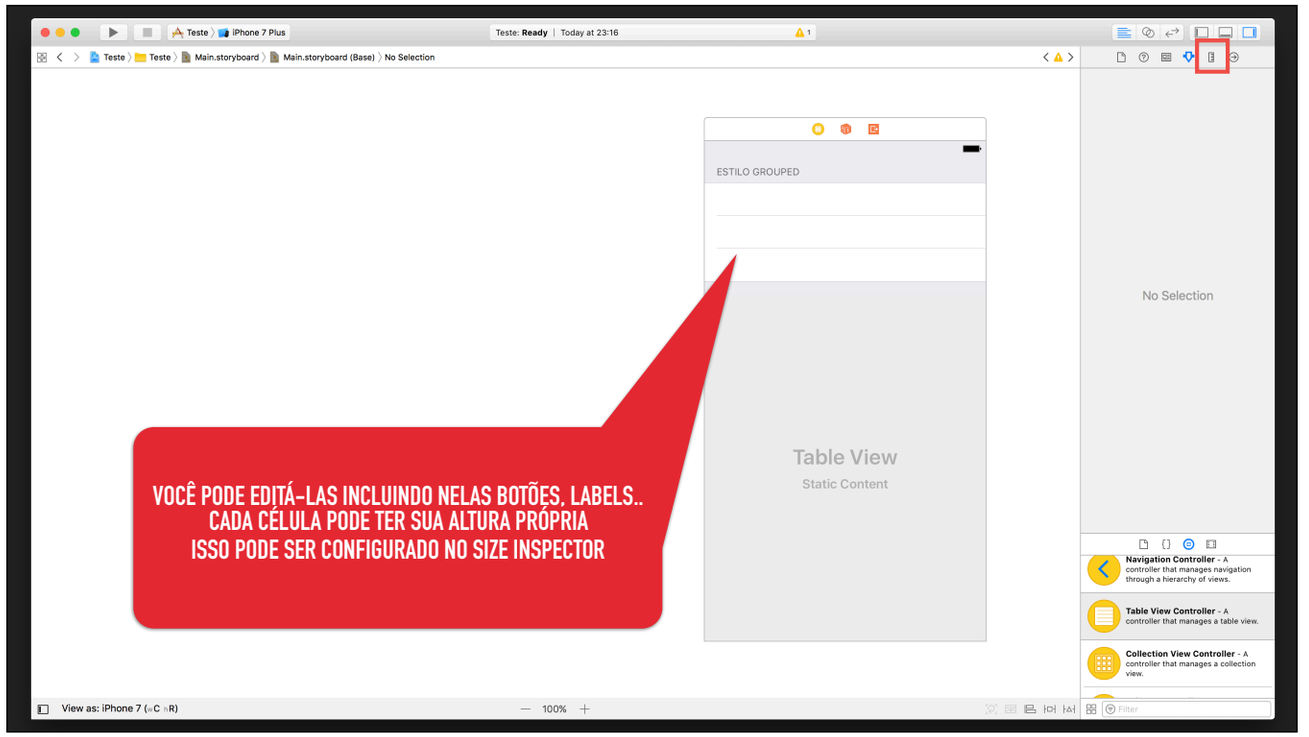
TABLEVIEW CONTROLLER

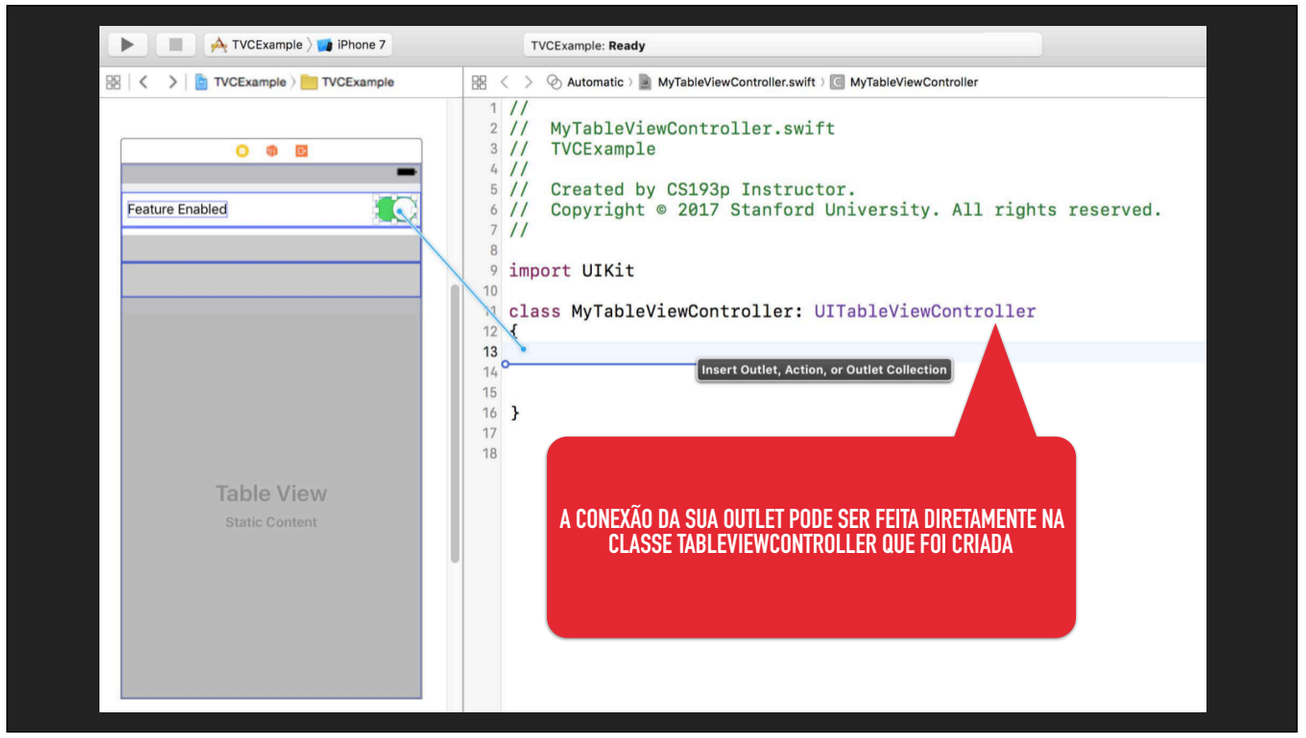
STATIC CELLS
OU
DYNAMIC
PROTOTYPES?

