

Programación en Móviles Avanzado

LABORATORIO 05

ViewControllers y TableView

CODIGO DEL CURSO:



Alumno(s):	Valladolid Castro Jhon Michael					Nota	
Grupo:				Ciclo: V			
Criterio de Evaluación	Excelente (4pts)	Bueno (3pts)	Regular (2pts)	Requiere mejora (1pts)	No acept. (0pts)	Puntaje Logrado	
Configura archivo Assets para modificar icono de App y reconoce la creación de Segues						3	
Trabaja con multiples vistas, pasando variables entre ellas						3	
Desarrolla adecuadamente los ejercicios propuestos						6	
Realiza observaciones y conclusiones que aporten un opinión crítica y técnica						3	
Se comunica de manera efectiva, trabaja con orden, limpieza y puntualidad.						2	
Evidencia avance en laboratorio						3	

I.- OBJETIVOS:

- Identificar los objetos que se pueden insertar en un **ViewController**
- Interactúa con múltiples **ViewController**
- Identifica el uso de **TableView**

II.- SEGURIDAD:**Advertencia:**

En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.

III.- FUNDAMENTO TEÓRICO:

Revise sus diapositivas del tema antes del desarrollo del laboratorio.

IV.- NORMAS EMPLEADAS:

No aplica

V.- RECURSOS:

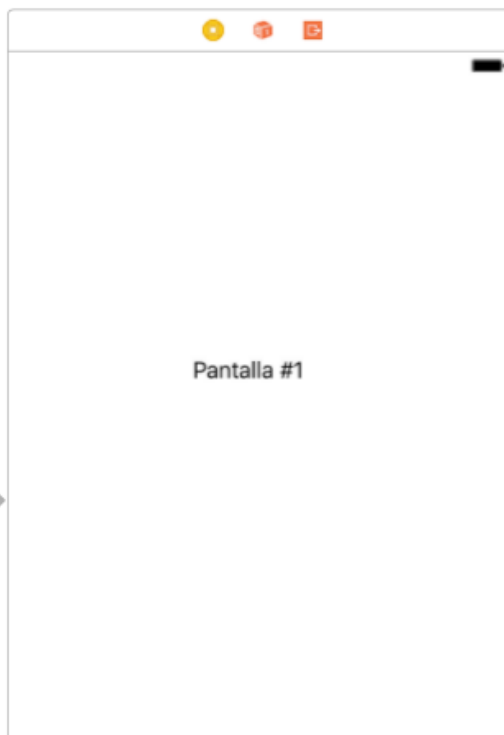
- En este laboratorio cada alumno trabajará con un equipo con MAC OS.

VI.- METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:

- El desarrollo del laboratorio es individual.

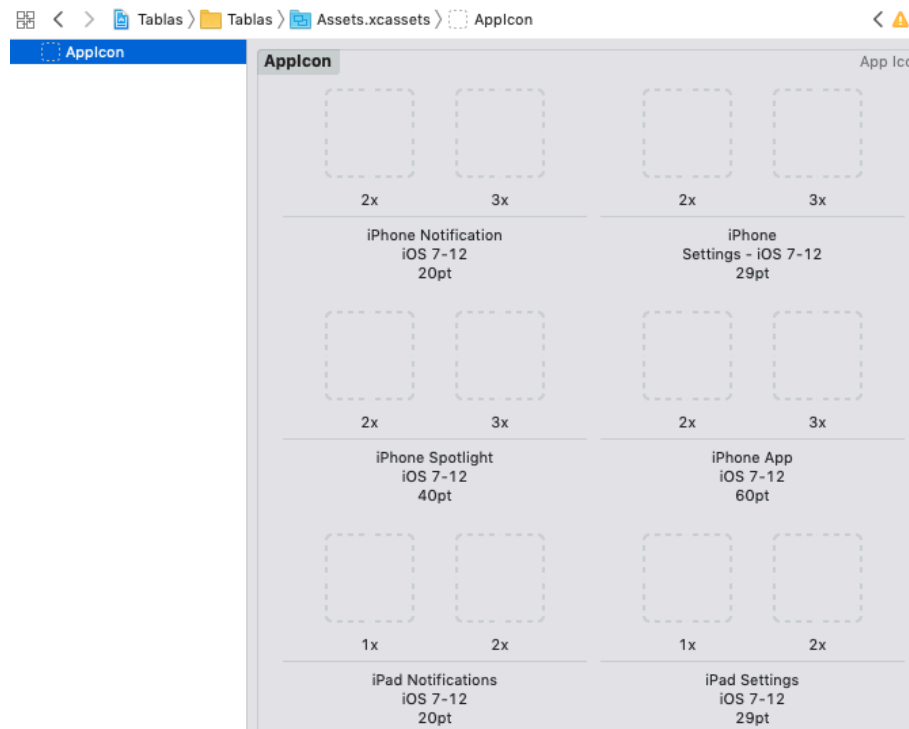
VII.- PROCEDIMIENTO:

1. Abra el software Xcode y haga clic en Create a new Xcode project
 - a. El proyecto debe ser para iOS de tipo **Single View App**
 - b. Product Name: "**TablasSuApellido**".
 - c. Lenguaje: **Swift**.
2. En la única vista creada (**Main.storyboard**), agregue un **label** de texto: "**Pantalla #1**".



