



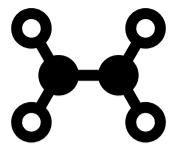
PAMELA RUÍZ

SWIFT

PRIMERA EDICIÓN



SWIFT	3
¿Que es swift?	3
Declaración de variables.....	3
Arreglos.....	3
Condicionales	3
Crear un proyecto	3
ok	



SWIFT

¿QUE ES SWIFT?

Swift es fuertemente tipado, esto quiere decir que si asignamos de alguna forma el tipo, no podremos cambiarlo posteriormente

DECLARACIÓN DE VARIABLES

Declaramos Int y String

```
var name = "Pamela"  
var variable:String = String()  
var variable1:String = "Hola"  
var variable2:Int = 90  
var variable3:Int = Int()
```

Todo funciona por clases, es por ello que el tipo de variable inicia con minúscula.

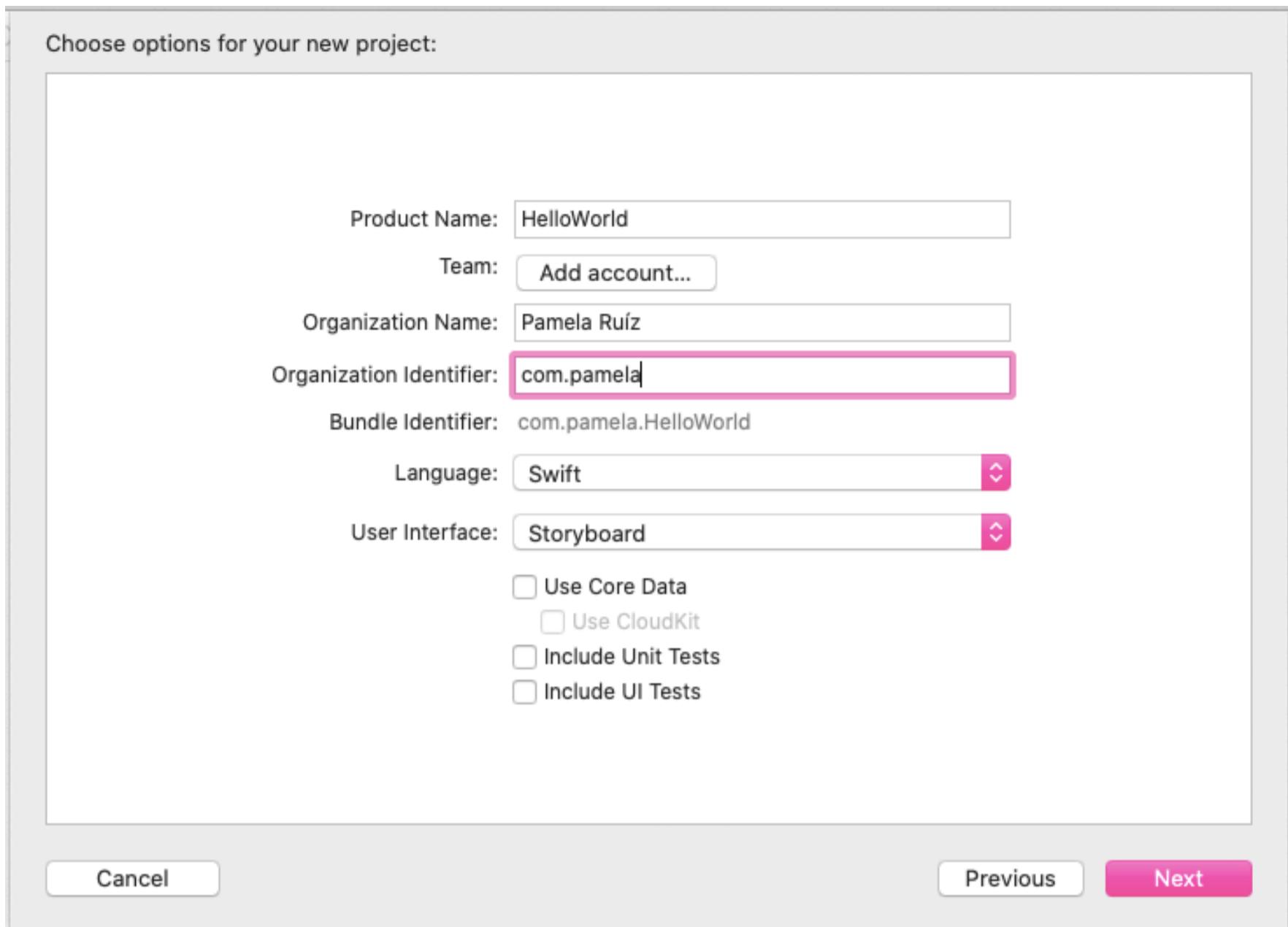
ARREGLOS

Declaramos un arreglo		var arregloNombre: [String]=["Hola", "como", "estás"]
.append	se usa para agregar algo mas al final del arreglo	arregloNombre.append("Rodolfo")
sortedby	arregloNombre.sorted(by: <)	Se usa para ordenar de menor a mayor, nos ordena primero por mayusculas

CONDICIONALES

IF	if variable5 == true { print("verdadero")} else{ print("falso")}
For	for nombre in arregloNombre{ print(nombre)}for i in 0...arregloNombre.count-1{ print(arregloNombre[i])}

CREAR UN PROYECTO



Foreground : Cuando la app carga en la pantalla (verla de frente) Background: Se ejecuta antes de que la aplicación se salga

Alertas

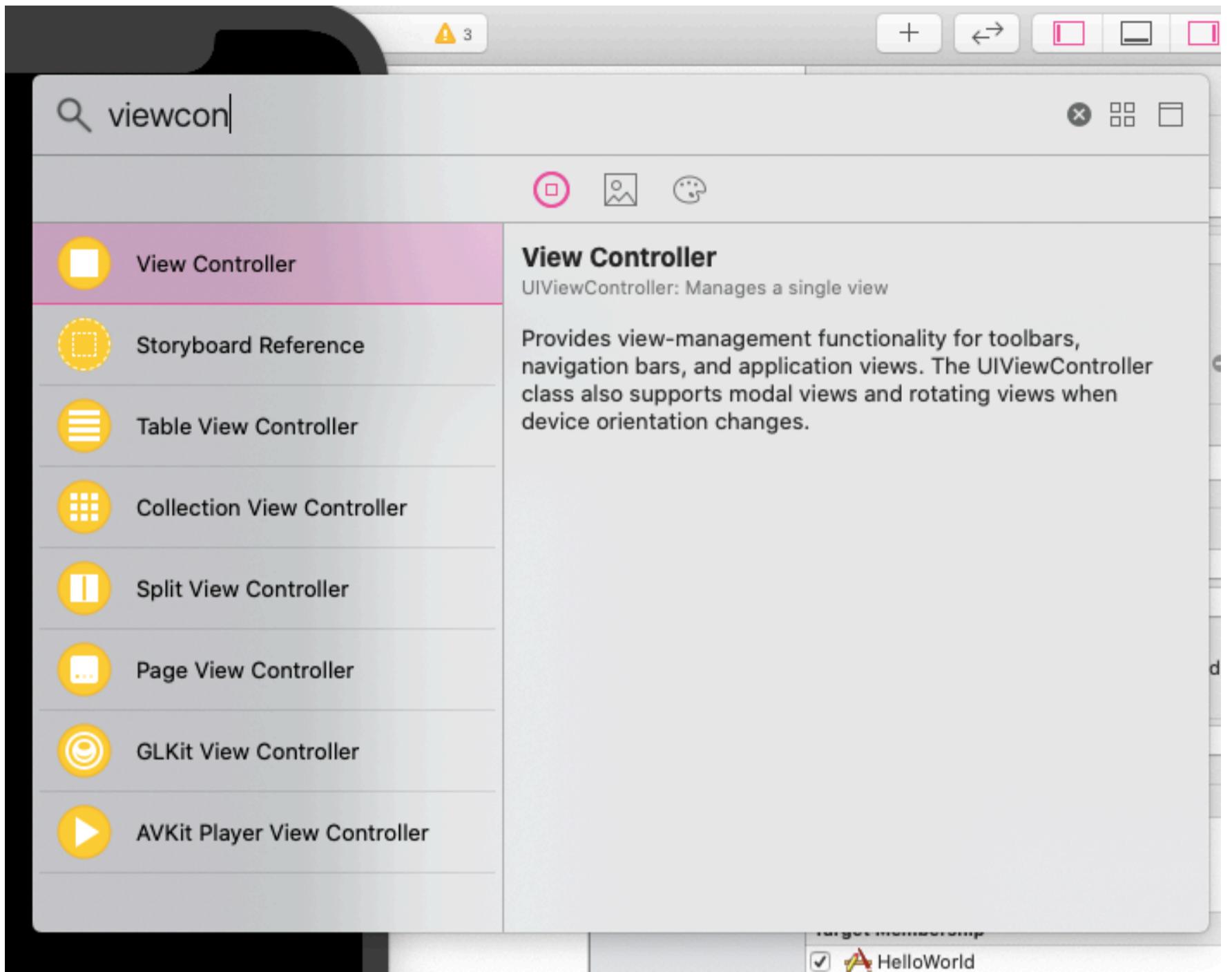
```
let alertController : UIAlertController=UIAlertController(title: "Holis =D, boton pulsado", message: "boton pulsado", preferredStyle: .alert)    present(alertController, animated: true, completion: nil)
```

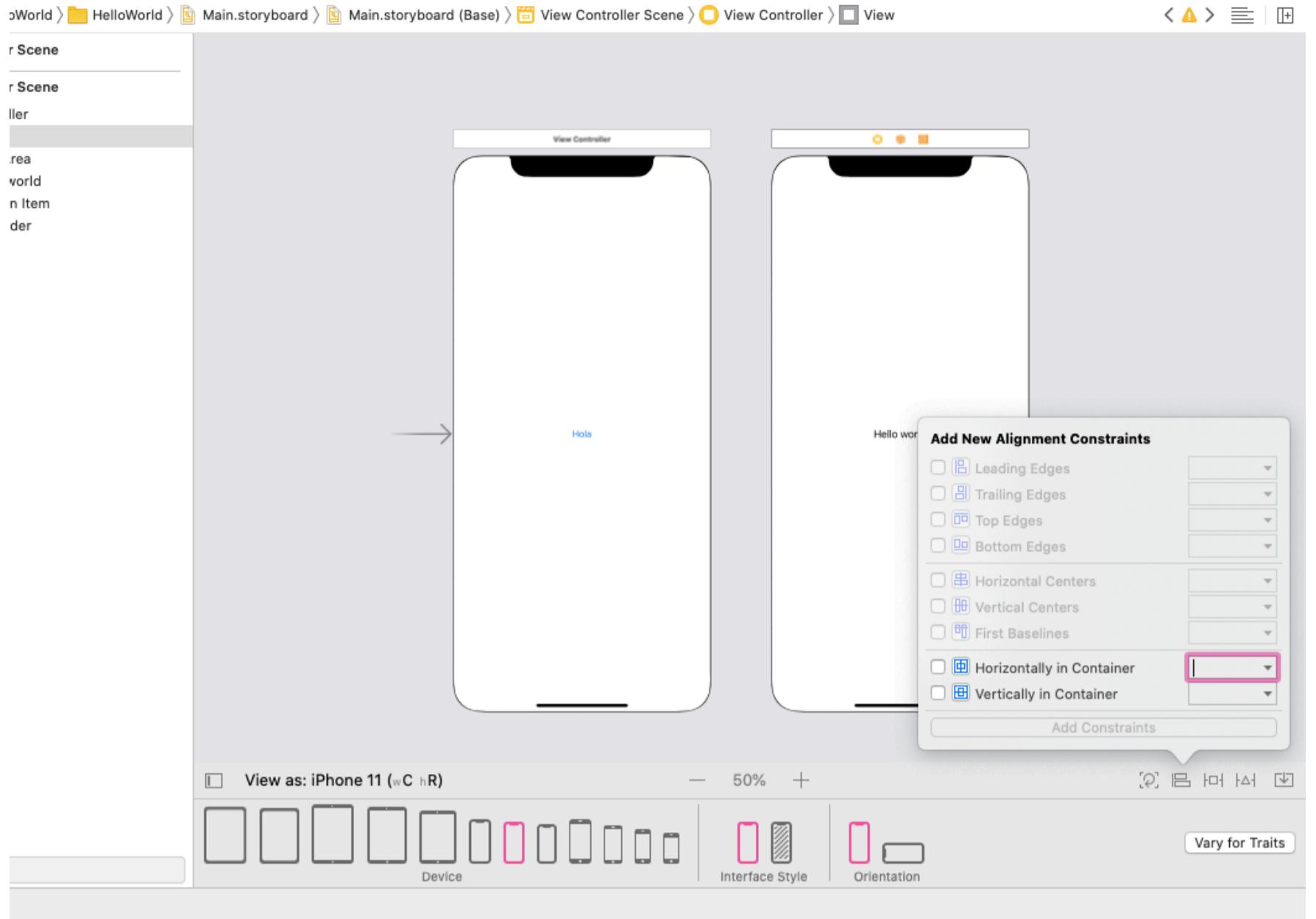
Detalles

```
let alertController : UIAlertController=UIAlertController(title: "Titulo, message: "mensaje", preferredStyle: color (.alert))  
present(alertController, animated: true, completion: nil)
```

View Controller

Es una pantalla donde podemos colocar componentes gráficos, basta con arrastrarlo hacia nuestro storyboard
Para poner otras pantallas, podemos usar los botones, seleccionando, presionando control y arrastrando a la siguiente pantalla para posteriormente seleccionar show y que se liguén las pantallas o views seleccionados