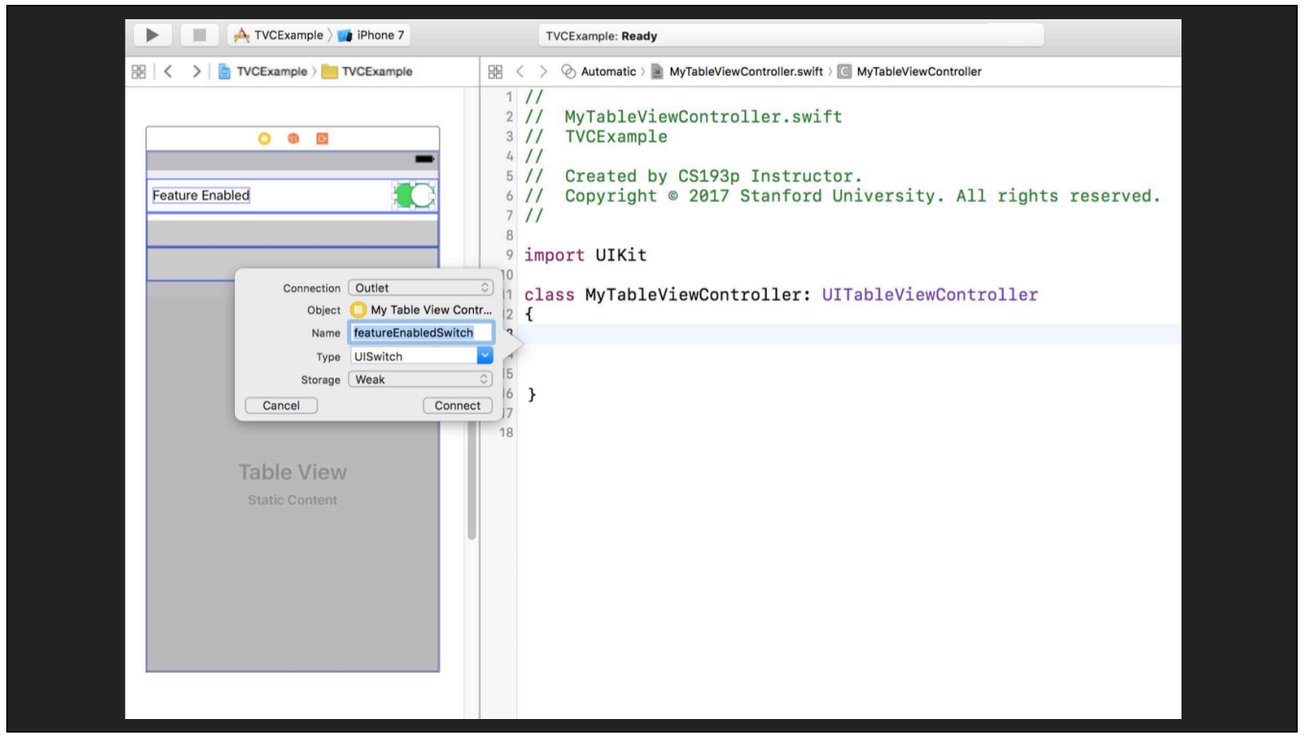
TABLEVIEW CONTROLLER

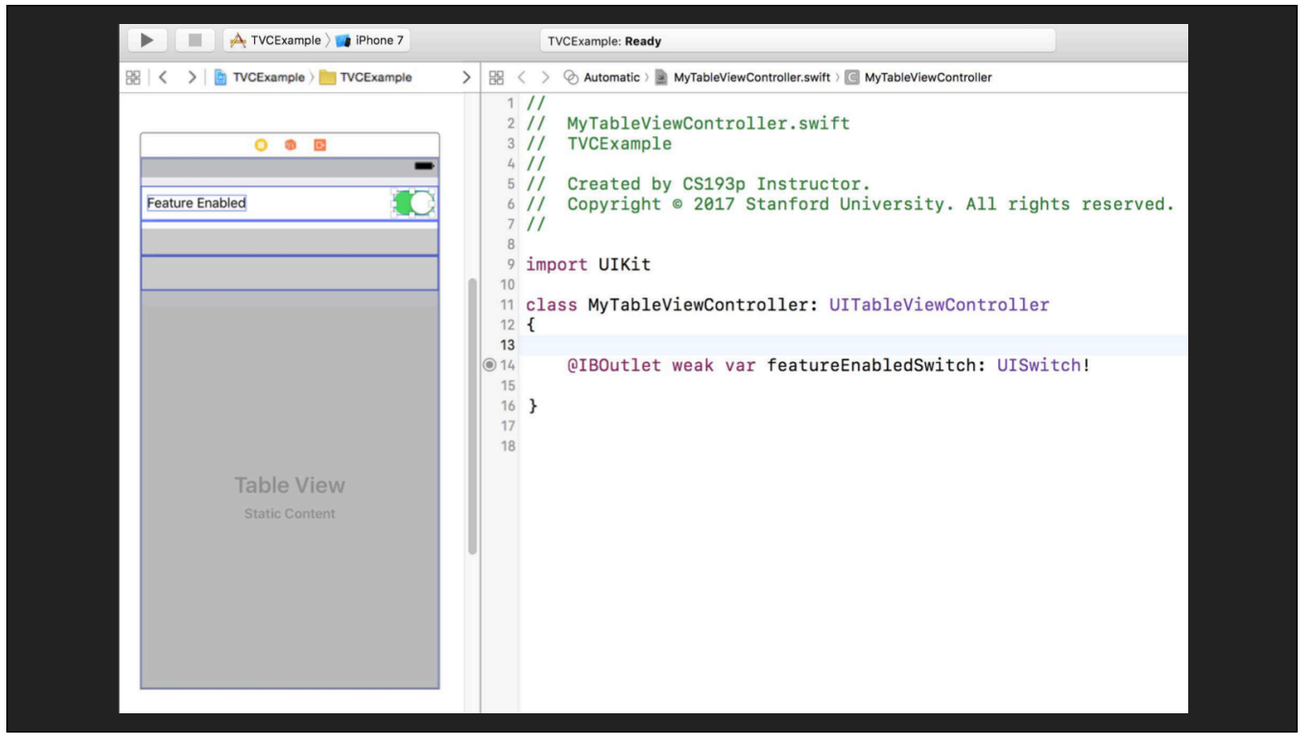
Static Cells:

- Servem para criar listas que você já sabe quantas células ela vai ter
- Muito usadas para configurações
- Podem fazer ligação diretamente do storyboard com o código
- Só existem para TableViewController, não para TableView

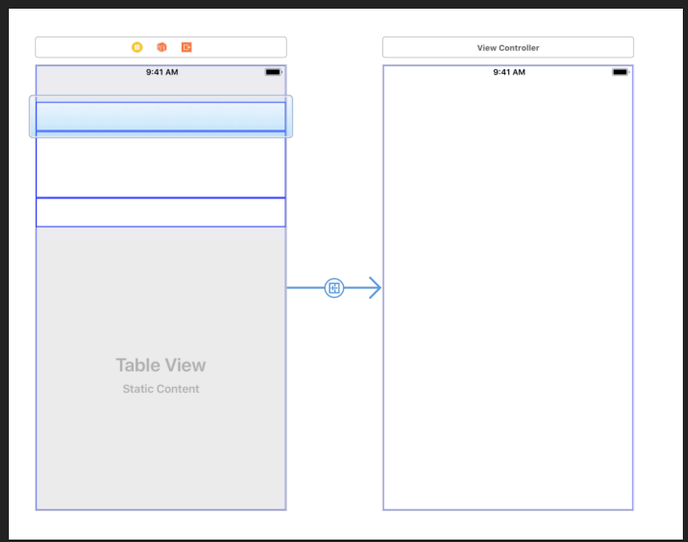

Teste 〉 iPhone 7 Plus
Teste: Ready | Today at 23:16
Teste 〉 Teste 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
ESTILO GROUPED
Table View
Static Content
VOCÊ PODE EDITÁ-LAS INCLUINDO NELAS BOTÕES, LABELS..
CADA CÉLULA PODE TER SUA ALTURA PRÓPRIA
ISSO PODE SER CONFIGURADO NO SIZE INSPECTOR
No Selection
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
View as: iPhone 7
100%


TVCExample 〉 iPhone 7
TVCExample: Ready
TVCExample 〉 TVCExample
Automatic 〉 MyTableViewController.swift 〉 MyTableViewController
Feature Enabled
Table View
Static Content
//
// MyTableViewController.swift
// TVCExample
//
// Created by CS193p Instructor.
// Copyright © 2017 Stanford University. All rights reserved.
//
import UIKit
class MyTableViewController: UITableViewController
{
Insert Outlet, Action, or Outlet Collection
}
A CONEXÃO DA SUA OUTLET PODE SER FEITA DIRETAMENTE NA CLASSE TABLEVIEWCONTROLLER QUE FOI CRIADA

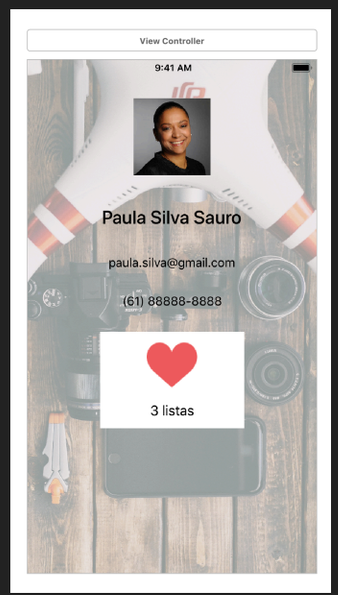

TVCExample
iPhone 7
TVCExample: Ready
TVCExample
TVCExample
Automatic
MyTableViewController.swift
MyTableViewController
Feature Enabled
Connection
Outlet
Object
My Table View Contr...
Name
featureEnabledSwitch
Type
UISwitch
Storage
Weak
Cancel
Connect
Table View
Static Content
//
// MyTableViewController.swift
// TVCExample
//
// Created by CS193p Instructor.
// Copyright © 2017 Stanford University. All rights reserved.
//
import UIKit
class MyTableViewController: UITableViewController
{
}

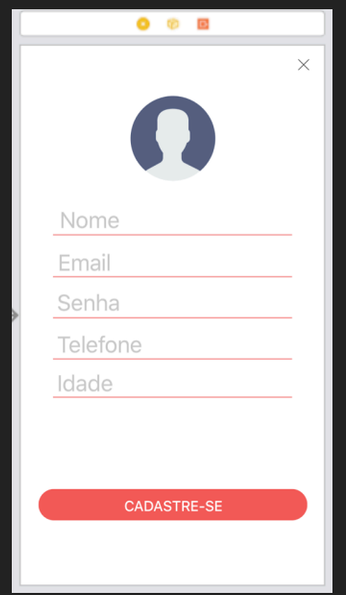

TVCExample
iPhone 7
TVCExample: Ready
TVCExample
TVCExample
Automatic
MyTableViewController.swift
MyTableViewController
Feature Enabled
Table View
Static Content
//
// MyTableViewController.swift
// TVCExample
//
// Created by CS193p Instructor.
// Copyright © 2017 Stanford University. All rights reserved.
//
import UIKit
class MyTableViewController: UITableViewController
{
    @IBOutlet weak var featureEnabledSwitch: UISwitch!
}

Outra coisa interessante é que conseguimos criar segues a partir da TableViewCell