3. Descargue una imagen en formato **.png**
4. Ingrese a **App Store** y descargue el programa **App Icon Resizer (AIR)**
OJO: Sino le permite descargar el aplicativo puede utilizar alguna pagina web como <https://hotpot.ai/icon-resizer> y configurar la generar iconos para plataforma iOS.
5. Abra el programa **App Icon Resizer** descargado

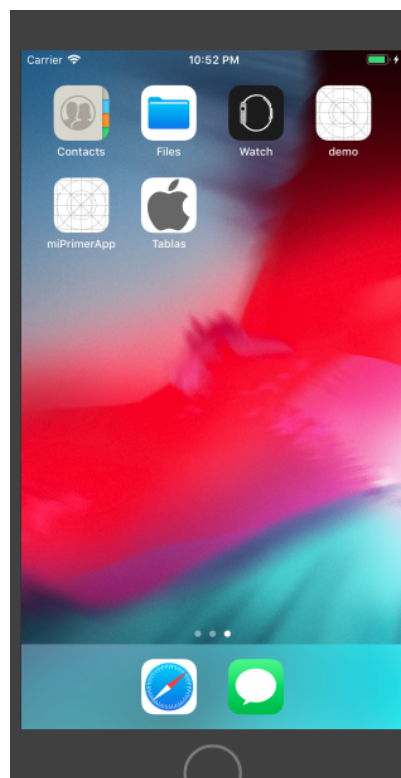
6. Seleccione la opción de Universal y en la parte inferior de **Drag large icon here** arrastre la imagen previamente descargada
7. Se le pedirá una ruta donde se creará una carpeta que contenga el paquete de iconos generados. Selección que la creación sea en **Desktop**
8. Verifique que en **Desktop** se creó la carpeta **AppIcon.appiconset**
9. Dirijase a su proyecto en Xcode y haga clic en la carpeta de **Assets.xcassets** ▢ **App Icon**. Verifique que se muestra al lado izquierdo los distintos tamaños de iconos que se usan para cada tipo de dispositivo que se utilice (iPhone, iPad, etc)



10. Abra la carpeta **AppIcon.appiconset** y seleccione todos los iconos generados, excepto el archivo .json, y arrastre estos iconos sobre su proyecto Xcode, en la lista mostrada por **AppIcon**.