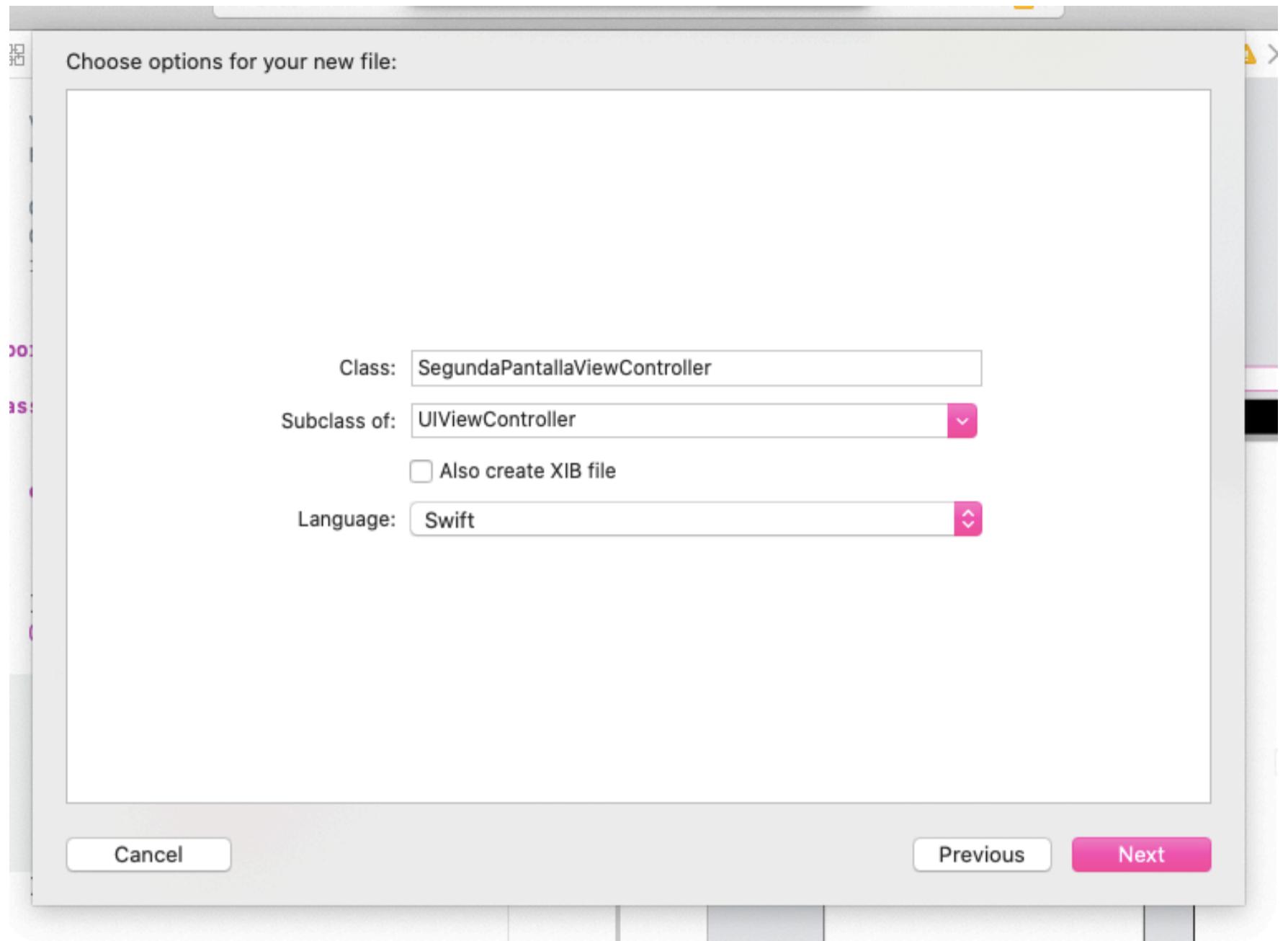




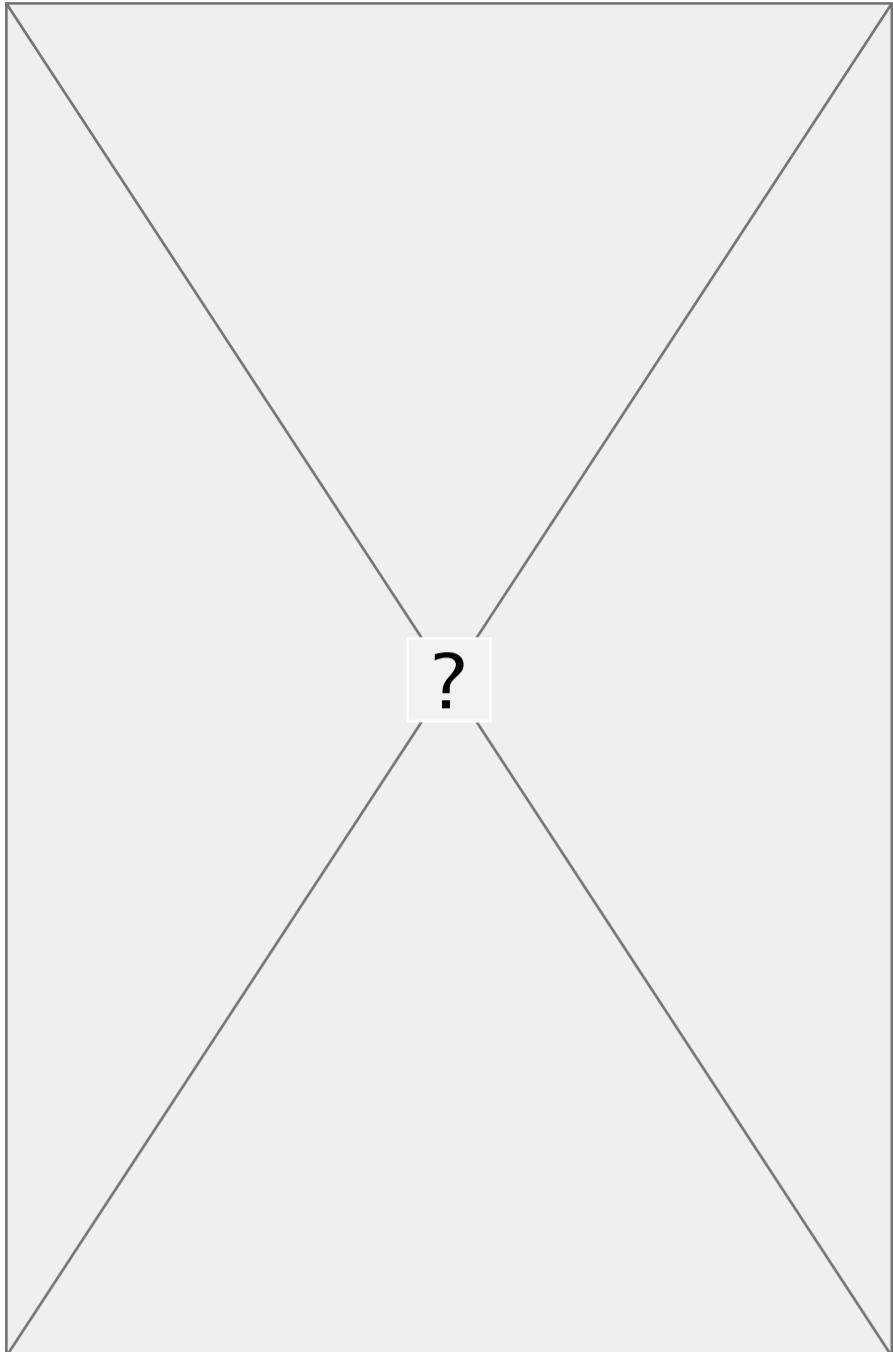
Patrones Mvc, mvc con coordinaciones bipper

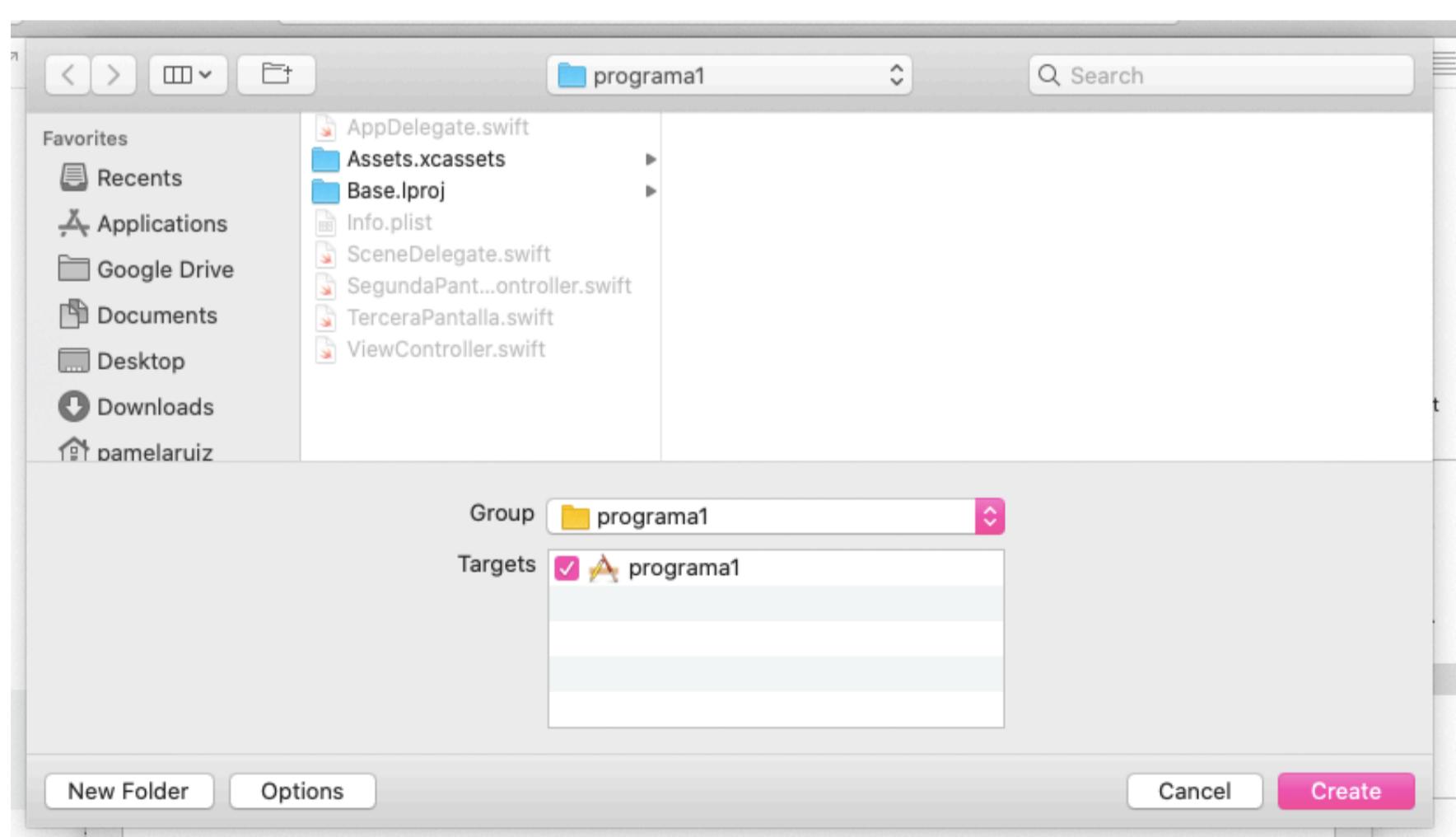
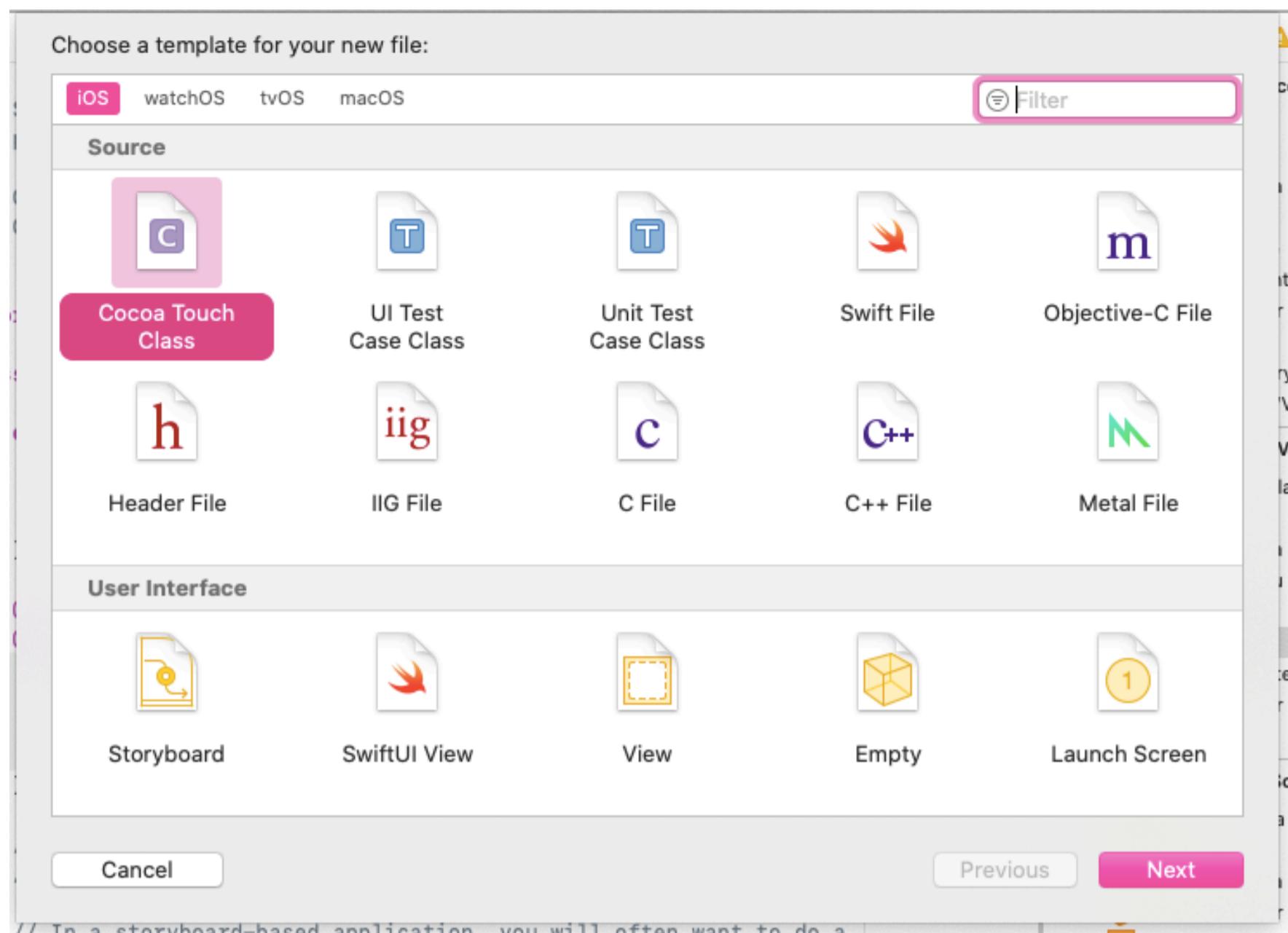
Clases de View Controller

Es importante resaltar que la subclase debe ser **UIViewController**. Cuando creamos una nueva pantalla, también debemos crear una nueva clase para nueva pantalla.