VAMOS AJUSTAR NOSSO PROJETO

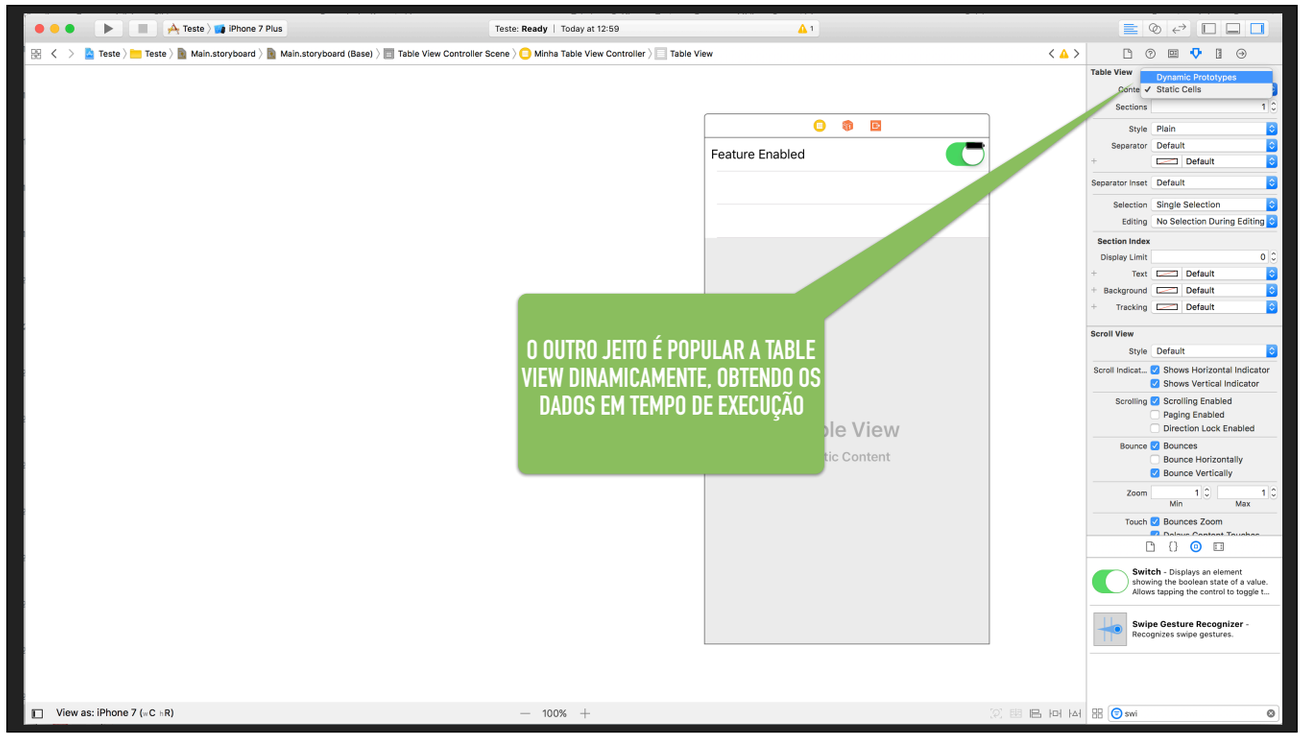


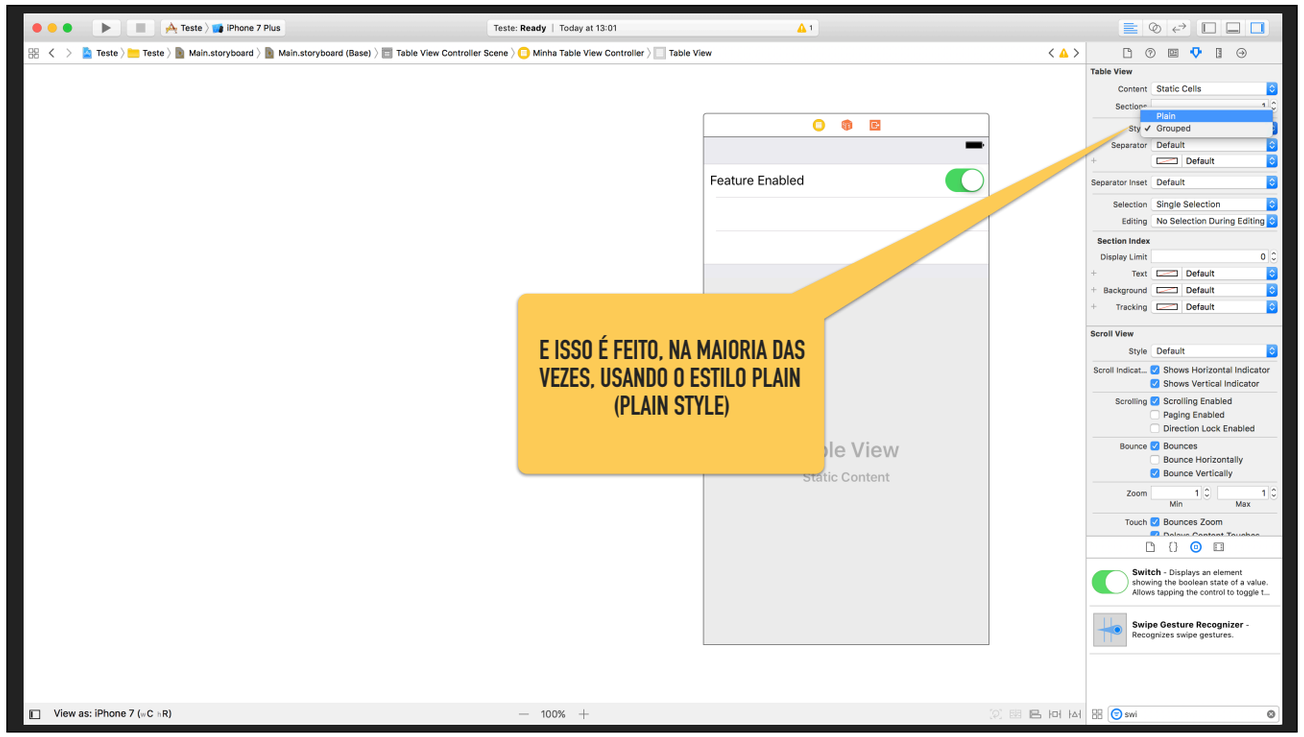


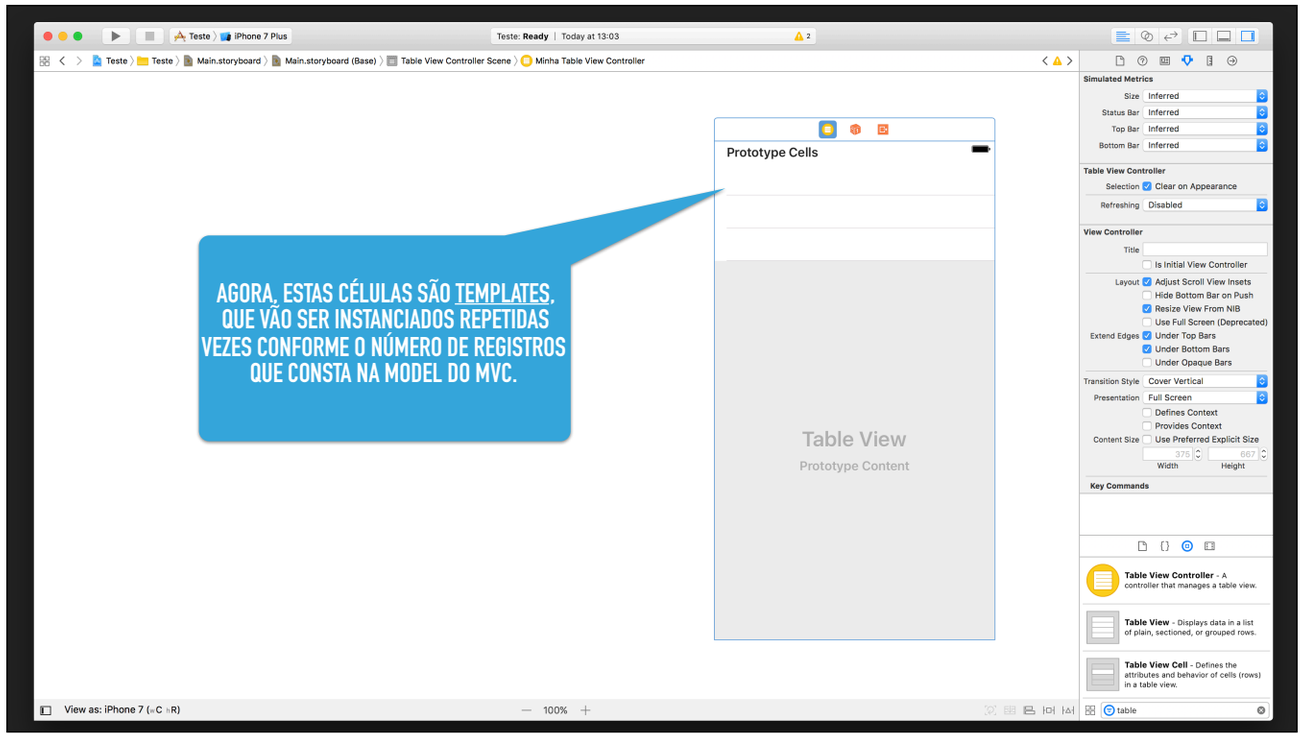
TABLEVIEW CONTROLLER

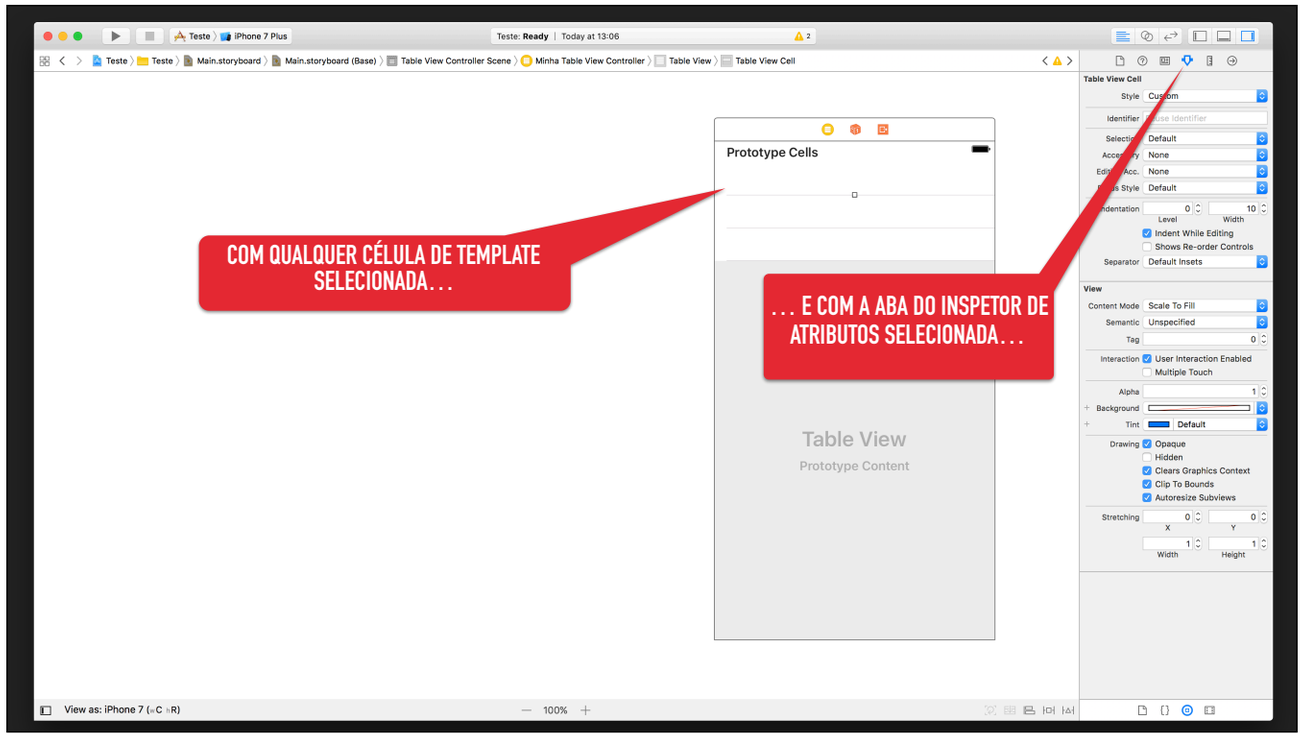
Dynamic Prototypes:

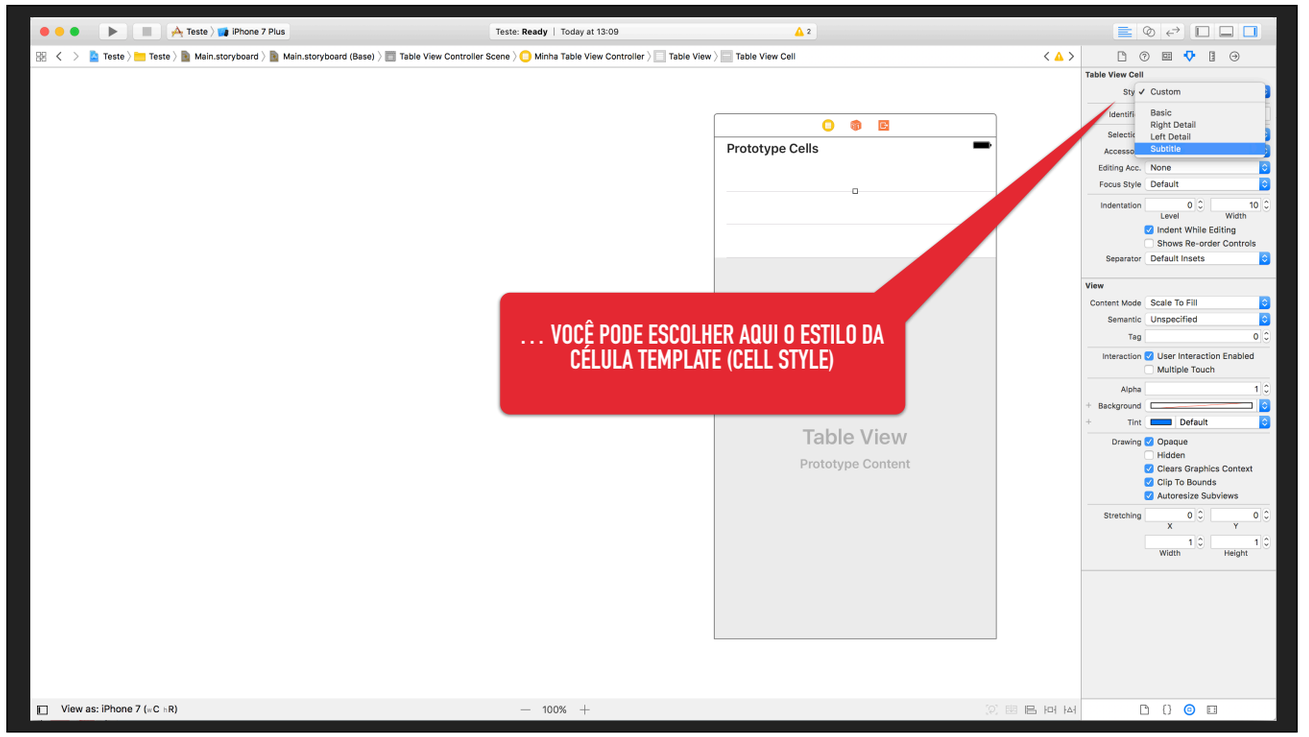
- Servem para criar listas que você não sabe quantas células ela vai ter
- Muito usadas para exibir informações vindas do servidor
- Não podem fazer ligação diretamente do storyboard com o código
- Existem tanto para a TableViewController, quanto para a TableView
- Cada célula funciona como um template a ser preenchido. Vão ter a mesma forma, porém informações diferentes
- É possível ter mais de um template de célula


Feature Enabled
O OUTRO JEITO É POPULAR A TABLE VIEW DINAMICAMENTE, OBTENDO OS DADOS EM TEMPO DE EXECUÇÃO


Feature Enabled
E ISSO É FEITO, NA MAIORIA DAS VEZES, USANDO O ESTILO PLAIN (PLAIN STYLE)
Static Content
Table View
Plain
Grouped
Scroll View
Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle t...
Swipe Gesture Recognizer - Recognizes swipe gestures.
View as: iPhone 7
100%

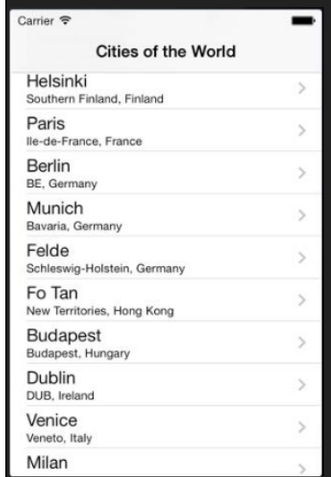

Teste
iPhone 7 Plus
Teste: Ready | Today at 13:03
Teste
Main.storyboard
Main.storyboard (Base)
Table View Controller Scene
Minha Table View Controller
AGORA, ESTAS CÉLULAS SÃO TEMPLATES, QUE VÃO SER INSTANCIADOS REPETIDAS VEZES CONFORME O NÚMERO DE REGISTROS QUE CONSTA NA MODEL DO MVC.
Prototype Cells
Table View
Prototype Content
View as: iPhone 7 (wC hR)
100%
Simulated Metrics
Size
Status Bar
Top Bar
Bottom Bar
Table View Controller
Selection
Clear on Appearance
Refreshing
Disabled
View Controller
Title
Is Initial View Controller
Layout
Adjust Scroll View Insets
Hide Bottom Bar on Push
Resize View From NIB
Use Full Screen (Deprecated)
Extend Edges
Under Top Bars
Under Bottom Bars
Under Opaque Bars
Transition Style
Cover Vertical
Presentation
Full Screen
Defines Context
Provides Context
Content Size
Use Preferred Explicit Size
Width
Height
Key Commands
Table View Controller - A controller that manages a table view.
Table View - Displays data in a list of plain, sectioned, or grouped rows.
Table View Cell - Defines the attributes and behavior of cells (rows) in a table view.
table


COM QUALQUER CÉLULA DE TEMPLATE SELECIONADA...
... E COM A ABA DO INSPETOR DE ATRIBUTOS SELECIONADA...
Prototype Cells
Table View
Prototype Content


Teste
iPhone 7 Plus
Teste: Ready | Today at 13:09
Main.storyboard
Main.storyboard (Base)
Table View Controller Scene
Minha Table View Controller
Table View
Table View Cell
Prototype Cells
Table View
Prototype Content
. . . VOCÊ PODE ESCOLHER AQUI O ESTILO DA CÉLULA TEMPLATE (CELL STYLE)
Table View Cell
Custom
Basic
Right Detail
Left Detail
Subtitle
Editing Acc.
None
Focus Style
Default
Indentation
Level
Width
Indent While Editing
Shows Re-order Controls
Separator
Default Insets
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
X
Y
Width
Height
View as: iPhone 7
100%

TIPOS DE CÉLULA PRÉ-DEFINIDOS (CELL STYLE)

TIPOS DE CÉLULA PRÉ-DEFINIDOS (CELL STYLE)

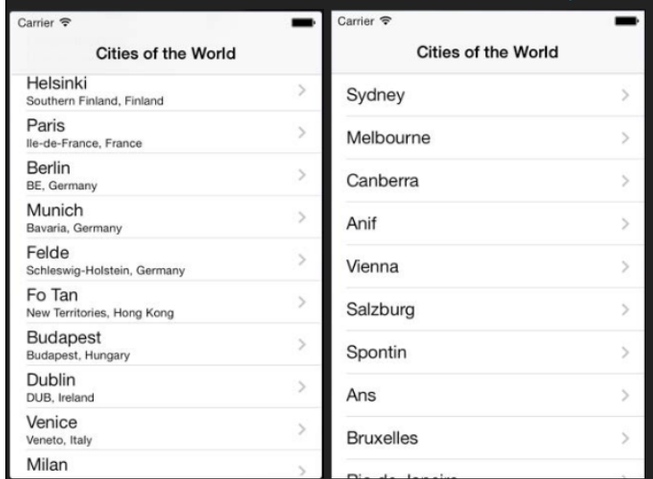


Subtitle

UITableViewCellStyle.subtitle

CADA UM NO SEU QUADRADO

TIPOS DE CÉLULA PRÉ-DEFINIDOS (CELL STYLE)