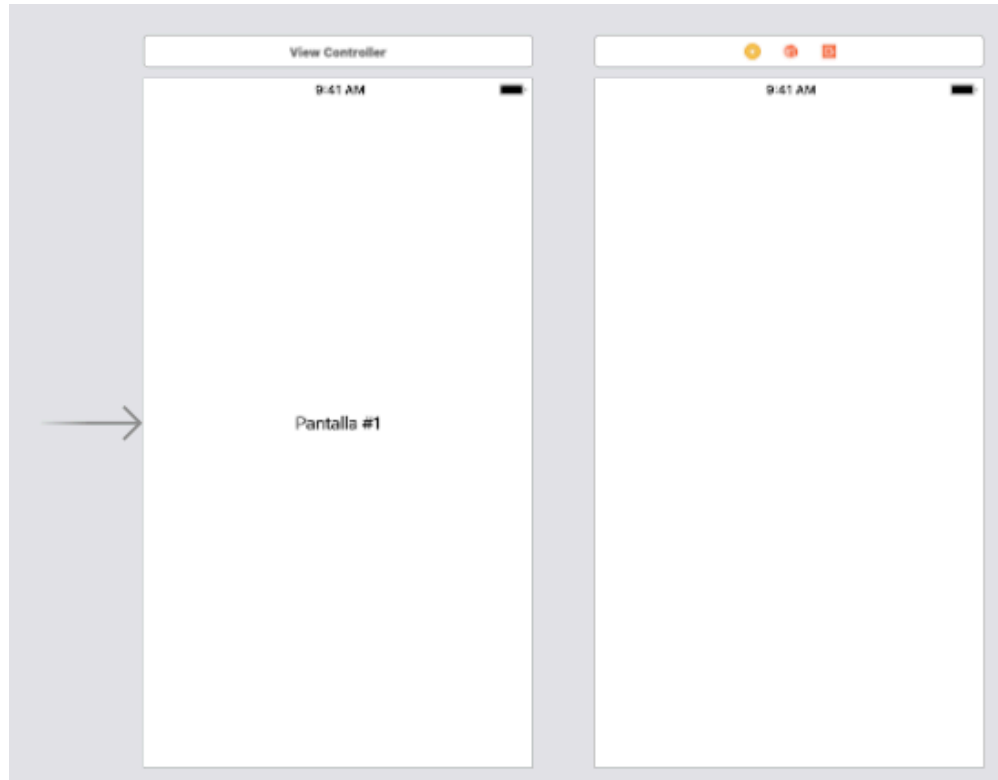


11. Ejecute el proyecto y verifique que se muestra el icono que se agrego



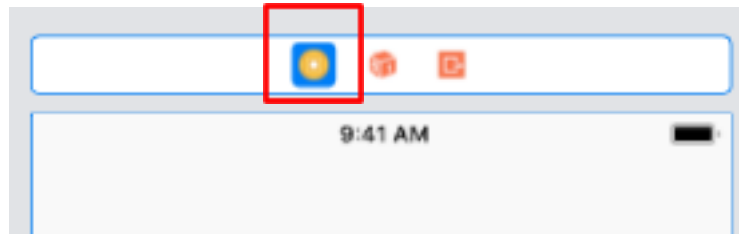
AGREGANDO UNA SEGUNDA PANTALLA

1. Desde la librería de objetos (**Library**) arrastre a **Main.storyboard** un objeto **viewController**

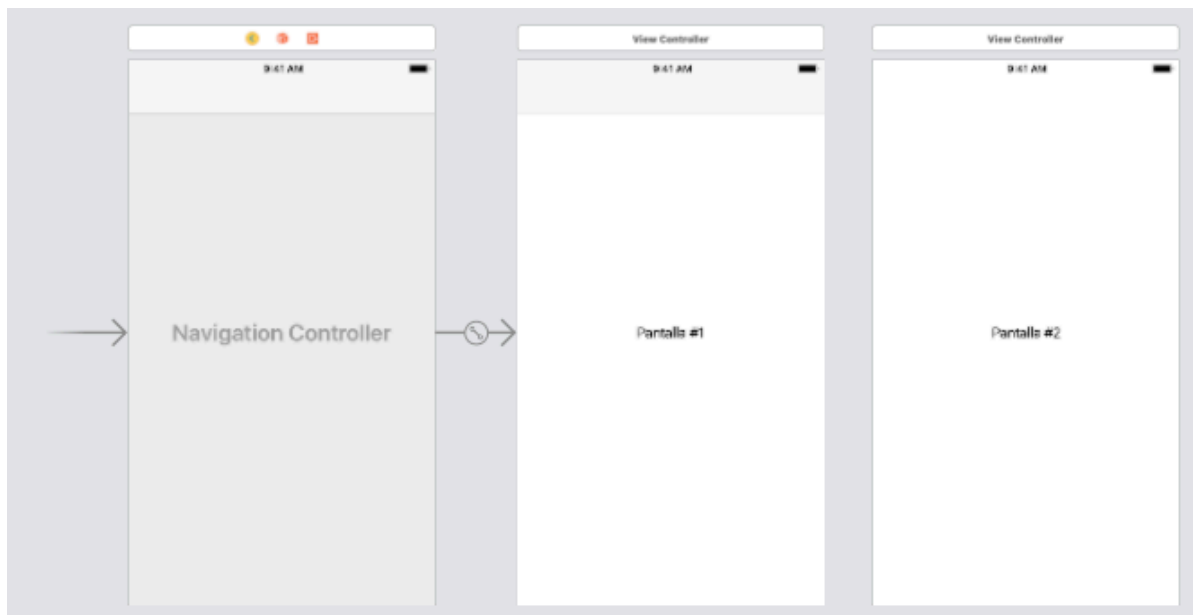


2. Coloque un **Label** con el texto “**Pantalla 2**” en la nueva vista.

3. Seleccione la primera vista y de clic en su botón “**ViewController**”



4. Una vez seleccionado, diríjase al **menú Editor -> Embed In -> Navigation Controller**.
5. Se habrá aumentado una vista de navegación inicial, antes de la primera pantalla.



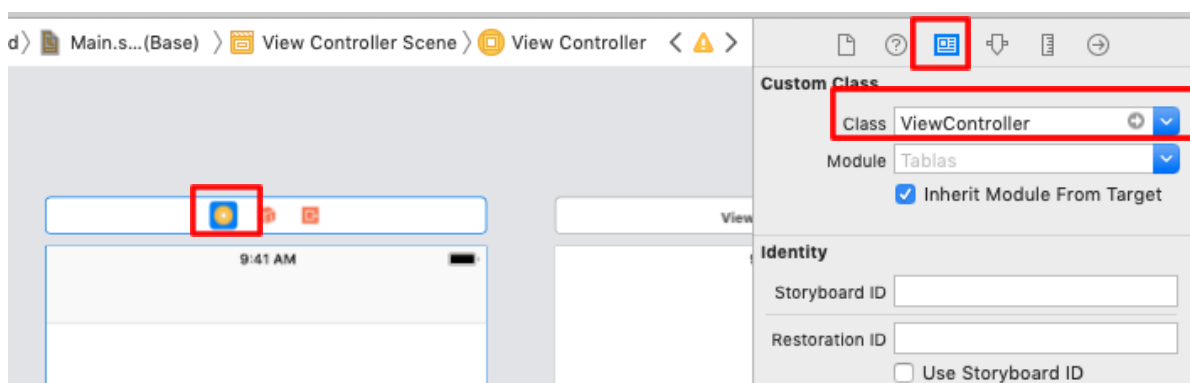
6. Que cambio pudo notar en el diseño de la vista 1

Al realizar la selección de Navigate controller, se reduce el contenido de la pantalla agregando un contenido en blanco en la parte superior una sección de navigation item.

7. Investigue para que sirve un **Navigation Controller**

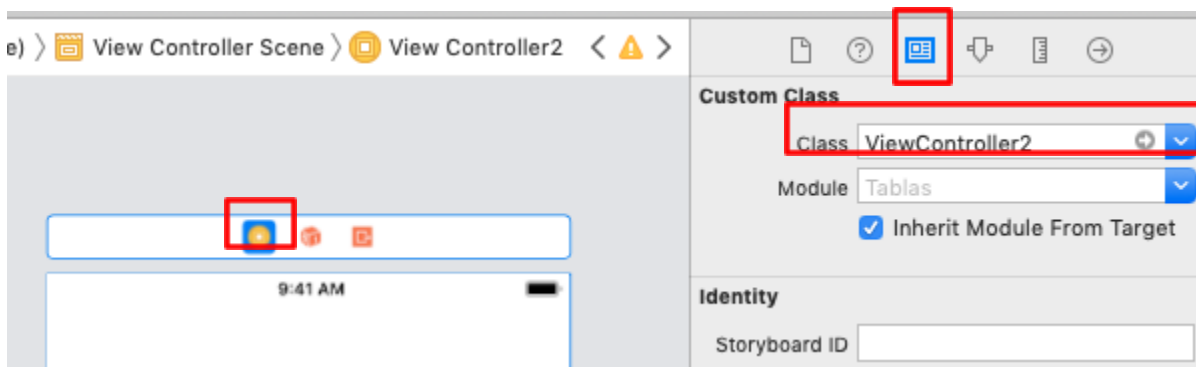
Las navigate controller sirve para poder navegar entre las diferentes vistas de la misma storyboard.