Esto lo logramos dando click derecho en la carpeta de nuestro programa >>New File >>Cocoa Touch Class >> créate





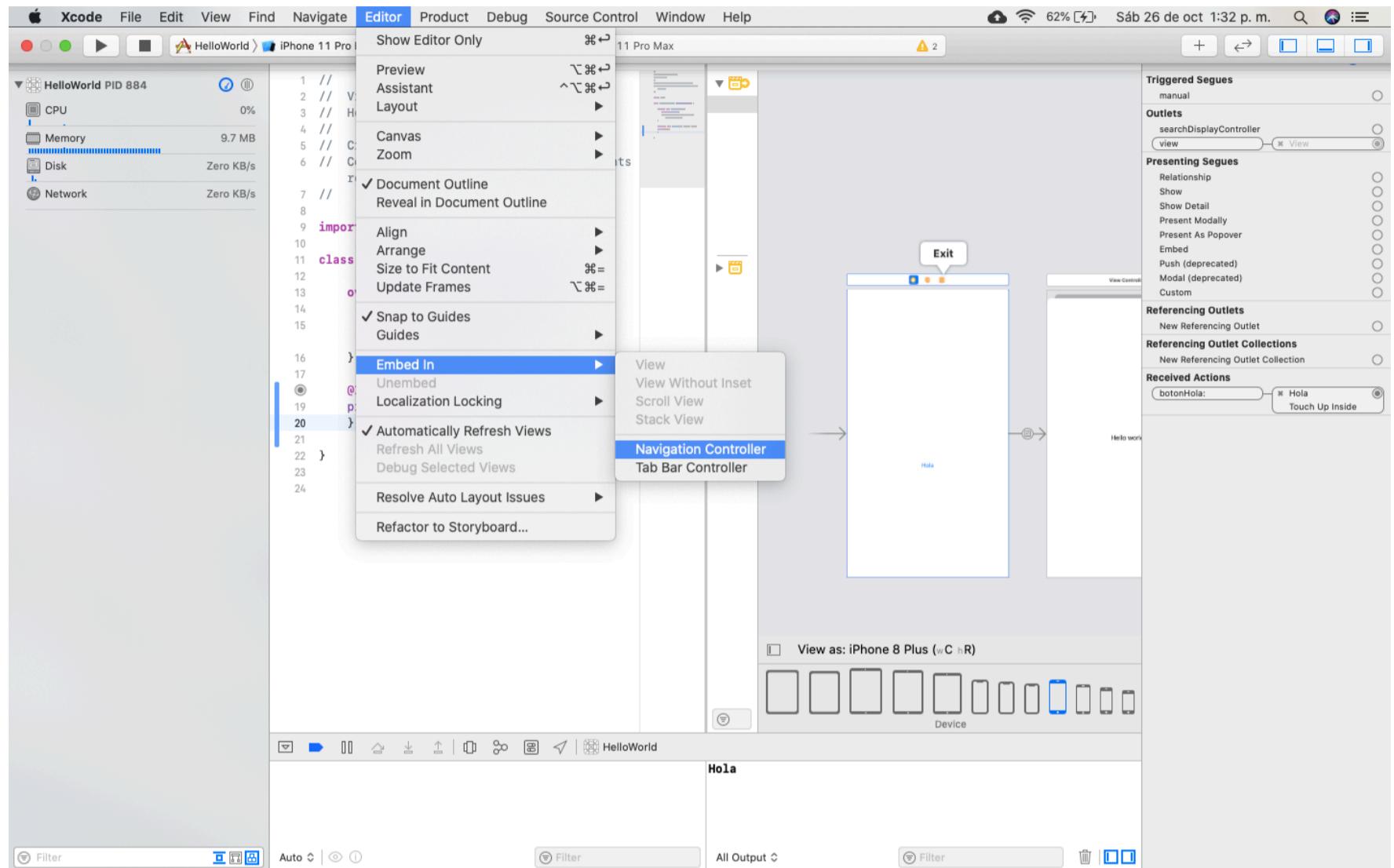
Show y present.

Diferencia: Present si la podemos desviar pero show no

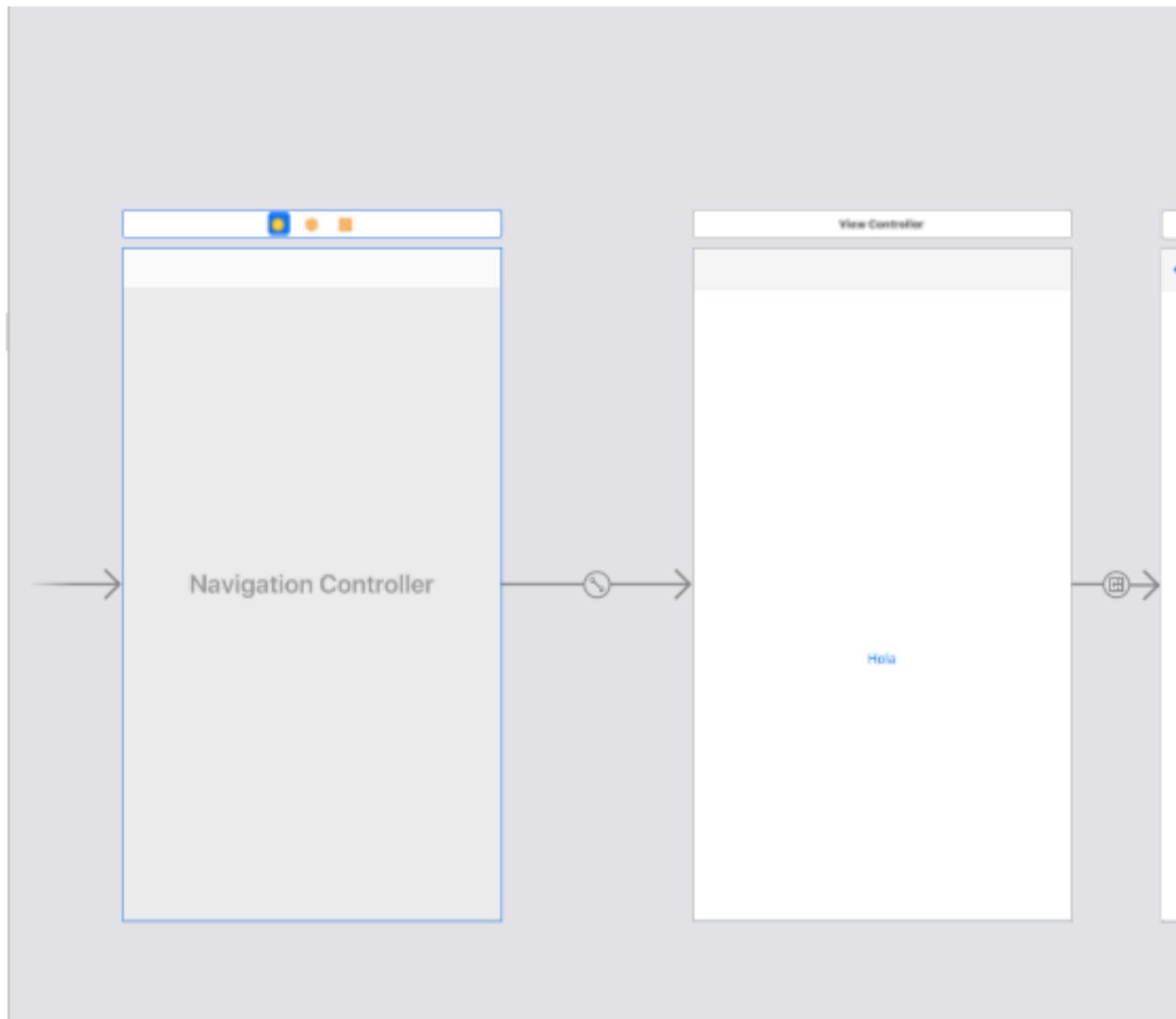
Navigation controller

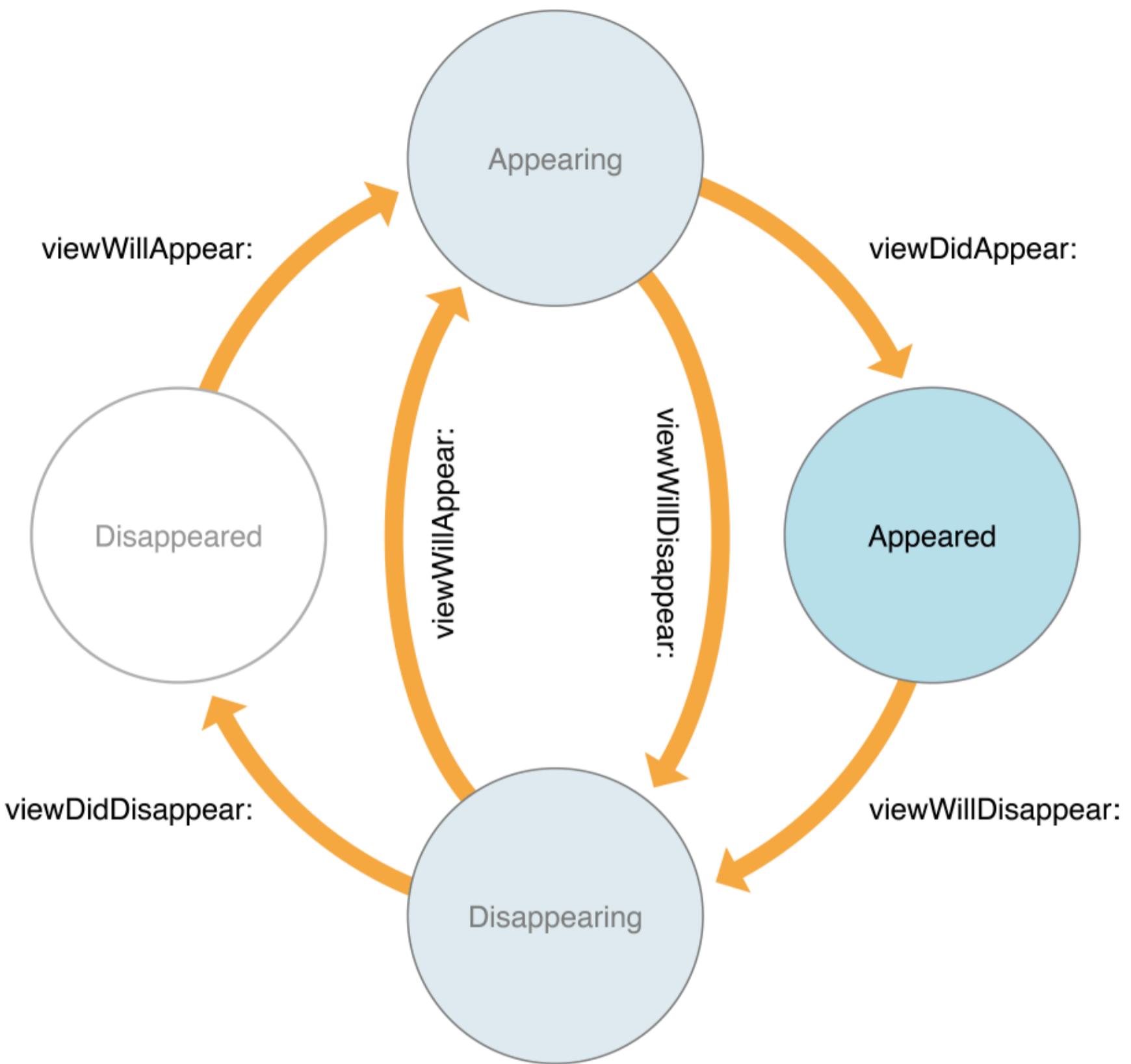
(barra de navegación en la parte de arriba) Nos permite trabajar con flujo entre pantallas

La agregamos dando click en Editor>>Embed In >>Navigation controller



Nos genera el navigation controller





<code>viewDidLoad()</code>	cada componente tiene un ciclo de vida o las fases por las que pasa	<code>override func viewDidLoad(){ super.viewDidLoad() print("load") }</code>
<code>viewLoad()</code>	L(solo se ejecuta una vez) se carga cuando esta completamente cargado todo	
<code>viewWillAppear()</code>	(she ejecuta cada que la pantalla aparece)	<code>override func viewWillAppear(_ animated: Bool){ super.viewWillAppear(animated) print("appear") }</code>

COCOA - core central de iOS Clase central de los(NSObject) Todo lo que es con NS es de Objective c que se vienen acarreando

En el Main.storyboard podemos observar como está estructurado todo en xml

Create XIB file

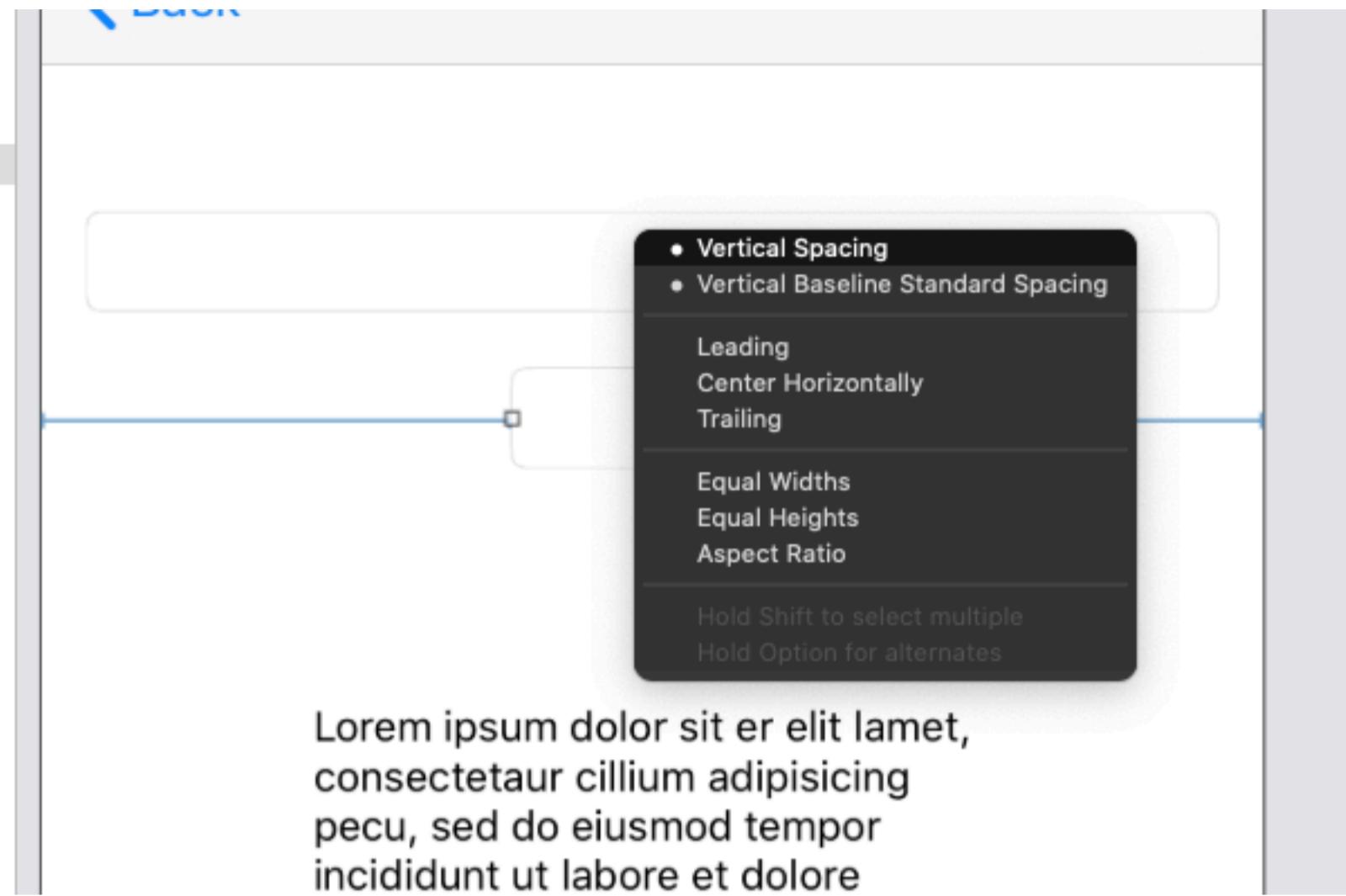
XIB se usaba antes, es un archivo que solo contiene una pantalla, si tenemos un flujo muy grande. Aquí ponemos la versión mínima y lo que usaremos en nuestra aplicación