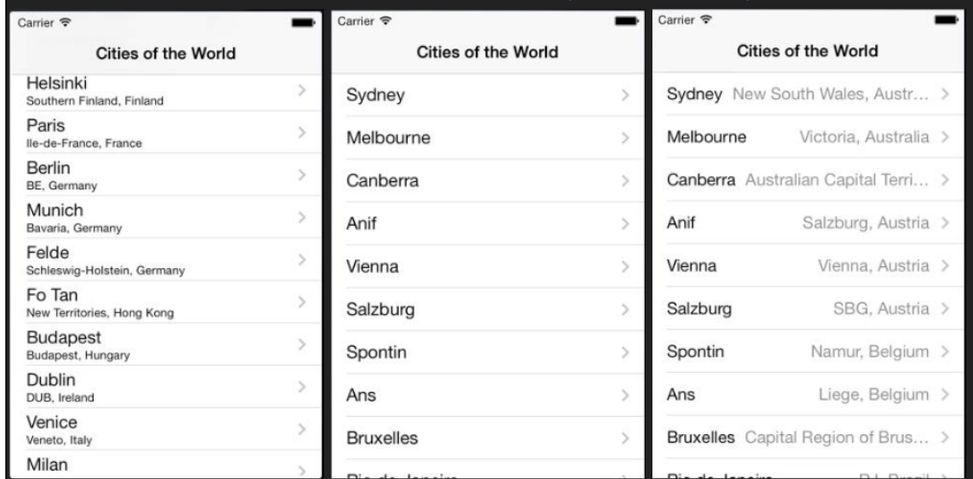


Subtitle
UITableViewCellStyle.subtitle

Basic
.basic

CADA UM NO SEU QUADRADO

TIPOS DE CÉLULA PRÉ-DEFINIDOS (CELL STYLE)



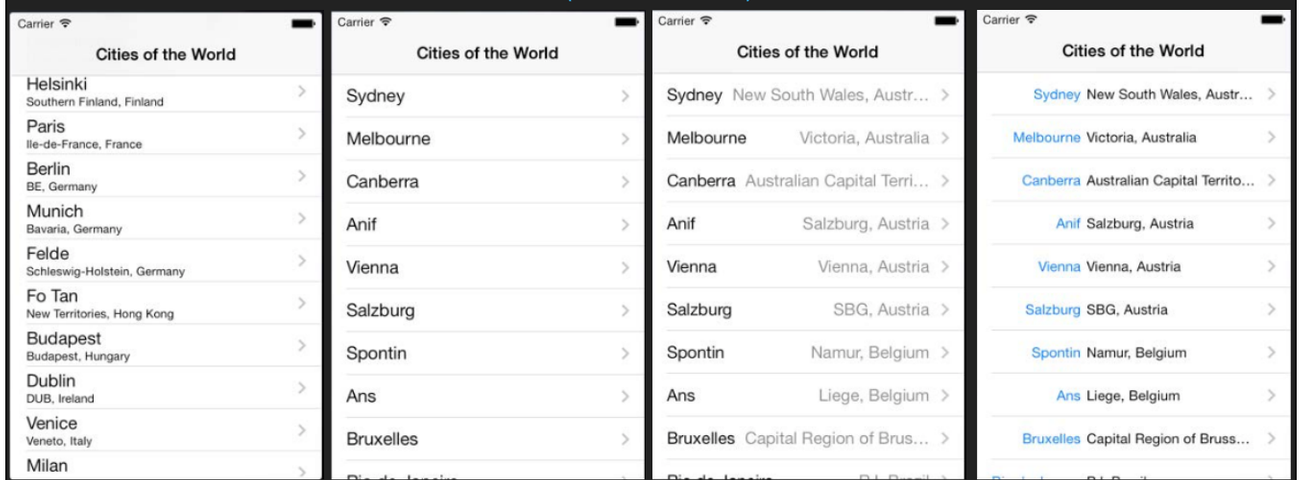
Subtitle
UITableViewCellStyle.subtitle

Basic
.basic

Right Detail
.value1

CADA UM NO SEU QUADRADO

TIPOS DE CÉLULA PRÉ-DEFINIDOS (CELL STYLE)