8. Seleccione nuevamente el botón del **ViewController** en la primera vista, y active la pestaña “**Show the Identity Inspector**” en la barra lateral derecha.

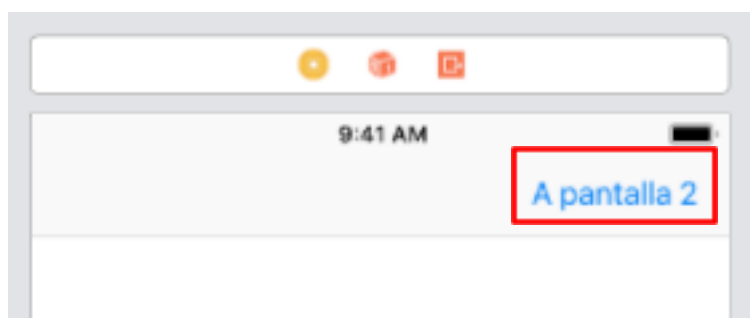


9. Notará que la clase **ViewController**, es la que controla a esta vista(es aquí donde se coloca el código que controlara la vista), deberíamos hacer lo mismo con la nueva vista creada.

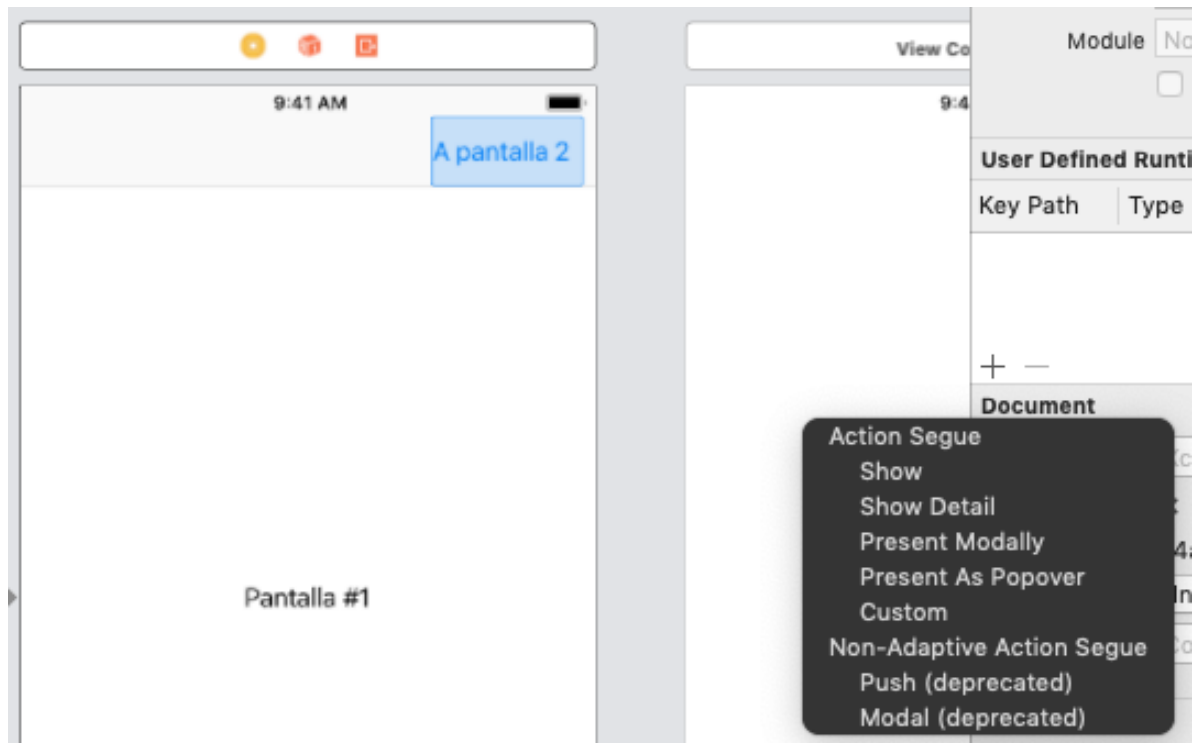
10. Para ello, agregue un nuevo archivo en la carpeta **Tablas**, el archivo debe ser de tipo: **Cocoa Touch** y el nombre del archivo póngalo como **"ViewController2"**, debe ser una subclase de **UIViewController** y con lenguaje **Swift**, además el checkbox de **"Also create XIB file"**, debe estar sin marcar.
11. Ahora seleccione la segunda vista creada.
12. Clic en su botón **"ViewController"**.
13. Acceda a sus propiedades en la pestaña **"Show the Intendity Inspector"**.
14. Como el cuadro estará vacío, elija el archivo **ViewController2**



15. En el cuadro de búsqueda de controles, busque el control **"Bar Button Item"** y arrástrelo hasta la zona indicada en la **vista 1**. A continuación cambie de texto del botón a **"A Pantalla 2"**



16. Ahora mantenga presionada la tecla **Ctrl** y con el **clic presionado arrastre el enlace hasta la pantalla 2**.
17. Aparecerá un menú contextual en el que deberá elegir la opción **"Show"**.