Constrain

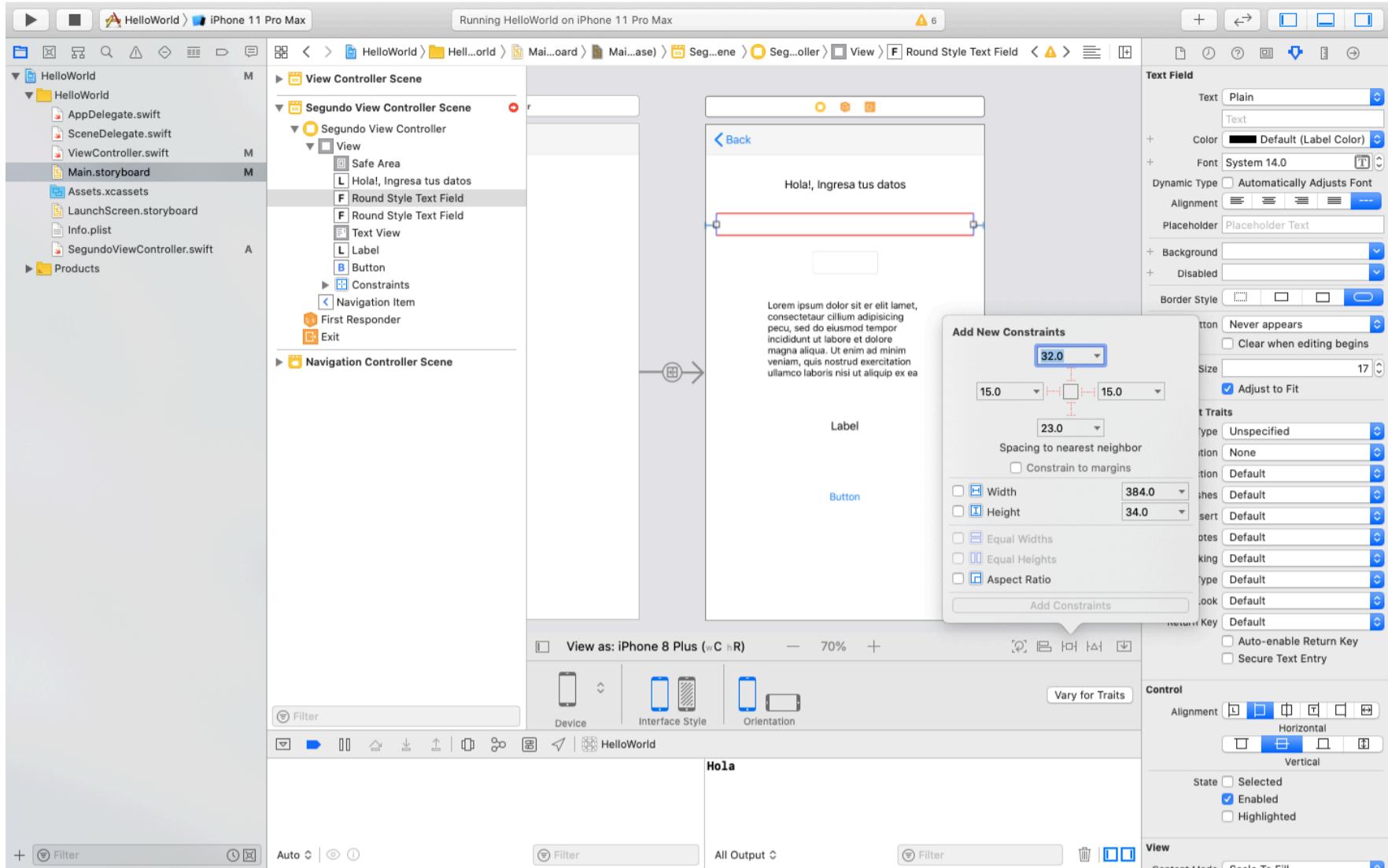
Ya insertados nuestros componentes podemos hacer que se ajusten a la pantalla, ya sea de iPad o cualquier tamaño de iPhone.

¿Cómo?

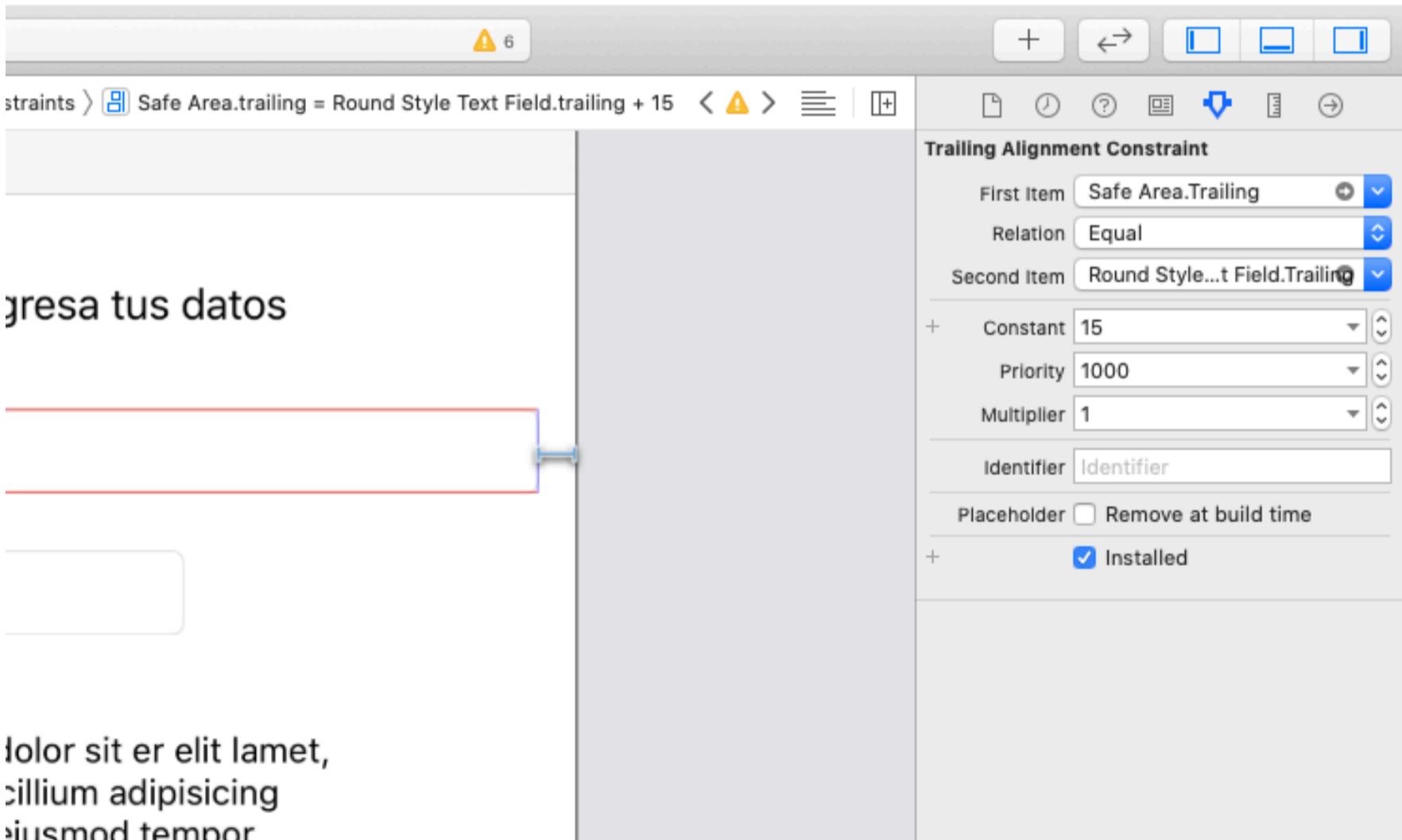
- Seleccionando nuestro componente, presionando ctrl y soltando hacia el lado que deseamos fijar



Usando Add New Constraints

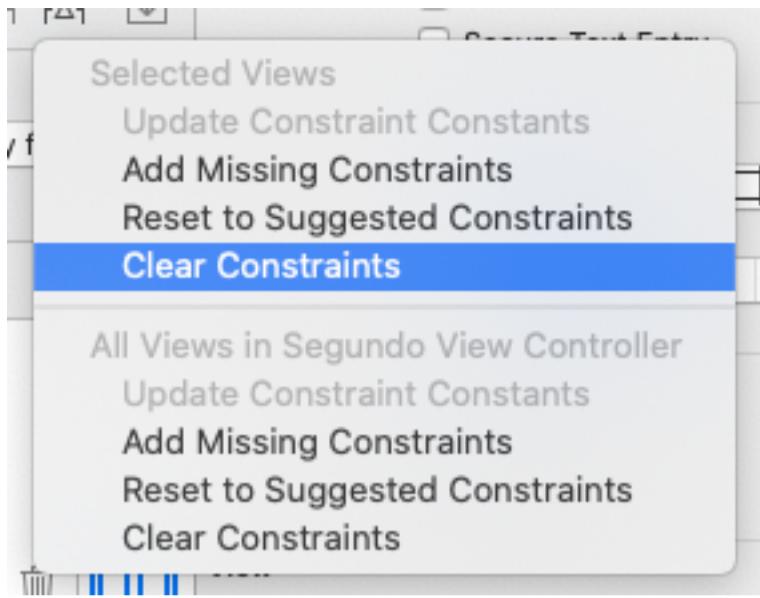


- Seleccionando la linea marcada y después observando Trailing Alignment constraint