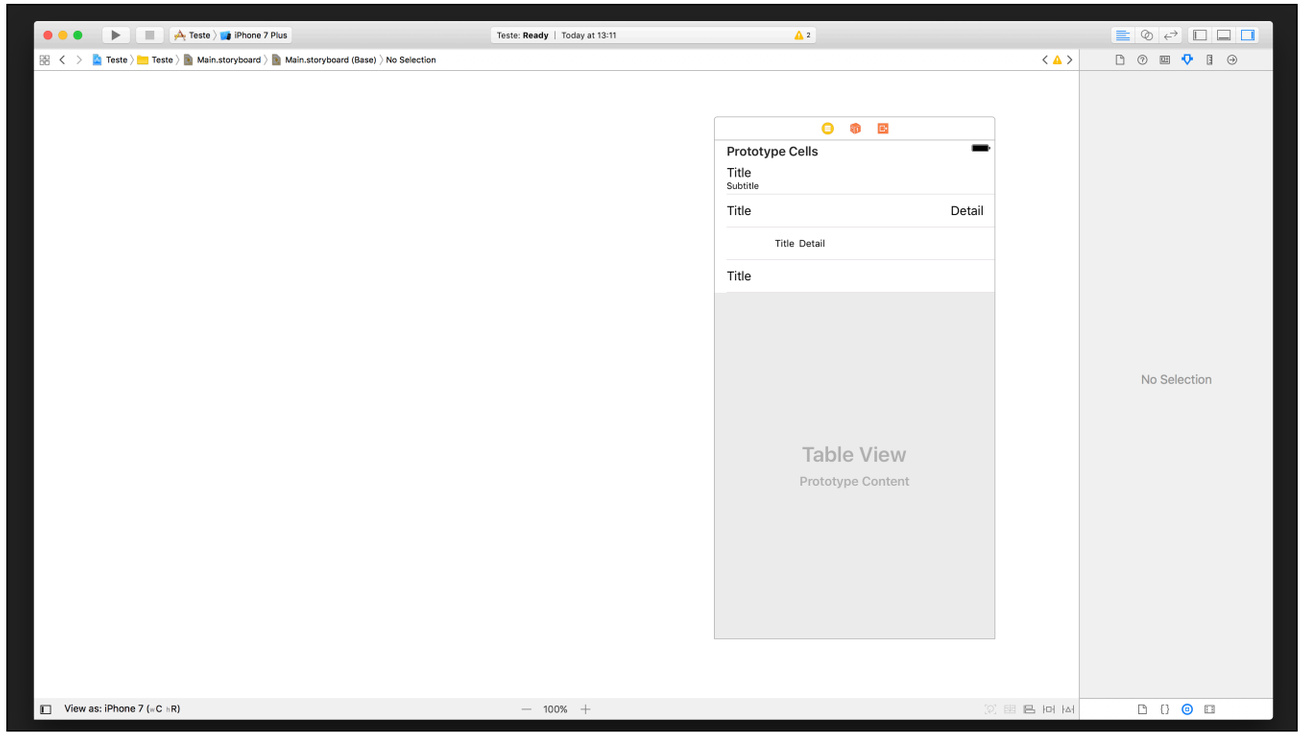


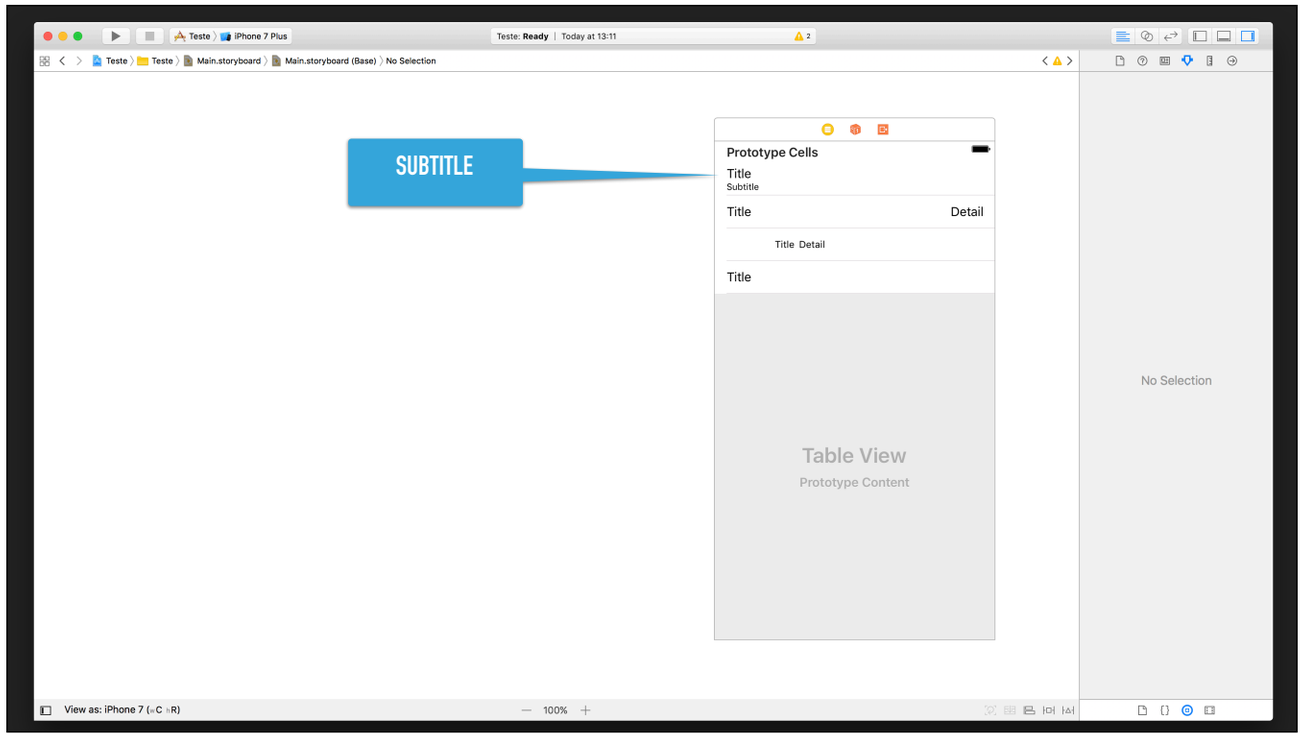
Subtitle
UITableViewCellStyle.subtitle

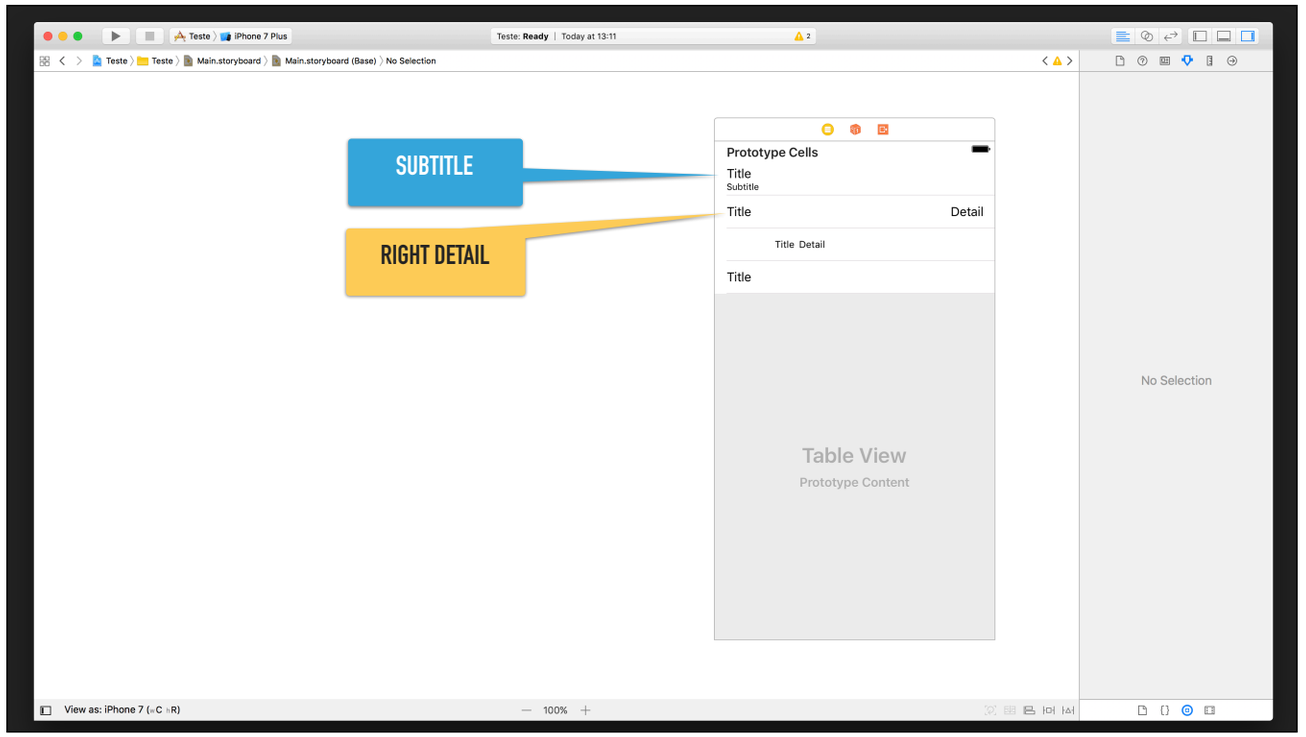
Basic
.basic

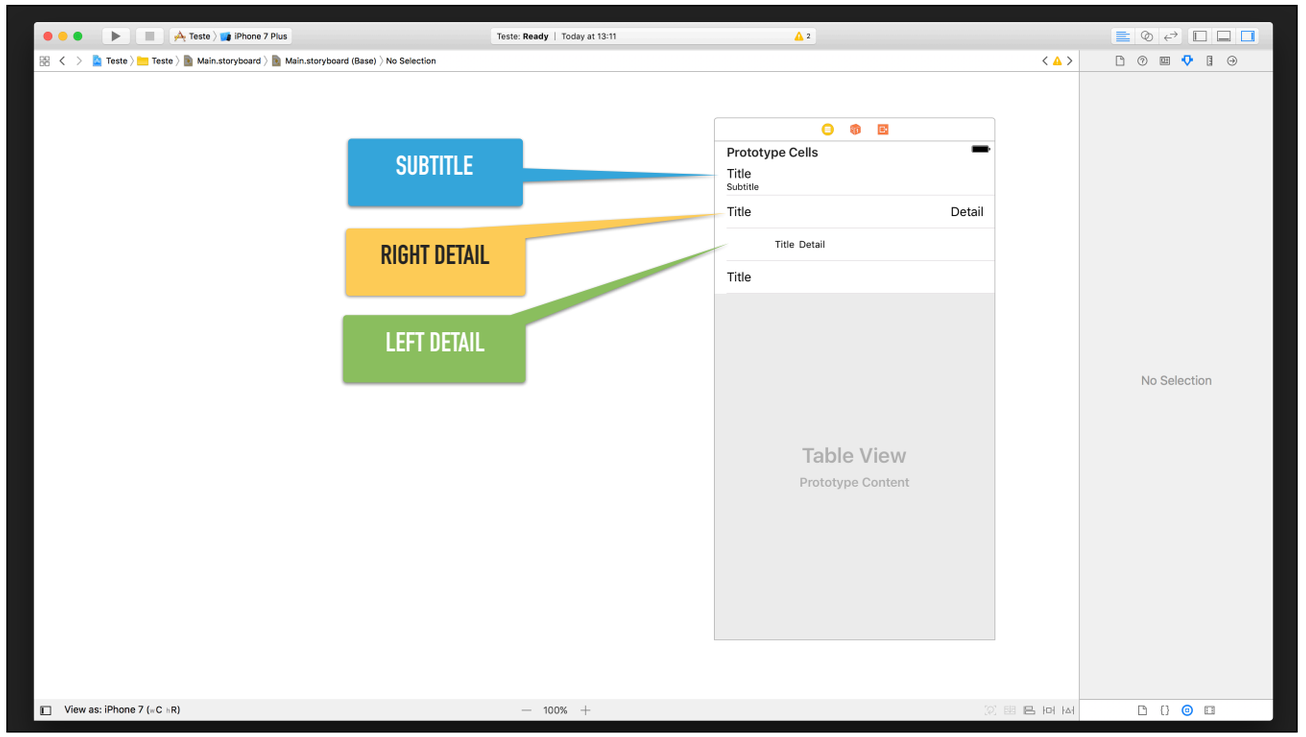
Right Detail
.value1

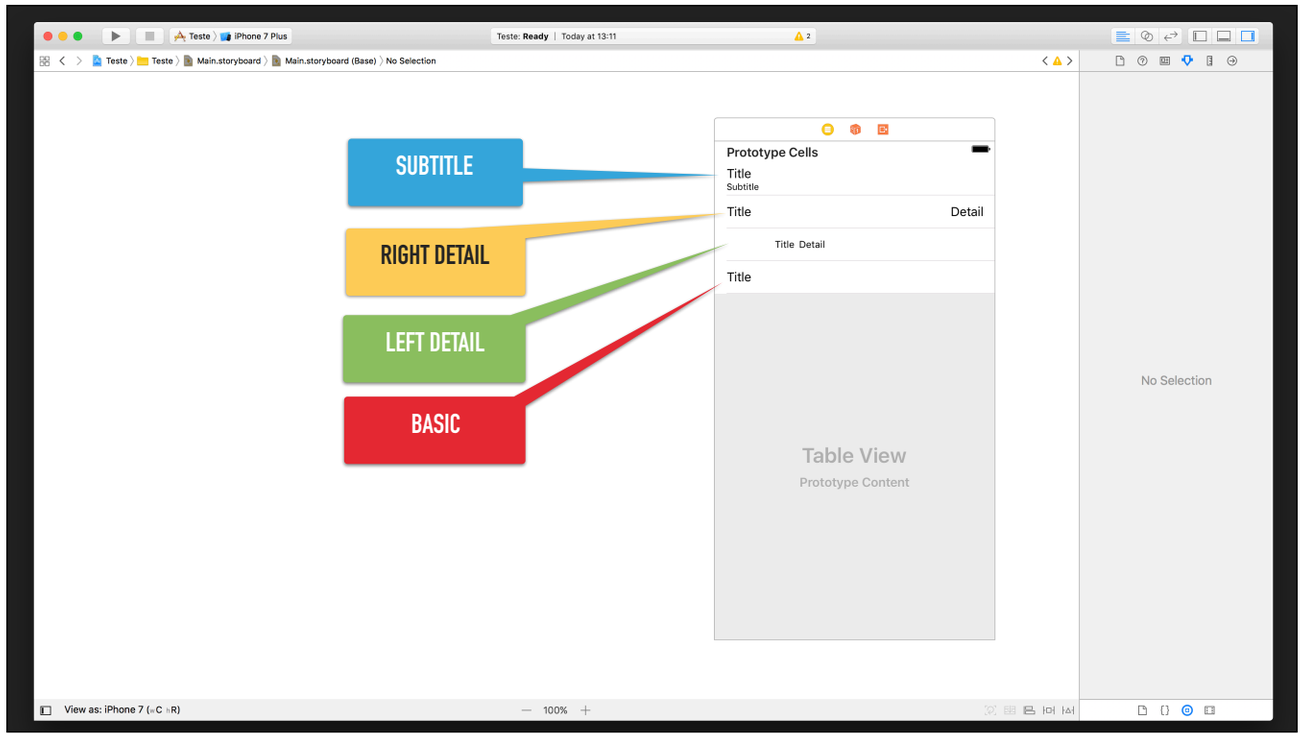
Left Detail
.value2

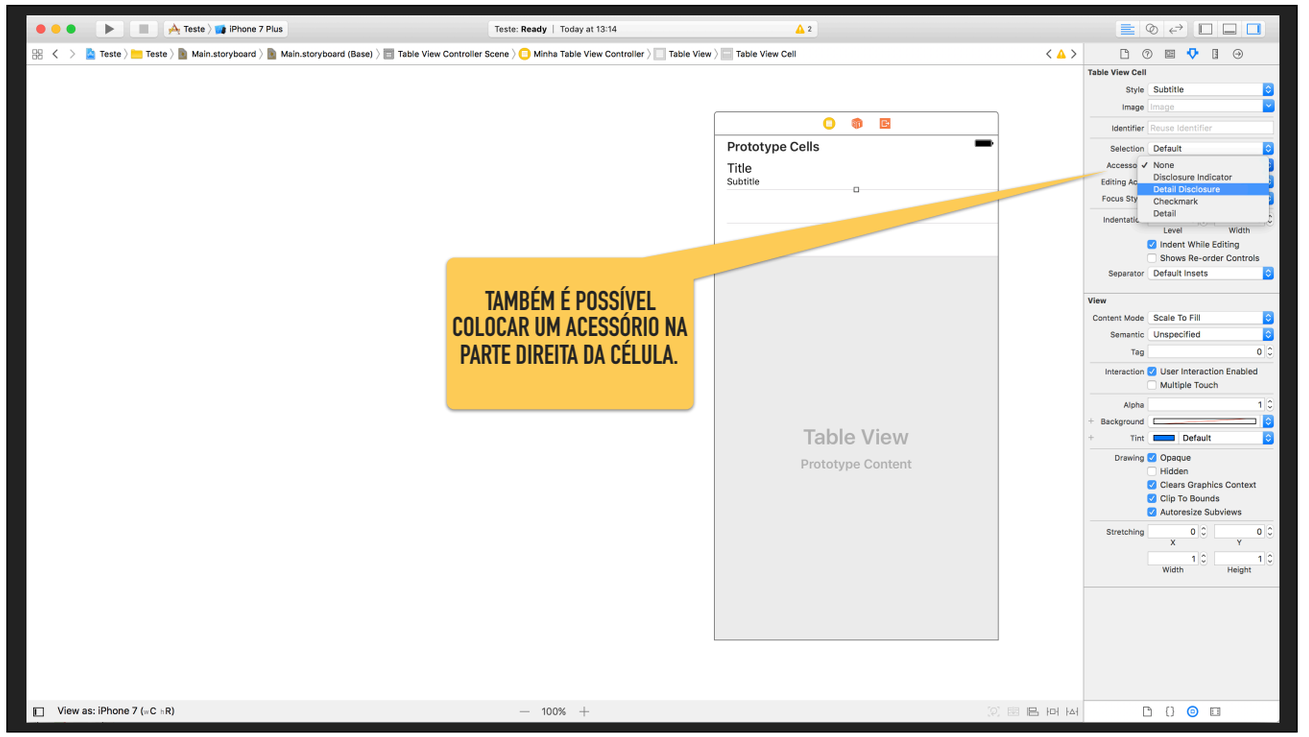

Teste
iPhone 7 Plus
Teste: Ready | Today at 13:11
Teste
Teste
Main.storyboard
Main.storyboard (Base)
No Selection
Prototype Cells
Title
Subtitle
Title
Detail
Title Detail
Title
Table View
Prototype Content
No Selection
View as: iPhone 7
100%


Teste
iPhone 7 Plus
Teste: Ready | Today at 13:11
Teste
Teste
Main.storyboard
Main.storyboard (Base)
No Selection
SUBTITLE
Prototype Cells
Title
Subtitle
Title
Detail
Title Detail
Title
Table View
Prototype Content
No Selection
View as: iPhone 7
100%


Teste
iPhone 7 Plus
Teste: Ready | Today at 13:11
Teste
Teste
Main.storyboard
Main.storyboard (Base)
No Selection
SUBTITLE
RIGHT DETAIL
Prototype Cells
Title
Subtitle
Title
Detail
Title Detail
Title
Table View
Prototype Content
No Selection
View as: iPhone 7 (wC hR)
100%


Teste
iPhone 7 Plus
Teste: Ready | Today at 13:11
Teste
Teste
Main.storyboard
Main.storyboard (Base)
No Selection
SUBTITLE
RIGHT DETAIL
LEFT DETAIL
Prototype Cells
Title
Subtitle
Title
Detail
Title Detail
Title
Table View
Prototype Content
No Selection
View as: iPhone 7
100%


Teste
iPhone 7 Plus
Teste: Ready | Today at 13:11
Teste
Teste
Main.storyboard
Main.storyboard (Base)
No Selection
SUBTITLE
RIGHT DETAIL
LEFT DETAIL
BASIC
Prototype Cells
Title
Subtitle
Title
Detail
Title Detail
Title
Table View
Prototype Content
No Selection
View as: iPhone 7
100%


Teste
iPhone 7 Plus
Teste: Ready | Today at 13:14
Teste
Teste
Main.storyboard
Main.storyboard (Base)
Table View Controller Scene
Minha Table View Controller
Table View
Table View Cell
Prototype Cells
Title
Subtitle
Table View
Prototype Content
TAMBÉM É POSSÍVEL COLOCAR UM ACESSÓRIO NA PARTE DIREITA DA CÉLULA.
Table View Cell
Style
Subtitle
Image
Identifier
Reuse Identifier
Selection
Default
None
Disclosure Indicator
Detail Disclosure
Checkmark
Detail
Level
Width
Indent While Editing
Shows Re-order Controls
Separator
Default Insets
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
X
Y
Width
Height
View as: iPhone 7
100%

TABLEVIEW CONTROLLER

SE EU NÃO POSSO CONECTAR MEUS ELEMENTOS DA CÉLULA COM O CÓDIGO VIA OUTLETS, COMO EU FAÇO?