18. Investigue para que sirve las opciones de **Show, Show Detail, Present Modaly, Present As Popover** para que sirve **Show, Show Detail, Present Modaly, Present As Popover** en Swift?
19. Ejecute la aplicación y pruebe su funcionamiento. Verifique que las pantallas cambian.
20. Indique los detalles mas importantes de la ejecución del proyecto

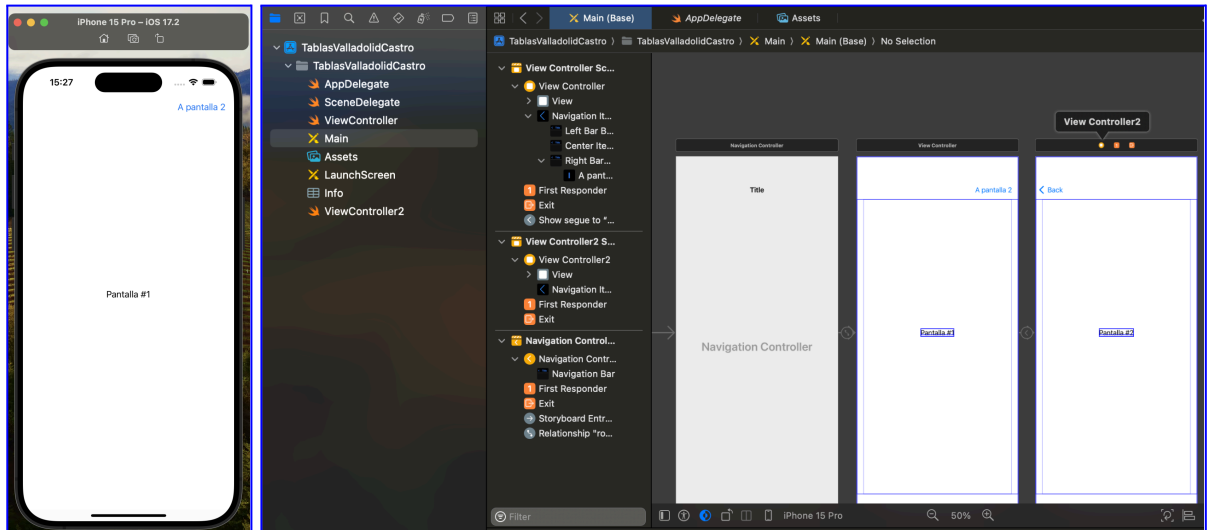
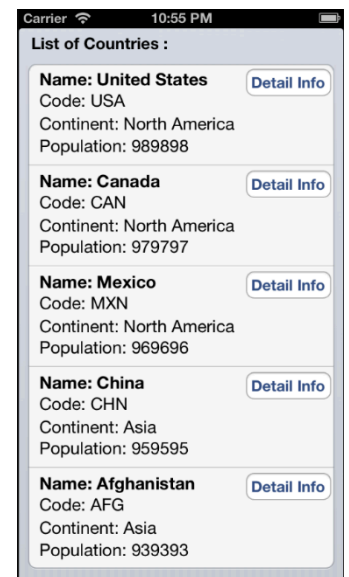
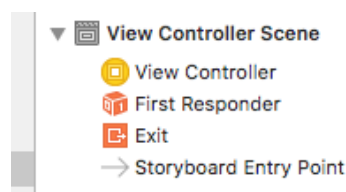


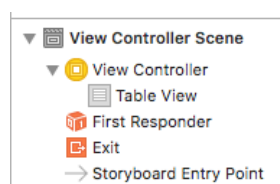
TABLE VIEW

En esta sección se hará uso de la **TableView**, que es una forma de representar varias filas de forma horizontal en una aplicación. Similar a un **ListView** en Android.

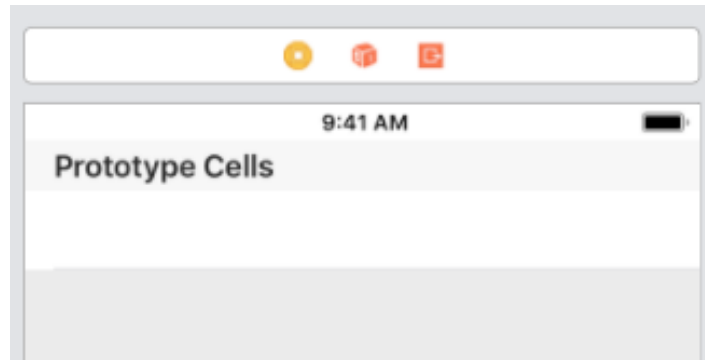
1. Cree un nuevo proyecto en Xcode
2. Tipo **Single View App**
3. Nombre **TableView**
4. Lenguaje **Swift**
5. Ingrese al archivo **Main.storyboard**.
6. Seleccione el contenedor principal de la aplicación "**view**", y presione suprimir o elimínela. El **view Controller** debería quedar vacío:



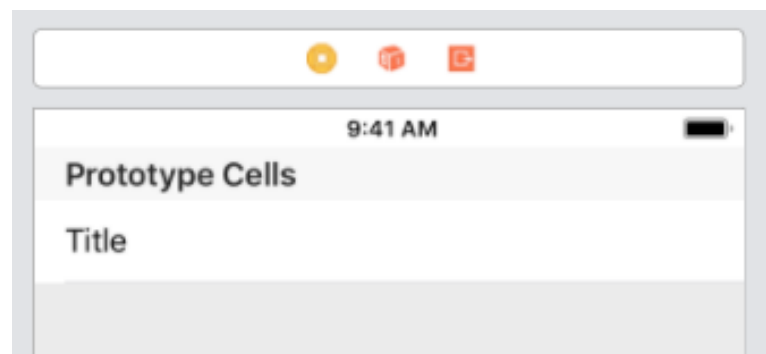
7. En la librería de objetos (Library), busque el control "**Table View**" y arrástrelo al **View Controller** para que ahora sea un **Table View**. Ahora el **View Controller**, debería contener a un control **Table View**:



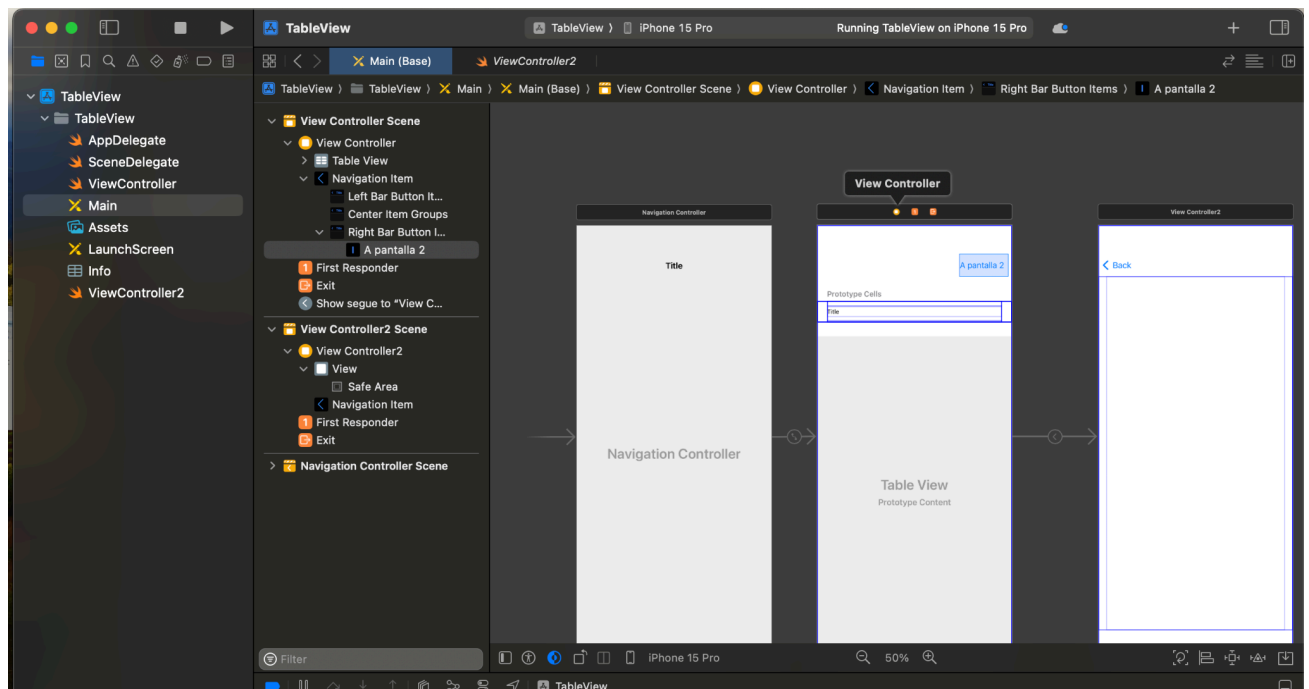
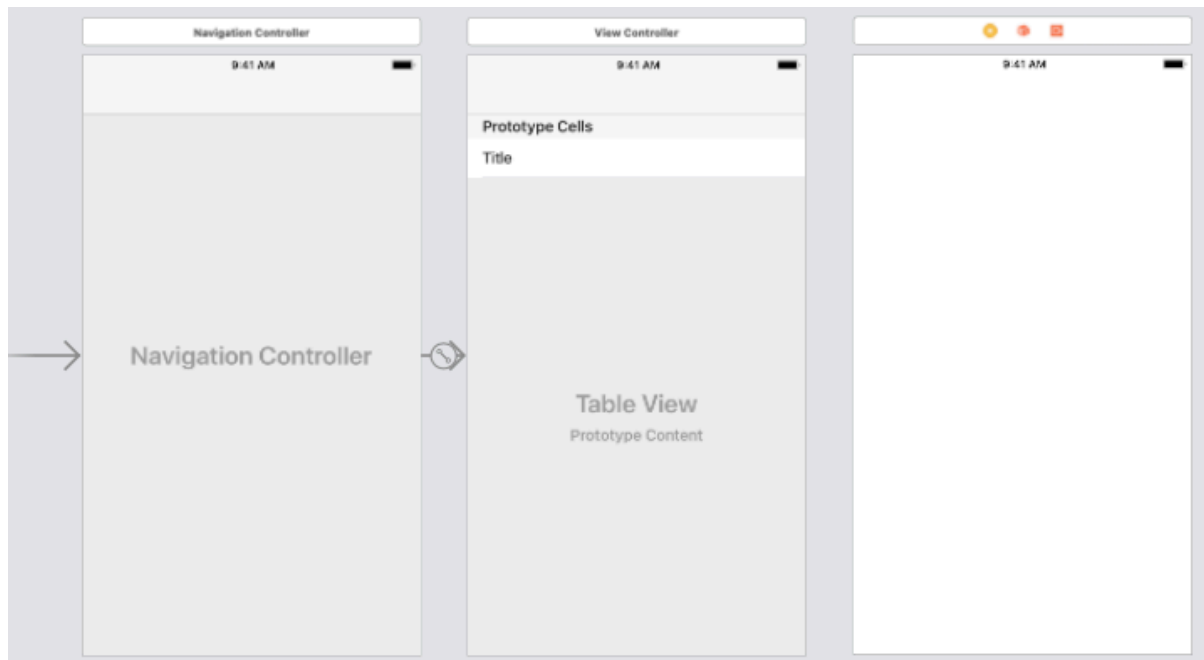
8. Ahora clic en el **Table View**, recién creado. Ingrese al **inspector de atributos** del control y modifique el atributo:
 - a. **Prototype Cells : 1**
9. Se creará una celda física:



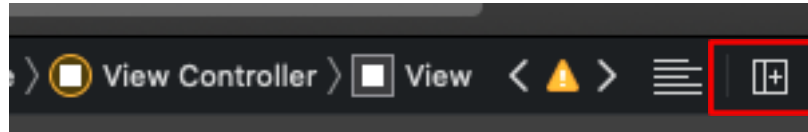
10. En el **View Controller Scene**, clic en “**Table View Cell**” para seleccionar la celda recién creada.
11. Modifique su atributo:
 - a. **Style** : Basic
 - b. **Identifier**: Cell
12. Se visualizará de la siguiente manera:



13. Ahora, repita los pasos desde el **1** al **12** de la sección **AGREGANDO UNA SEGUNDA PANTALLA**. Obvie el paso en el que sea crea un Label en el segundo View Controller (solo obvie la creación del Label).
14. Debería tener un resultado como el siguiente (Incluido el archivo **ViewController2**, asociado al **segundo ViewController**)



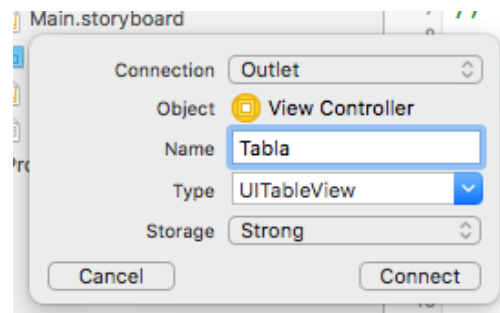
15. Seleccione la primera vista (donde se encuentra el tableView) y Haga clic en el botón **Add Editor on Right** . En el lado derecho para ganar espacio puede optar por cerrar el editor de Scenes



16. Verificara que el archivo **main.storyboard** se visualiza ahora en dos secciones. En el lado derecho, en la nueva sección creada seleccione en el **Jump Bar** el archivo **viewController.swift** (archivo asociado a la vista correspondiente)



17. Debe mostrarse a lado izquierdo las **vistas** y al lado derecho el archivo **.swift**
18. Seleccione el **TableView** y haga clic sin soltar, y presione la tecla **Ctrl**. Arrastre el puntero hasta sección del código mostrado al lado derecho, y suelte debajo de la línea de **Class**
19. Cree una **conexión de tipo Outlet** del **TableView** para su **ViewController(ViewController.swift)** con las siguientes características:



20. Verifique que se creó un **@IBOutlet**

```
9  import UIKit
10
11  class ViewController: UIViewController {
12
13      @IBOutlet var Tabla: UITableView!
14      override func viewDidLoad() {
```

21. Investigue para que sirven las conexiones tipo **Outlet**

Un outlet permite que, al presionar ese botón, el texto cambie o que aparezca un mensaje. Es como un enlace que te deja controlar lo que sucede en la pantalla desde detrás.

22. En código, ingrese al archivo **ViewController** y modifique su herencia e implementación de protocolos de la siguiente manera:

```
class ViewController: UIViewController, UITableViewDelegate,
UITableViewDataSource{
```

23. Implemente las funciones que pertenecen a los protocolos implementados para poder controlar al **Table View** de la siguiente manera:

```
class ViewController: UIViewController, UITableViewDelegate,
UITableViewDataSource{

    @IBOutlet var Tabla: UITableView!
    override func viewDidLoad() {
        super.viewDidLoad()
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSectionSection
section: Int) -> Int {
        code
    }

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

24. Investigue que utilidad cumplen las funciones creadas

25. Modifique el código de las funciones implementadas de la siguiente manera

```
class ViewController: UIViewController,UITableViewDelegate,
UITableViewDataSource{

    @IBOutlet var Tabla: UITableView!
    override func viewDidLoad() {
        super.viewDidLoad()
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSectionSection
section: Int) -> Int {
        return 2
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {
        let celda = UITableViewCell()
        celda.textLabel?.text = "Celda 1"
        return celda
    }
}
```

26. Finalmente en la función **viewDidLoad()**, agregue las siguientes dos líneas que van a servir para vincular las funciones implementadas con el objeto con referencia de tipo **Outlet** que se ha creado. Es decir, ambas funciones van a trabajar ahora con el **Table View** que se ha creado.