color sit er elit lamet,
cillum adipisicing
eiusmod tempor

- También podemos botar los constraints

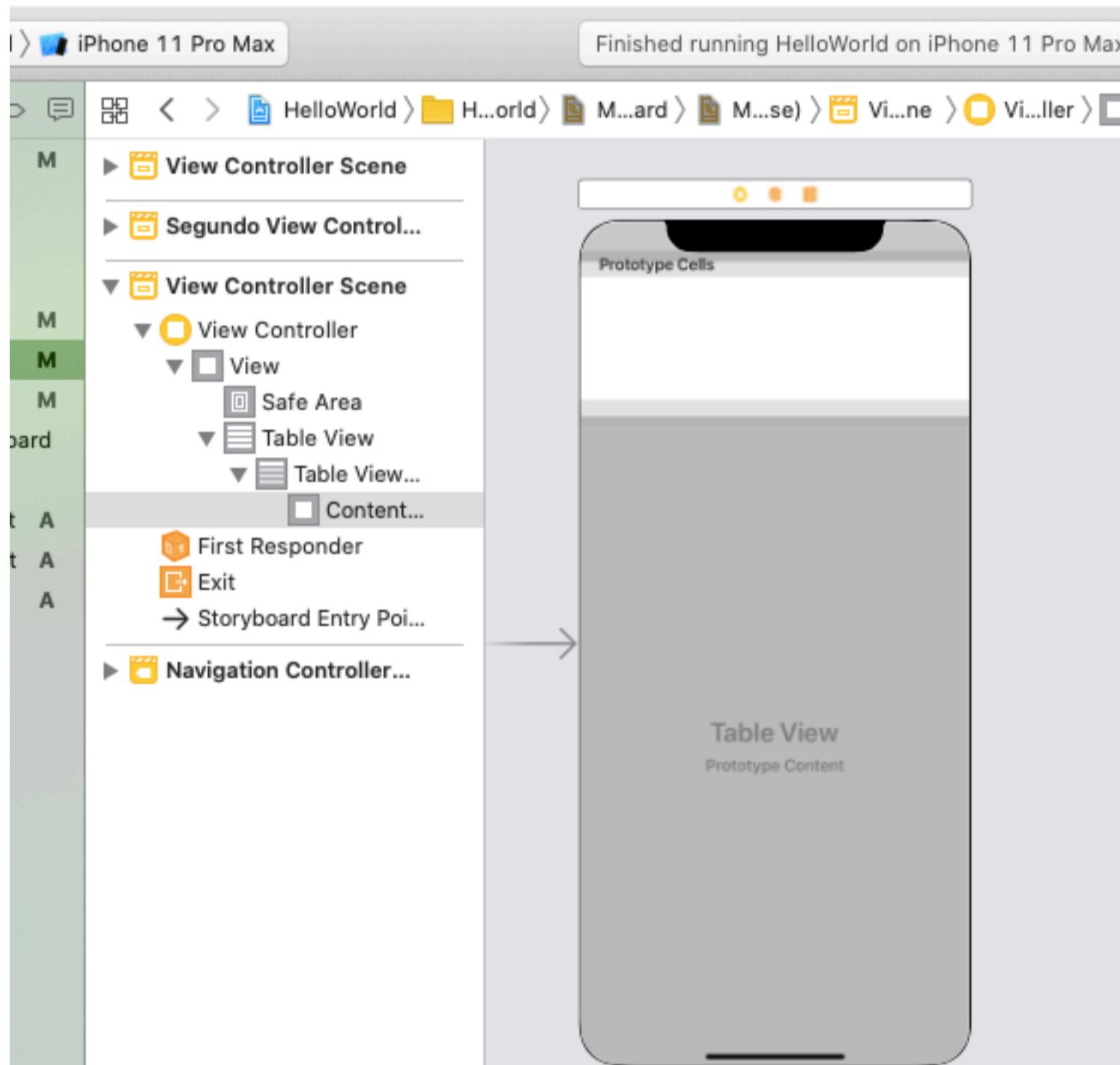


Tables

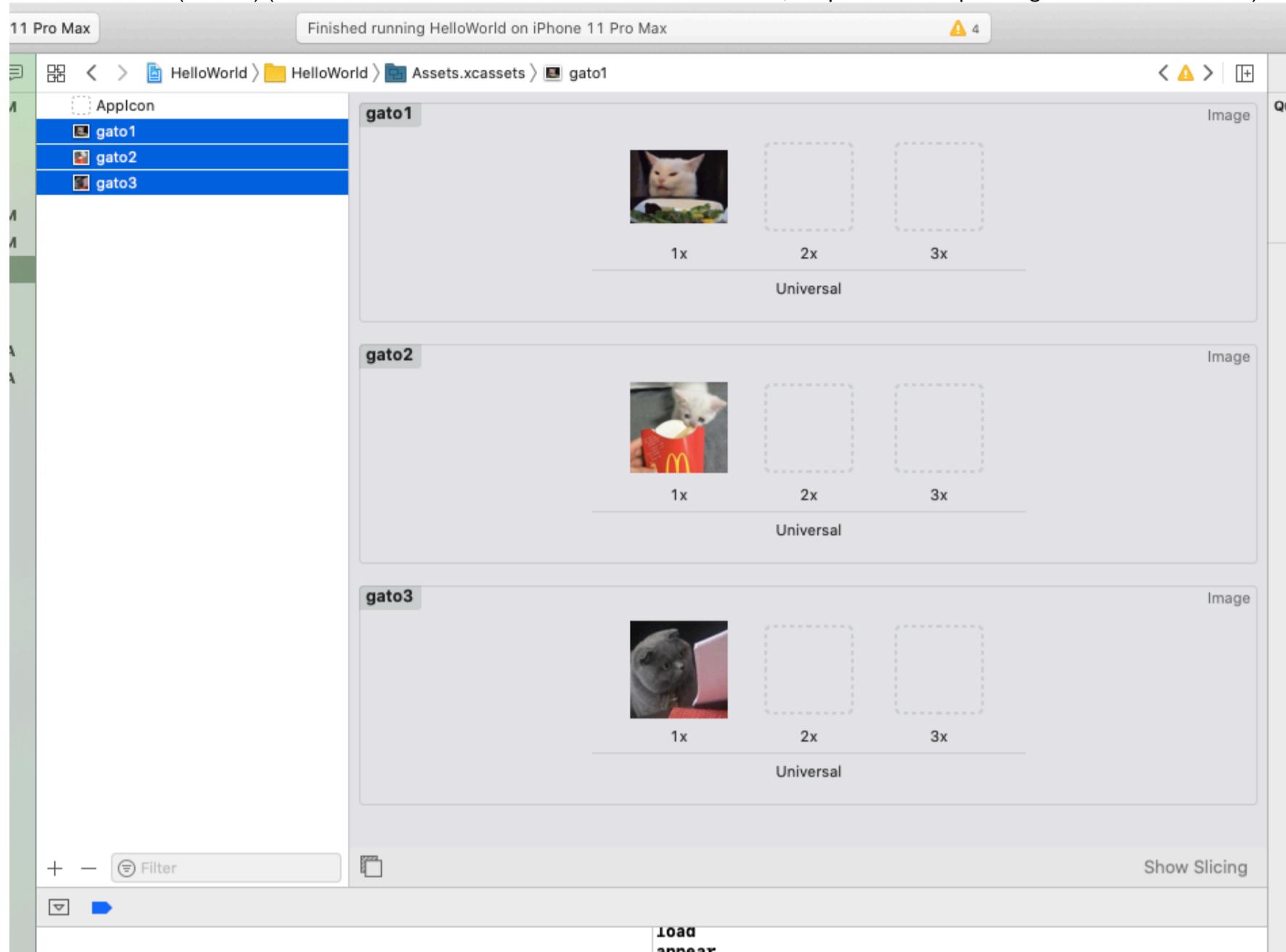
Table/collection

Tableview es un control para meter todo tipo de celdas. Utilizaremos Table view Controller y lo colocaremos en nuestro main.storyboard

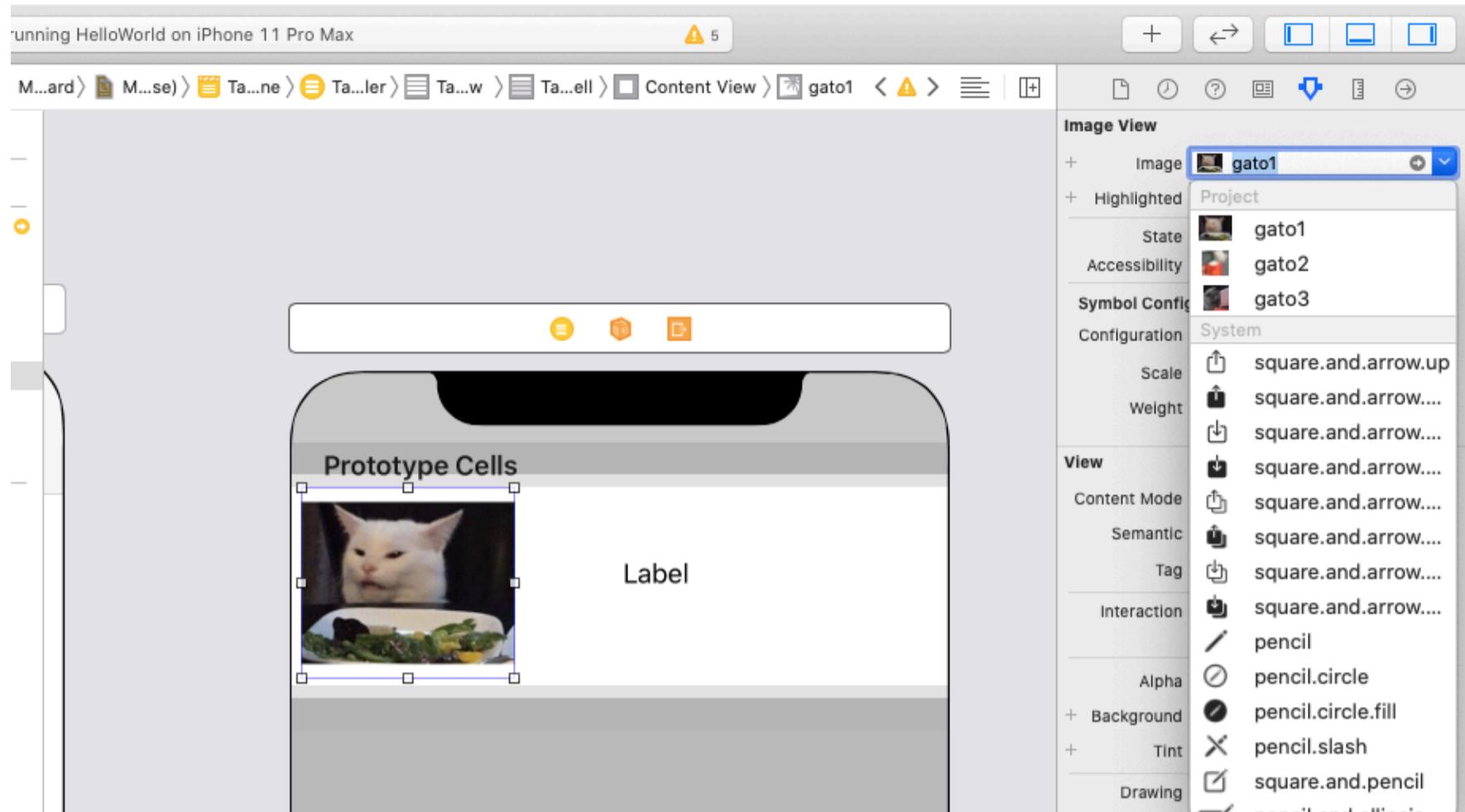
Primero colocamos un view controller luego ponemos un table view y al final un table view cell



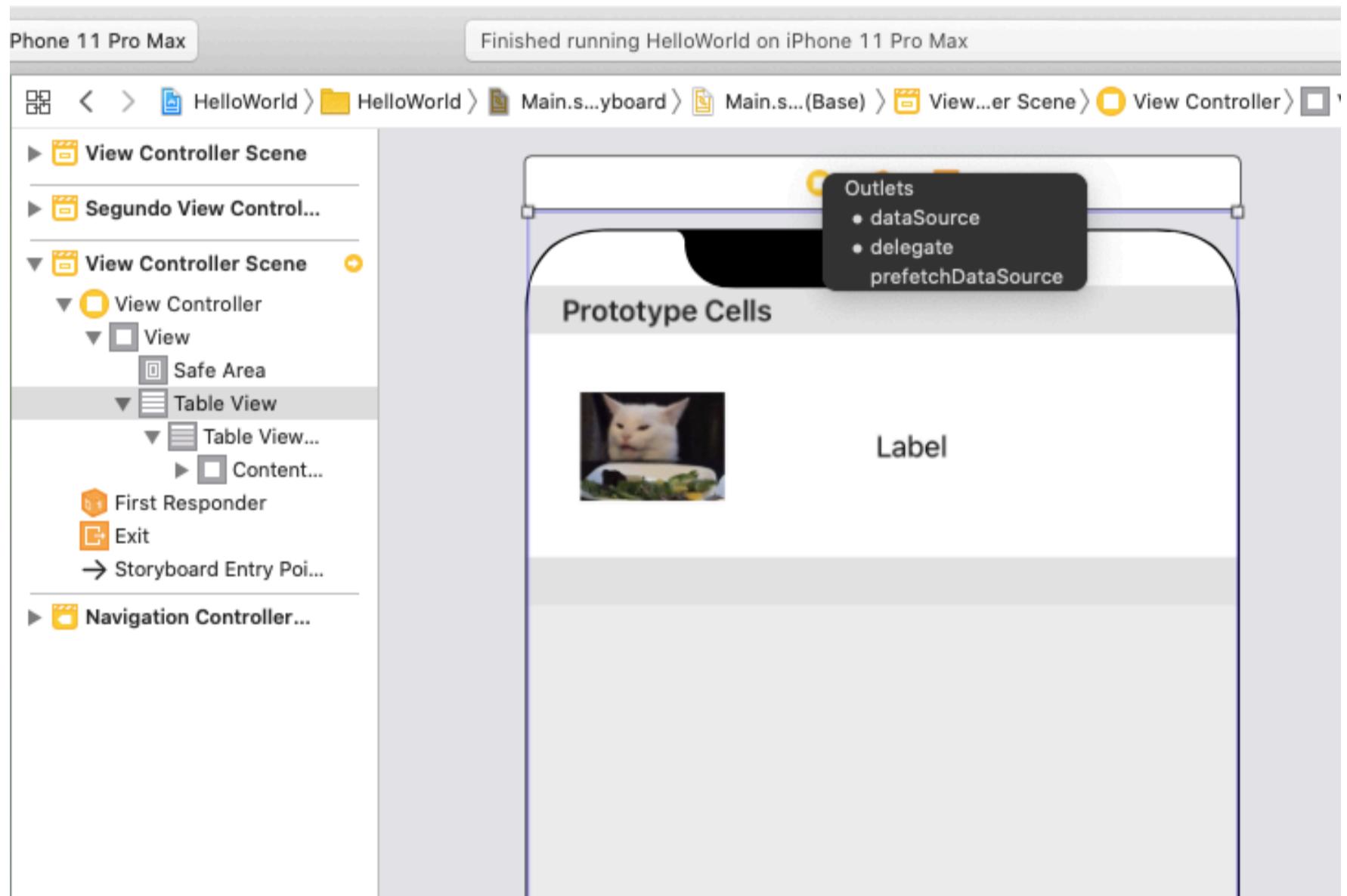
Nuestros assets o imágenes las arrastraremos desde alguna carpeta de nuestra computadora a la carpeta Assets.xcassets(abierta) (Recuerda cuidar/saber el nombre de los archivos; de preferencia que tengan un nombre corto)



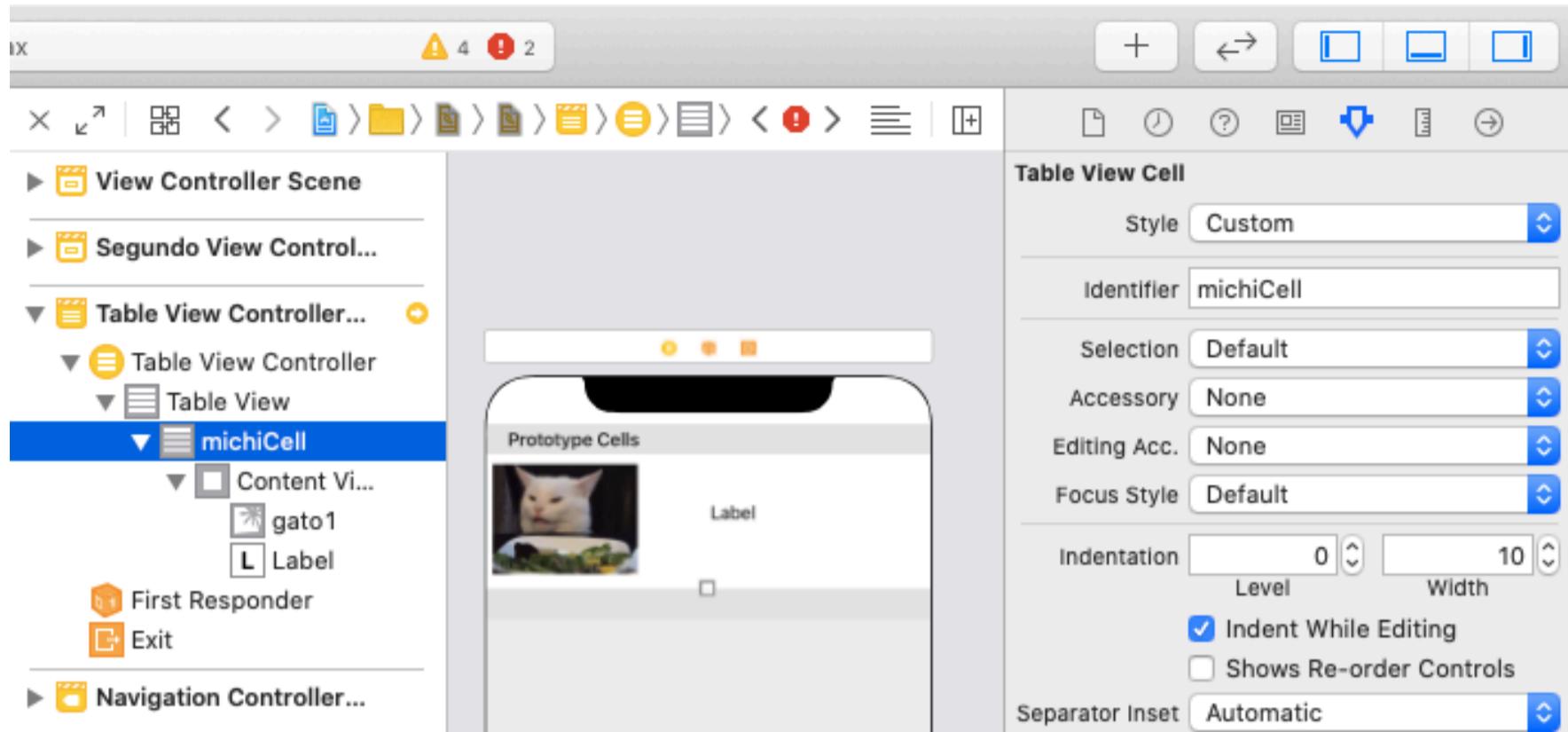
Ahora dentro de la celda o el table view cell colocarmos una imagen con ImageView y un label, para poder seleccionar la imagen, podemos ver en la sección de Image View>> Image