TABLEVIEW CONTROLLER

SE EU NÃO POSSO CONECTAR MEUS ELEMENTOS DA CÉLULA COM O CÓDIGO VIA OUTLETS, COMO EU FAÇO?

Nossa UITableViewController é uma classe muito especial

TABLEVIEW CONTROLLER

SE EU NÃO POSSO CONECTAR MEUS ELEMENTOS DA CÉLULA COM O CÓDIGO VIA OUTLETS, COMO EU FAÇO?

Nossa UITableViewController é uma classe muito especial

Ela é como a UIViewController, só que com mais funcionalidades

TABLEVIEW CONTROLLER

SE EU NÃO POSSO CONECTAR MEUS ELEMENTOS DA CÉLULA COM O CÓDIGO VIA OUTLETS, COMO EU FAÇO?

Nossa UITableViewController é uma classe muito especial

Ela é como a UIViewController, só que com mais funcionalidades

Você notou que quando ela foi criada, ela já veio com código comentado?

TABLEVIEW CONTROLLER

SE EU NÃO POSSO CONECTAR MEUS ELEMENTOS DA CÉLULA COM O CÓDIGO VIA OUTLETS, COMO EU FAÇO?

Nossa UITableViewController é uma classe muito especial

Ela é como a UIViewController, só que com mais funcionalidades

Você notou que quando ela foi criada, ela já veio com código comentado?

```
// MARK: - Table view data source

override func numberOfSections(in tableView: UITableView) -> Int {
    // #warning Incomplete implementation, return the number of sections
    return 0
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection
    section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return 0
}

/*
override func tableView(_ tableView: UITableView, cellForRowAt indexPath:
    IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier:
        "reuseIdentifier", for: indexPath)

    // Configure the cell...

    return cell
}
*/

/*
// Override to support conditional editing of the table view.
override func tableView(_ tableView: UITableView, canEditRowAt indexPath:
    IndexPath) -> Bool {
    // Return false if you do not want the specified item to be editable.
    return true
}
*/

/*
// Override to support editing the table view.
override func tableView(_ tableView: UITableView, commit editingStyle:
    UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        // Delete the row from the data source
        tableView.deleteRows(at: [indexPath], with: .fade)
    } else if editingStyle == .insert {
        // Create a new instance of the appropriate class, insert it into
            the array, and add a new row to the table view
    }
}
*/
```

TABLEVIEW CONTROLLER

ELES SÃO DELEGATES!

A UITableViewController delega para nós implementarmos esses métodos.

Se não implementarmos, ela exibe uma TableView vazia

Vamos dar uma olhada em um método, o cellForRowAt:

```
tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
```

```
// MARK: - Table view data source

override func numberOfSections(in tableView: UITableView) -> Int {
    // #warning Incomplete implementation, return the number of sections
    return 0
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection
    section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return 0
}

/*
override func tableView(_ tableView: UITableView, cellForRowAt indexPath:
    IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier:
        "reuseIdentifier", for: indexPath)

    // Configure the cell...

    return cell
}
*/

/*
// Override to support conditional editing of the table view.
override func tableView(_ tableView: UITableView, canEditRowAt indexPath:
    IndexPath) -> Bool {
    // Return false if you do not want the specified item to be editable.
    return true
}
*/

/*
// Override to support editing the table view.
override func tableView(_ tableView: UITableView, commit editingStyle:
    UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        // Delete the row from the data source
        tableView.deleteRows(at: [indexPath], with: .fade)
    } else if editingStyle == .insert {
        // Create a new instance of the appropriate class, insert it into
            the array, and add a new row to the table view
    }
}
*/
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- A UIView que você vai fornecer para cada célula precisa ser subclasse de UITableViewCell (que, por sua vez, é subclasse de UIView);

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- A UIView que você vai fornecer para cada célula precisa ser subclasse de UITableViewCell (que, por sua vez, é subclasse de UIView);
- Somente as células que estiverem visíveis irão possuir uma instância dessa classe;

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- A UIView que você vai fornecer para cada célula precisa ser subclasse de UITableViewCell (que, por sua vez, é subclasse de UIView);
- Somente as células que estiverem visíveis irão possuir uma instância dessa classe;
- Isso significa, por outro lado, que as instâncias de UITableViewCell são **reutilizadas** à medida que células aparecem e desaparecem;

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- A UIView que você vai fornecer para cada célula precisa ser subclasse de UITableViewCell (que, por sua vez, é subclasse de UIView);
- Somente as células que estiverem visíveis irão possuir uma instância dessa classe;
- Isso significa, por outro lado, que as instâncias de UITableViewCell são **reutilizadas** à medida que células aparecem e desaparecem;
- Como é de se imaginar, isso tem complicadores para cenários multithread, então seja cuidadoso!

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- O indexPath é uma estrutura de dados que recebemos com a informação da tela que vamos desenhar;

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- O indexPath é uma estrutura de dados que recebemos com a informação da tela que vamos desenhar;
- As propriedades mais importantes são:

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- O indexPath é uma estrutura de dados que recebemos com a informação da tela que vamos desenhar;
- As propriedades mais importantes são:
  - indexPath.section: indica qual section está sendo criada

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- O indexPath é uma estrutura de dados que recebemos com a informação da tela que vamos desenhar;
- As propriedades mais importantes são:
  - indexPath.section: indica qual section está sendo criada
  - indexPath.row: indica qual a linha da section está sendo crianda

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Essa cell, já contém as properties que queremos acessar:
  - textLabel - é o título
  - detailTextLabel - é o detail/subtitle

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Essa cell, já contém as properties que queremos acessar:
  - textLabel - é o título
  - detailTextLabel - é o detail/subtitle
- Vamos preencher:

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Essa cell, já contém as properties que queremos acessar:
  - textLabel - é o título
  - detailTextLabel - é o detail/subtitle
- Vamos preencher:

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    cell.textLabel?.text = "Titulo"
    cell.detailTextLabel?.text = "Detalhe"
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

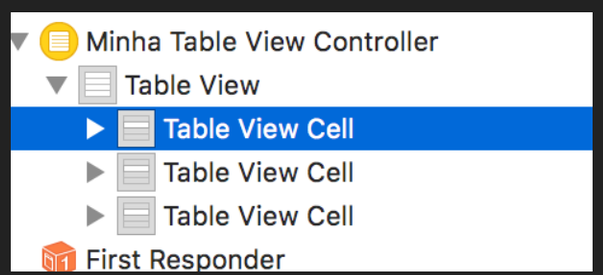
- Essa cell, pede um identifier
- Ele é definido nas propriedades das célula no storyboard

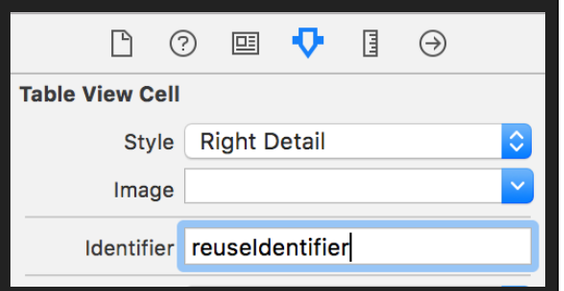
```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)
    // Configure the cell...
    cell.textLabel?.text = "Titulo"
    cell.detailTextLabel?.text = "Detalhe"
    return cell
}
```

TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Essa cell, pede um identifier
- Ele é definido nas propriedades das célula no storyboard
- Selecione a célula no Storyboard
- Preencha o Identifier da célula

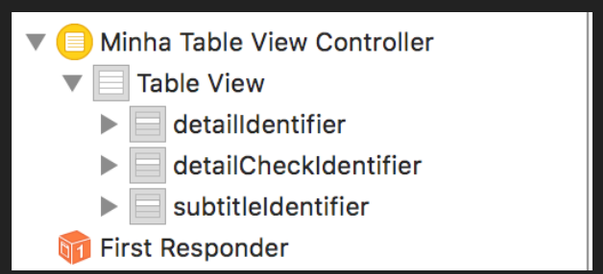


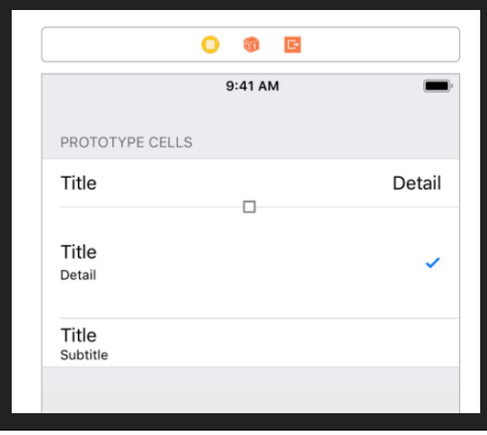


TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Como são templates, não faz sentido terem vários templetas iguais
- Muito menos utilizarem o mesmo identificador

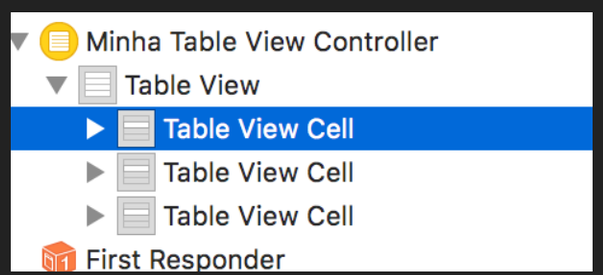


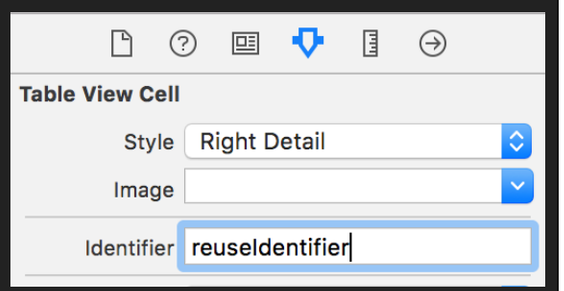


TABLEVIEW CONTROLLER

PROVENDO UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Essa cell, pede um identifier
- Ele é definido nas propriedades das célula no storyboard
- Selecione a célula no Storyboard
- Preencha o Identifier da célula



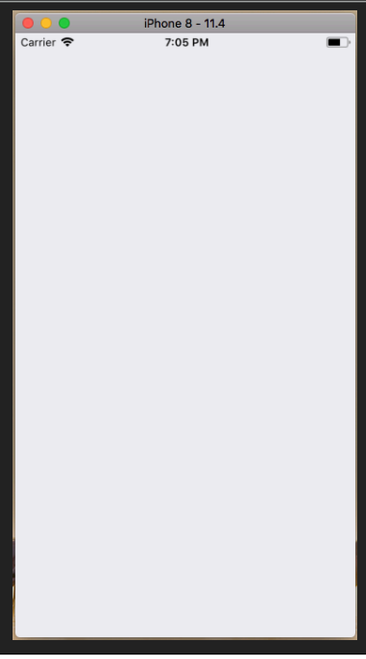


TABLEVIEW CONTROLLER

EXECUTEM

TABLEVIEW CONTROLLER

EXECUTEM



TABLEVIEW CONTROLLER

APRENDEMOS A ENTREGAR UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Quantas células devem ser desenhadas?
- Quantas sections tem?

TABLEVIEW CONTROLLER

APRENDEMOS A ENTREGAR UMA UIVIEW PARA DESENHAR CADA CÉLULA

- Quantas células devem ser desenhadas?
- Quantas sections tem?

```
override func numberOfSections(in tableView: UITableView) -> Int {
    // #warning Incomplete implementation, return the number of sections
    return 0
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return 0
}
```

TABLEVIEW CONTROLLER

- Estamos sobreescrevendo o método numberOfRowsInSection, se removermos, ele considera como return 1
- Estamos sobreescrevendo o método cellForRowAt, se removermos, ele considera como return 0

TABLEVIEW CONTROLLER

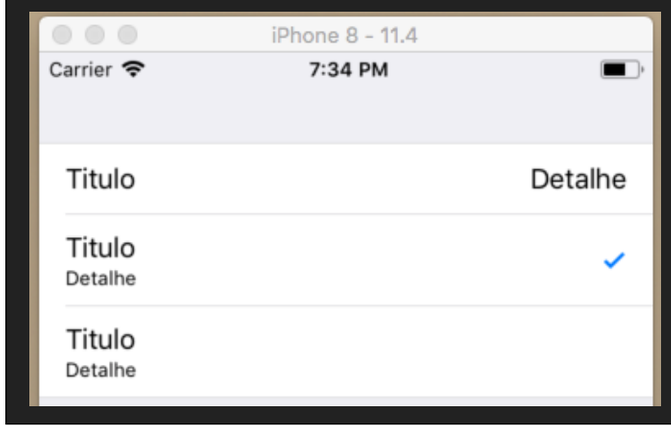
- Estamos sobreescrevendo o método numberOfRowsInSection, se removermos, ele considera como return 1
- Estamos sobreescrevendo o método cellForRowAt, se removermos, ele considera como return 0

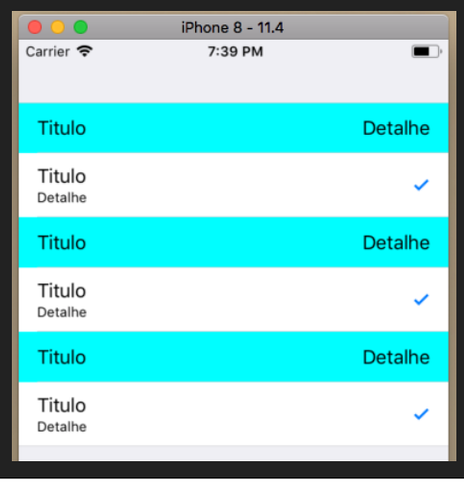
```
override func numberOfSections(in tableView: UITableView) -> Int {
    // #warning Incomplete implementation, return the number of sections
    return 0
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return 0
}
```

TABLEVIEW CONTROLLER

EXERCÍCIO





TABLEVIEW CONTROLLER

EXERCÍCIO

- Dica: indexPath