Seleccionamos el TableView+ctrl y arrastramos a el signo Dejando el menú que se despliega de la siguiente forma



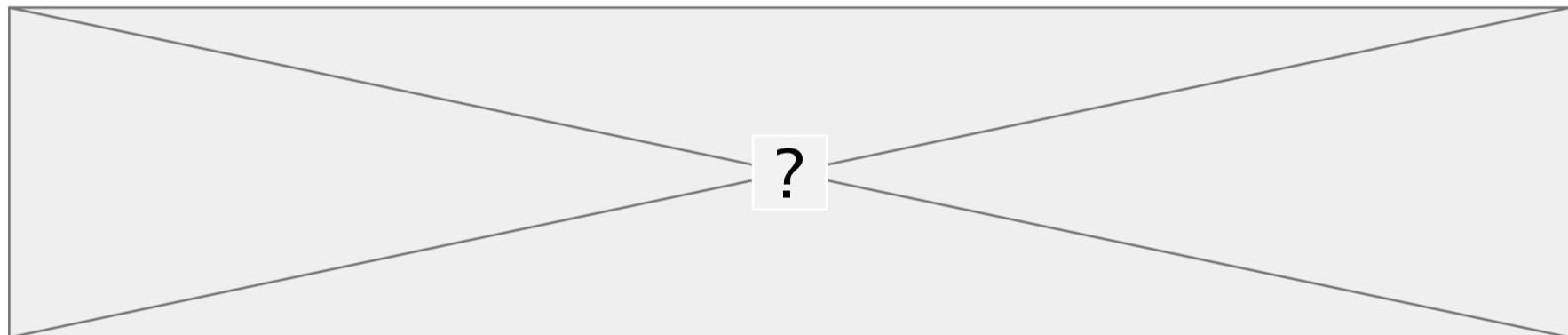
No olvides colocar un identifica a la celda, ya que después lo ocuparemos



Dentro de la clase, después de UIViewController agregamos:

`UITableViewDelegate, UITableViewDataSource`

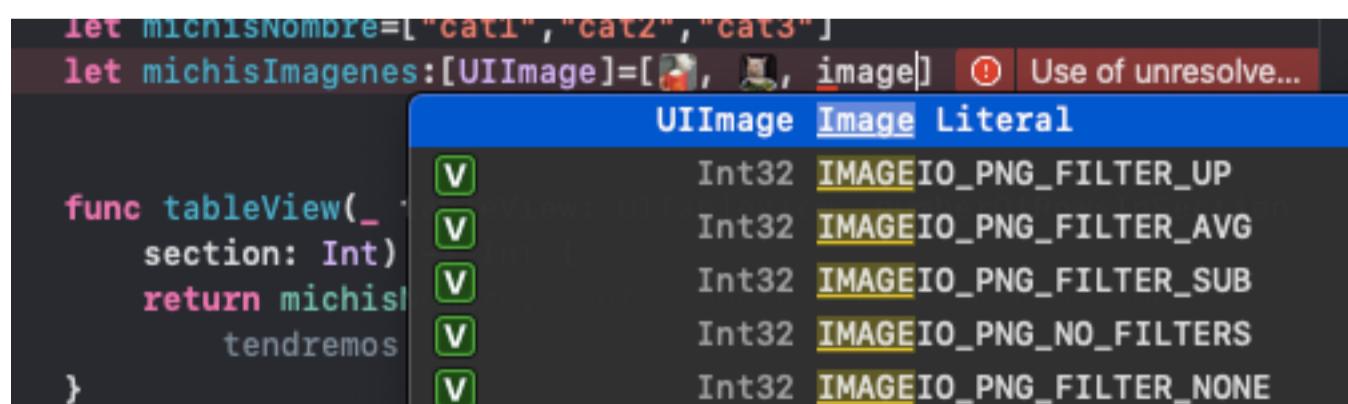
nos marcará un error, seleccionaremos fix y agregará una sección de código:



Ahora, antes de ese código, apenas iniciando la clase colocaremos dos arreglos, uno con nombres y el otro con las imágenes:

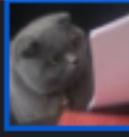
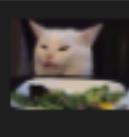
```
let michisNombre=["cat1","cat2","cat3"]
let michisImagenes:[UIImage]=[#imageLiteral(resourceName: "gato2"),
#imageLiteral(resourceName: "gato1"), #imageLiteral(resourceName: "gato3")]
```

Al seleccionar `ImageLiteral` en las opciones nos aparecerá una imagen vacía pero al seleccionarla nos proporcionará las imágenes que agregamos al proyecto como opciones



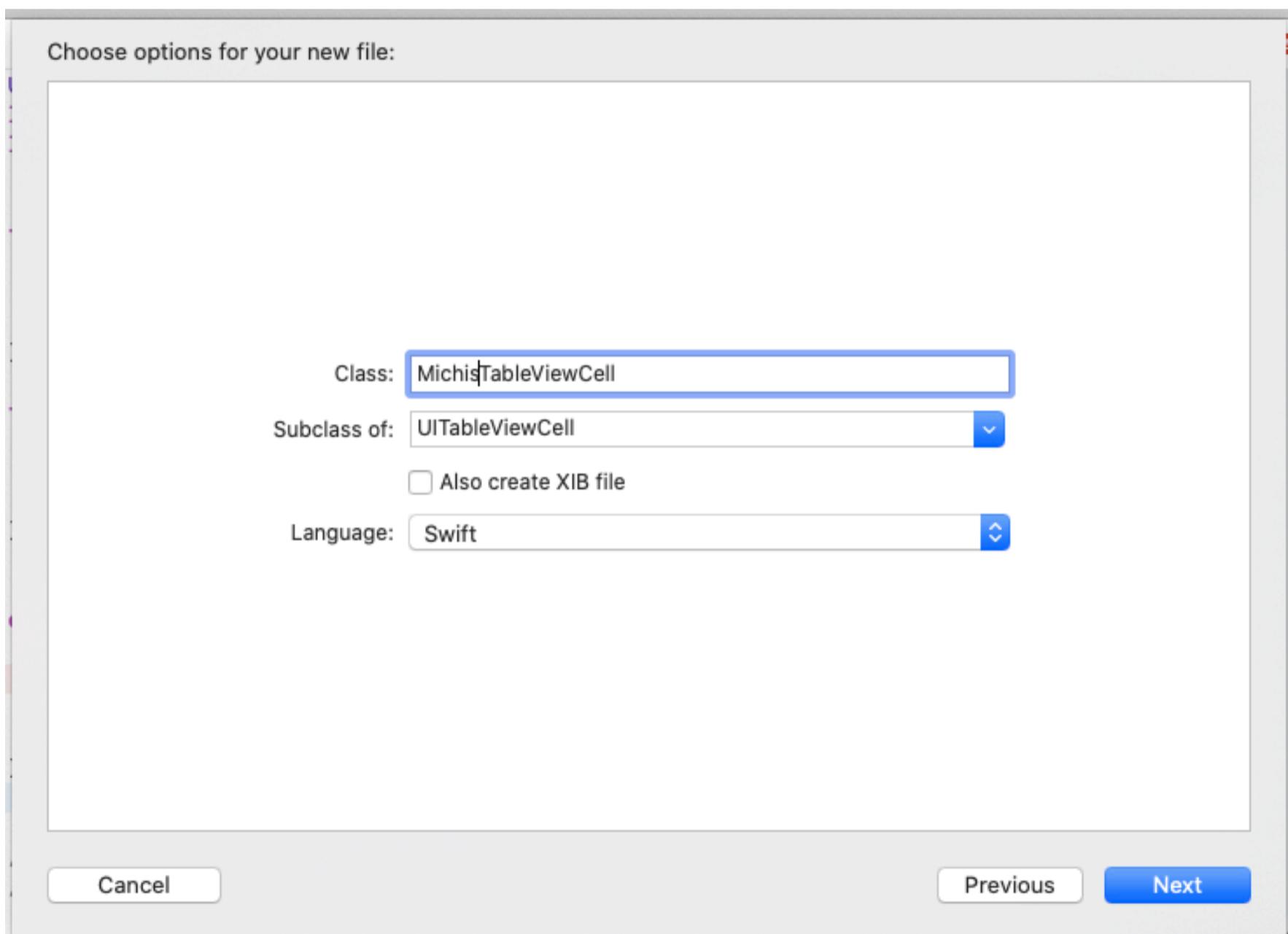
```
Copyright © 2019 Pamela Ruiz. All rights reserved.

import UIKit

class MichisViewController: UIViewController {
    @IBOutlet var michisTV: UITableView!
    let michisNombre=[ "cat1", "cat2", "cat3"]
    let michisImagenes:[UIImage]=[, , ]

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

Ahora crearemos otra clase de tipo cocoa touch , Esta vez, la subclase será UITableViewCell



Dentro del archivo **MichisTableViewCell** declaramos nuestras variables `lblMichi` e `imageMichi` arrastrando desde la celda, la imagen y el `lbl` y presionando `ctrl`.

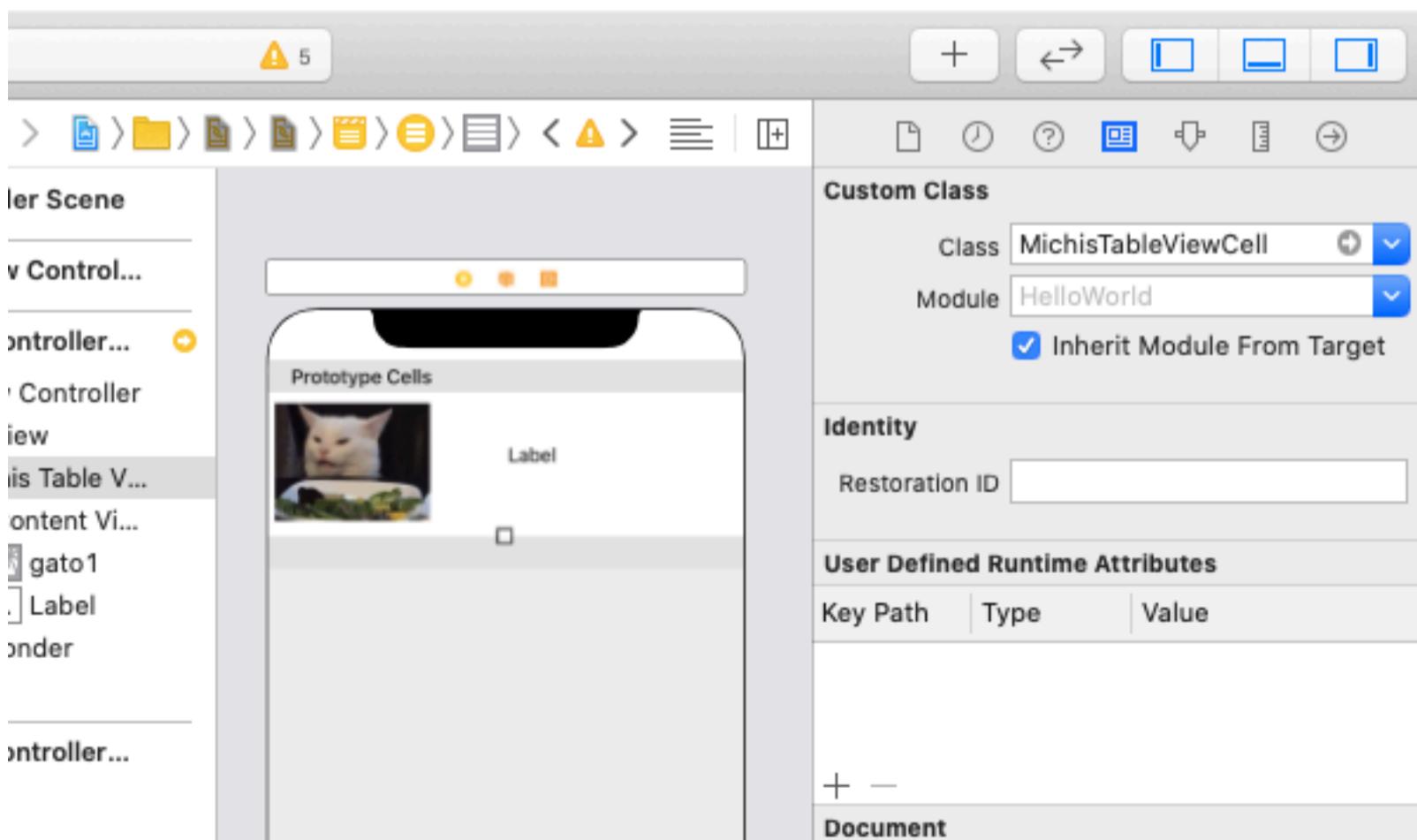
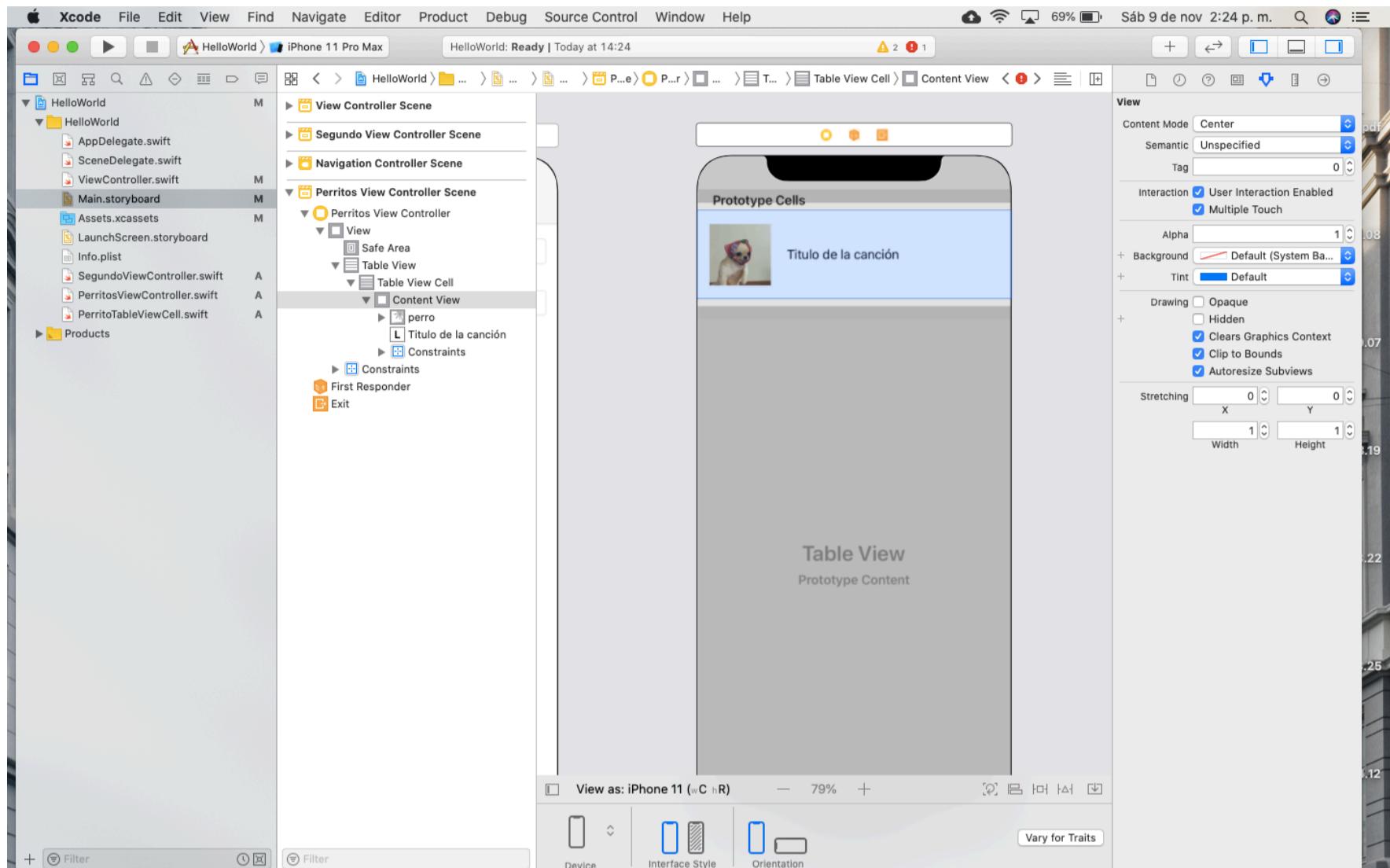
The screenshot shows the Xcode interface with the following details:

- Top Bar:** Find, Navigate, Editor, Product, Debug, Source Control, Window, Help.
- Project Navigator:** Shows the project structure: iPhone 11 Pro Max, HelloWorld, HelloWorld, PerritosViewController.swift.
- Status Bar:** Shows the device as iPhone 11 Pro Max, the project as HelloWorld: Ready, the date as Today at 14:23, and build statistics (2 warnings, 1 error).
- Code Editor:** Displays the Swift code for PerritosViewController.swift. The code initializes a UITableView with three sections, each containing an image of a dog. It includes imports for UIKit and the PerritosViewController class, and overrides for viewDidLoad(). A red callout "Expected pattern" points to the line "let".
- Right Sidebar:** Shows the Navigation pane.

```
1 //  
2 // PerritosViewController.swift  
3 // HelloWorld  
4 //  
5 // Created by Rodolfo Benjamin Alcantara Solorio on 09/11/19.  
6 // Copyright © 2019 Rodolfo Benjamin Alcantara Solorio. All rights reserved.  
7 //  
8  
9 import UIKit  
10  
11 class PerritosViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {  
12     let perritosNombre=["1","2","3"]  
13     let perritosImagenes:[UIImage]=[UIImage(named:"1"), UIImage(named:"2"), UIImage(named:"3")]  
14  
15     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
16         code  
17     }  
18  
19     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->  
20         UITableViewCell {  
21         code  
22     }  
23  
24     override func viewDidLoad() {  
25         super.viewDidLoad()  
26         let  
27             // Do any additional setup after loading the view.  
28     }  
29  
30 }
```

- [] 1 Declarar tableview en la clase???

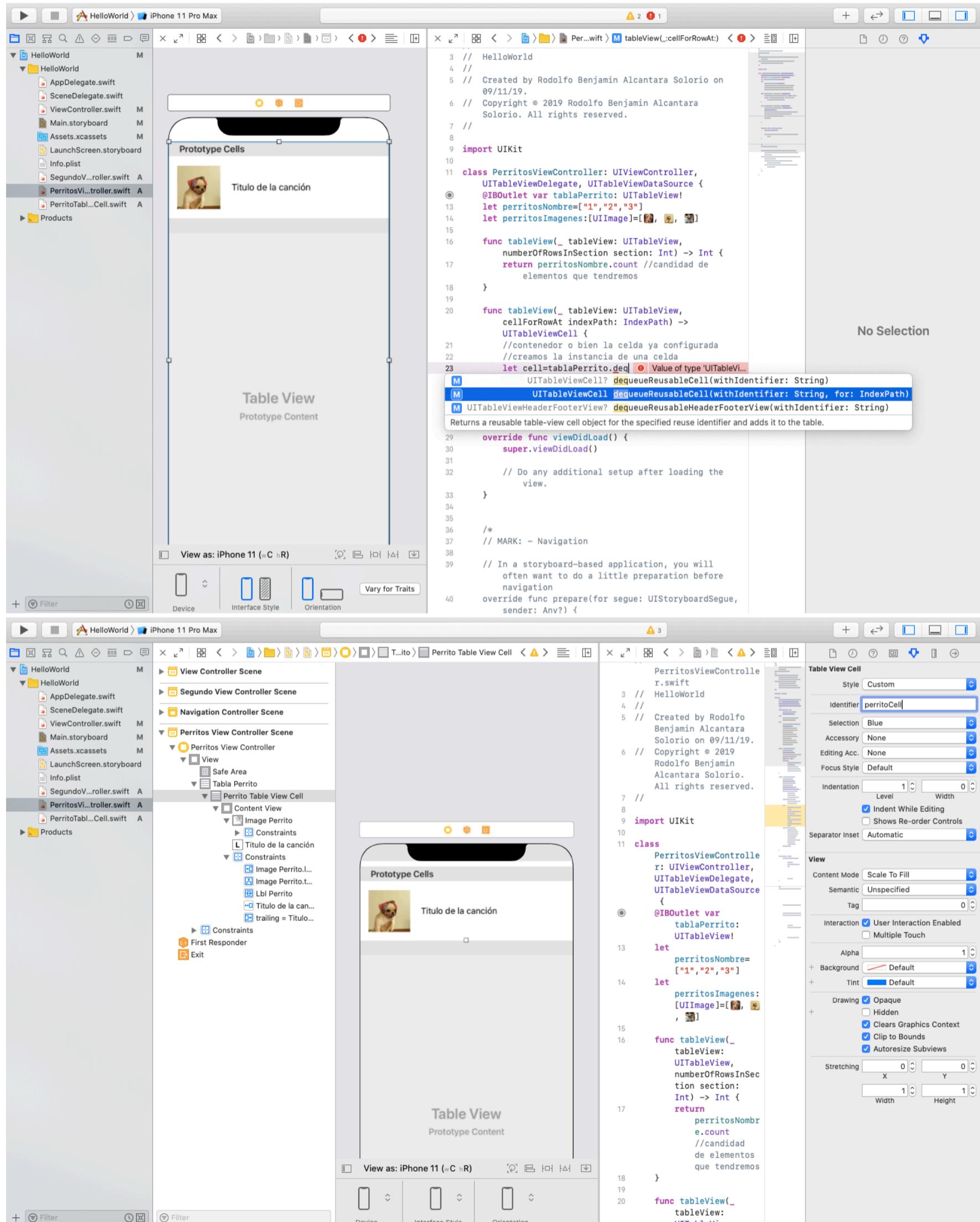
Seleccionamos nuestra celda y posteriormente colocamos el nombre de la clase dentro de “Custom Class”, presionamos enter

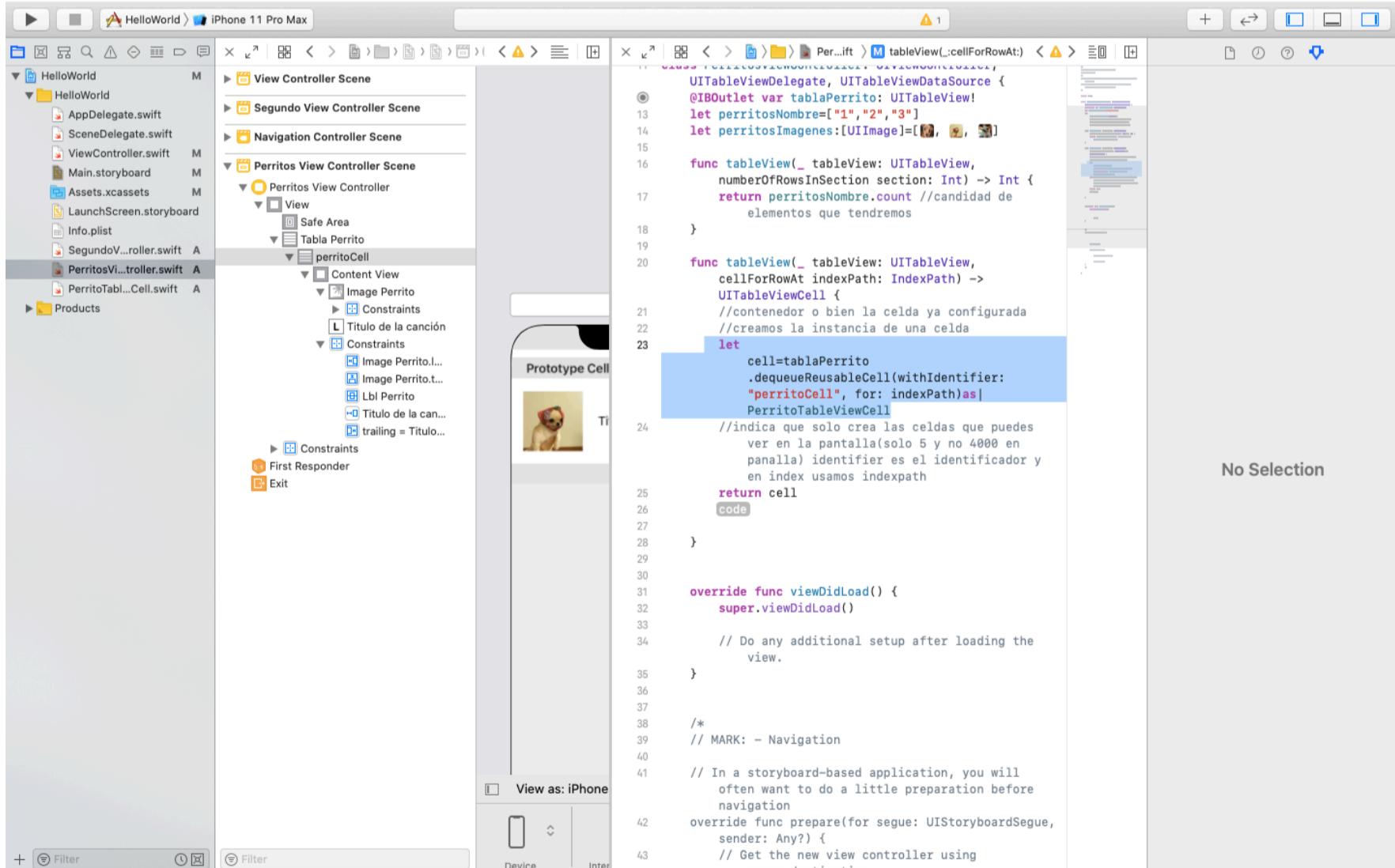
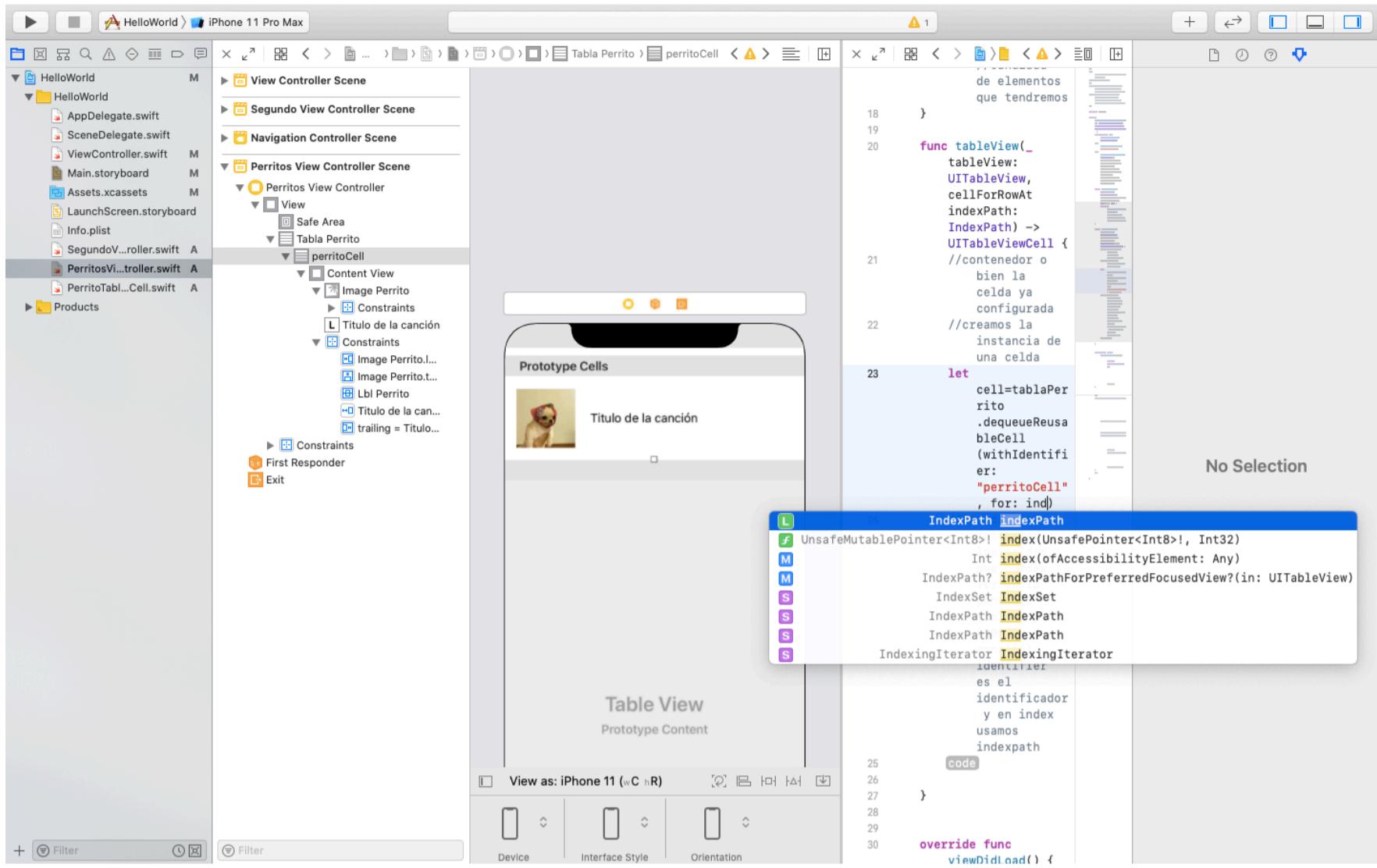


Dentro de la primer función colocaremos

```
return michisNombre.count //cantidad de elementos que tendremos
```

- [] - 2 necesito tabla perrito (table view declarado) para tomar los ss del programa (Sin dudas, solo faltan fotos)
Ahora bien, en la segunda función declararemos la variable cell (de celda) donde colocaremos el siguiente código





//contenedor o bien la celda ya configurada
//Creamos la instancia de una celda

```

let cell = tablaPerrito.dequeueReusableCell(withIdentifier: "perritoCell", for: indexPath) as! PerritoTableViewCell
cell.lblPerrito.text = perritosNombre[indexPath.row]
cell.imagePerrito.image = perritosImagenes[indexPath.row]
return cell
//indica que solo crea las celdas que puedes ver en la pantalla(solo 5 y no 4000 en panalla) identifier es el identificador y en index usamos indexpath
Dequeue lo que hace es que solo muestra en pantalla con la memoria las celdas que quieras ver, así ahorra la memoria

```

Después de la segunda función colocamos la función:

```

func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
    return 100
}
//retornamos el tamaño de cada celda

```

```

let cell = tablaPerrito.dequeueReusableCell(withIdentifier: "perritoCell", for: indexPath) as!
PerritoTableViewCell

cell.lblPerrito.text = perritosNombre[indexPath.row]
cell.imagePerrito.image = perritosImagenes[indexPath.row]
return cell

//indica que solo crea las celdas que puedes ver en la pantalla(solo 5 y no 4000 en panalla)
identifier es el identificador y en index usamos indexpath
}

func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
    return 100
}

```

```

override func viewDidLoad() {
    super.viewDidLoad()

    // Do any additional setup after loading the view.
}

```

Alertas

Ahora colocaremos la 3ra función, la cual nos aparece como opción al ingresar “didselectrowa”

```

27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

```

    return cell

    //indica que solo crea las celdas que puedes ver en la pantalla(solo 5 y no 4000 en panalla)
    identifier es el identificador y en index usamos indexpath

Tells the delegate that the specified row is now selected.

tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
tableView(_ tableView: UITableView, didDeselectRowAt indexPath: IndexPath)
tableViewDidEndMultipleSelectionInteraction(_ tableView: UITableView)
didChange<Value>(_ changeKind: NSKeyValueChange,...h: __owned KeyPath<PerritosViewController, Value>)
didselectrowa

```

3 ⓘ Expected '(' in argument list of function declaration

```

override func viewDidLoad() {

func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
    return 100
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    let alert = UIAlertController(title: perritosNombre[indexPath.row], message: "este es el
        mensaje", preferredStyle: .alert)
    alert.addAction(UIAlertAction(title: "ok", style: .default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}

override func viewDidLoad() {
    super.viewDidLoad()

    // Do any additional setup after loading the view.
}

```

En la tercera(primer en el recuadro) soloreturnamos 100, mientras que en la cuarta(segunda en el cuadro) colocaremos las alertas
tableView colocamos el siguiente código seleccionando la opción subrallada en azul

```
0
1     let alerta= UIAlertController(          2 ▲○ '= must have consistent whitespace on both sides
2
3     let ale
4     es
5     alerta.
6     self.
7 }  
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    let alert = UIAlertController(title: michisNombre[indexPath.row], message: "este es el mensaje", preferredStyle: .alert)
    alert.addAction(UIAlertAction(title:"ok", style: .default, handler:nil))
    self.present(alert,animated: true,completion: nil)
}
```

//Donde michisNombre es el nombre del arreglo que ya habíamos declarado

SWIFT: Lenguaje de protocolos

Protocolos: nosotros como humanos tenemos el protocolo de respirar

Overide .sobrecarga

Collection view

Pediremos datos a una pagina de internet no a un servicio

Detalle view controller colocamos el nombre en nuestra nueva pantalla, luego declaramos un label en nuestra clase
allocade(cuanta memoria necesito)

Init(lleno esa memoria de algo)

Usarlo

Init (vaciar lo que tengo y regresar en allocade la información)

Viewdidload()

En DetalleViewController

```
import UIKit
class DetalleViewController: UIViewController {
    @IBOutlet var detalles: UILabel!
    var nombre:String="Hola a todos"
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
```

En perritos view controller

Quitamos la sección de código

```
let alert = UIAlertController(title: perritosNombre[indexPath.row], message: "este es el mensaje", preferredStyle: .alert)
alert.addAction(UIAlertAction(title:"ok", style: .default, handler:nil))
```

```
self.present(alert,animated: true,completion: nil)
```

```
let viewController = UIStoryboard(name: "Main", bundle: nil)
```

name: nombre

Bundle:referencia a la construcción del proyecto: Hace referencia a un proyecto

```

33     return 100
34 }
35
36 func tableView(_ tableView: UITableView,
37 didSelectRowAt indexPath: IndexPath) {
37     let viewController = UIStoryboard(name: "Main",
38                                     bundle: nil)
39     //UIViewController? instantiateInitialViewController()
40     //ViewController? instantiateInitialViewController()
41     //ViewController? instantiateInitialViewController(creator: ((NSCoder) -> ViewController?)?)
42     //ViewController instantiateViewController(identifier: String)
43     //ViewController instantiateViewController(identifier: String, creator: ((NSCoder) -> ViewController?)?) {
44     //UIViewController instantiateViewController(withIdentifier: String) // Get the new view controller using segue.destination.
45     //UIStoryboard self.instantiateViewController(withIdentifier: String) // Pass the selected object to the new view controller.
46     //Self `self`()
47
48     /*
49     // MARK: - Navigation
50 */

```

Creates the specified view controller from the storyboard and initializes it using your custom initialization code.

```

17
18     // Do any additional setup after
19     // loading the view.
20
21     /*
22     // MARK: - Navigation
23
24     */
25
26     /*
27     // In a storyboard-based application, you
28     // might want to do some preparation before
29     // the controller is revealed.
30     */
31
32 */

```

Swift code in DetaillePerrito.swift:

```

24     cell.lblPerrito.text =
25         perritosNombre[indexPath.row]
26     cell.imagePerrito.image =
27         perritosImagenes[indexPath.row]
28     return cell
29
30     //indica que solo crea las celdas que
31     //puedes ver en la pantalla(solo 5 y
32     //no 4000 en panalla) identifier es
33     //el identificador y en index usamos
34     //indexpath
35
36     func tableView(_ tableView: UITableView,
37 heightForRowAt indexPath: IndexPath) ->
38     CGFloat {
39     return 100
40 }
41
42     func tableView(_ tableView: UITableView,
43 didSelectRowAt indexPath: IndexPath) {
44     let viewController =
45         UIStoryboard(name: "Main",
46                     bundle:
47                     nil)
48         .instantiateViewController
49         (identifier: "DetallePerrito")
50
51     override func viewDidLoad() {
52     super.viewDidLoad()
53
54     // Do any additional setup after
55     // loading the view.
56 }
57
58 /*
59 */

```

Xcode interface showing the storyboard and the Detalle View Controller scene selected. The Identity tab shows the Storyboard ID is set to "DetallePerrito".

Instanciar viewcontroller

```

let viewController = UIStoryboard(name: "Main", bundle: nil).instantiateViewController(withIdentifier: "DetallePerrito")as?
DetalleViewController viewController?.nombre="Perrito1"
self.present(viewController!, animated: true, completion:nil)
Creamos una instancia, as le decimos que tipo es, le asignamos lo que le instancias y lo presentamos con ser
@IBOutlet var detalles: UILabel!
var nombre:String="Hola a todos"
override func viewDidLoad() {
super.viewDidLoad()
detalles.text = nombre
// Do any additional setup after loading the view.
}

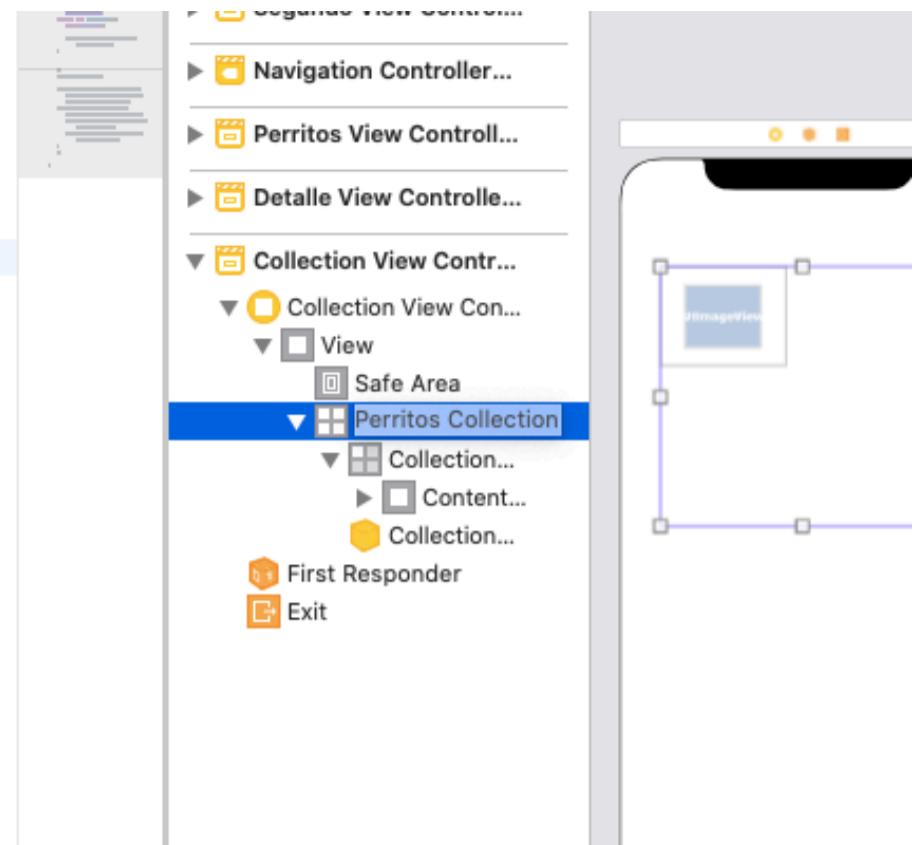
```

tableview.

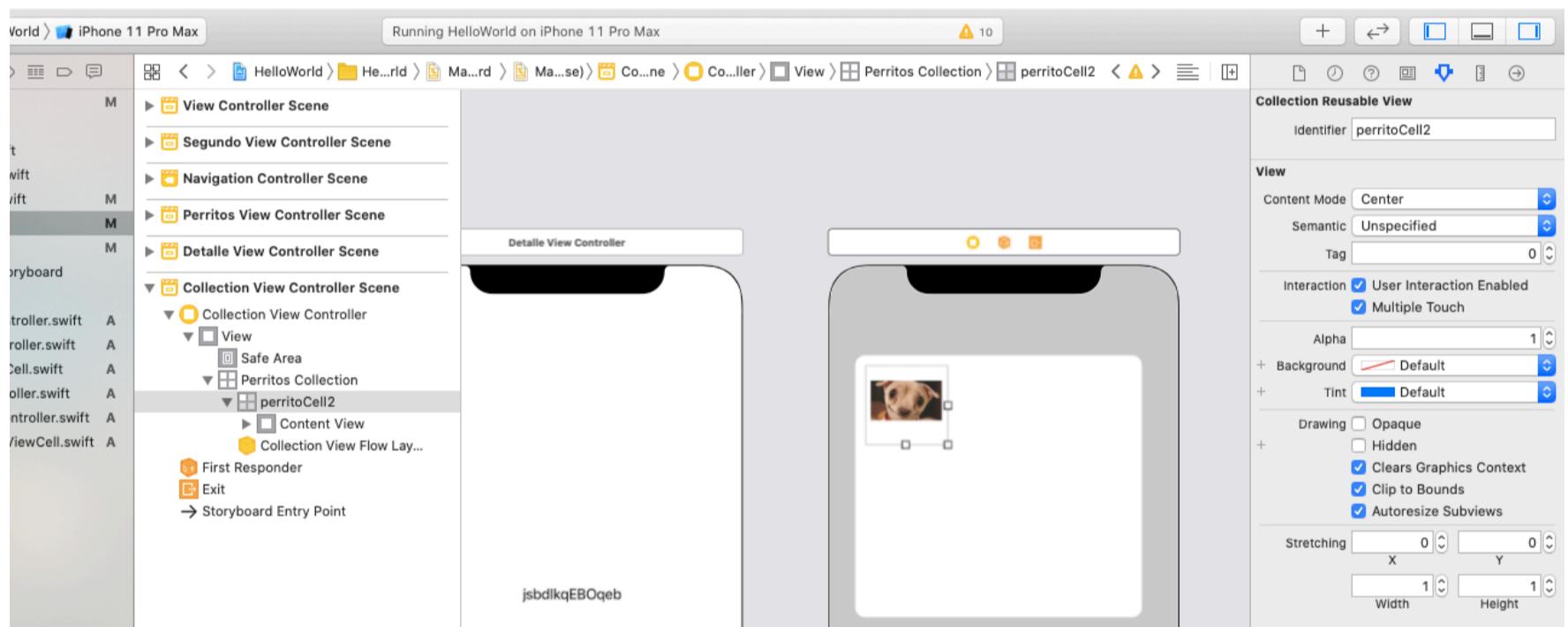
```

4 // 
5 // Created by Pamela Ruiz on 14/12/19.
6 // Copyright © 2019 Rodolfo Benjamin
7 // Alcantara Solorio. All rights reserved.
8 //
9 import UIKit
10
11 class CollectionViewController: UIViewController {
12
13     @IBOutlet var PerritosCollection: UICollectionView!
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17
18         // Do any additional setup after
19         // loading the view.
20
21     /*
22     // MARK: - Navigation

```



arrastramos perritos collection



Manda error en el view controller de el contenedor