```
class ViewController: UIViewController,UITableViewDelegate,
UITableViewDataSource{

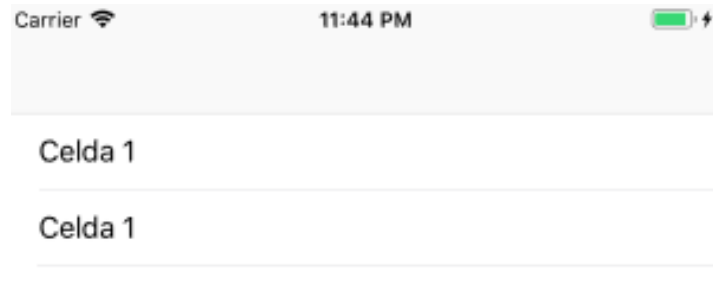
    @IBOutlet var Tabla: UITableView!
    override func viewDidLoad() {
        super.viewDidLoad()

        Tabla.dataSource = self
        Tabla.delegate = self
    }

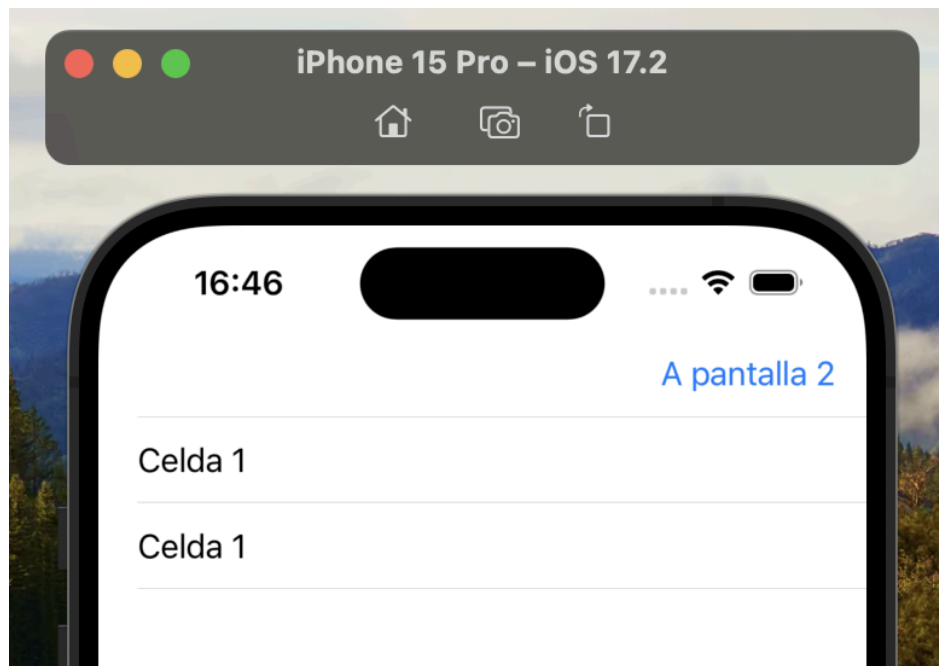
    func tableView(_ tableView: UITableView, numberOfRowsInSectionSection
```

NOTA: Se coloca **self**, para indicar que la misma clase en la que se está trabajando "**ViewController**", contiene las funciones que van a controlar al control **TableView**.

El resultado debería ser el siguiente:



Adjunte una captura de la ejecución de su proyecto y comente los detalles más importantes



NOTA: La impresión se da en dos líneas del Table View ya que en la función “**numberOfRowsInSection**” (la primera función implementada como parte del protocolo), se está retornando el valor de 2; Lo cual significa que se está inicializando al Table View con 2 filas. Una vez que se procedió a retornar ese valor numérico, la segunda función “**cellForRowAt**” con su parámetro de entrada **indexPath**, se ejecuta la cantidad de veces como número de filas se haya colocado en la primera función mencionada. En este caso, 2 veces se escribe la palabra **Celda 1** .

De

hecho, la variable entera **indexPath**, es la que contiene el número de fila actual, es decir, si la función se ejecuta 2 veces, en la primera vez la variable en mención tendrá el valor de 0 y en la siguiente el valor de 1.

27. Cree un **arreglo de Strings** como variable de clase:

```
@IBOutlet var Tabla: UITableView!
```

```
var contenidoCeldas = ["Usando", "TableView", "en", "aplicativo", "iOS"]
```

```
override func viewDidLoad() {
```

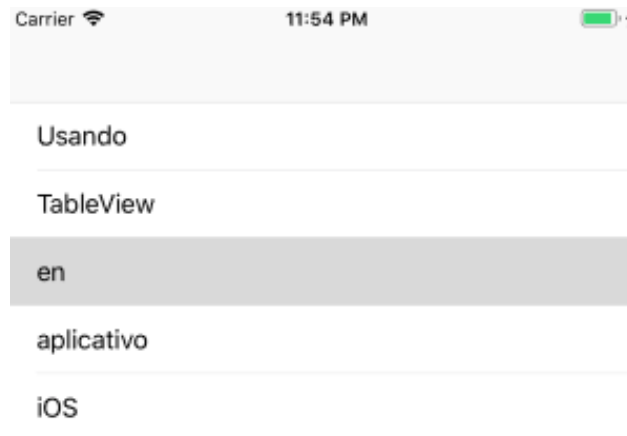
28. En la primera función implementada(**numberOfRowsInSection**), retorne la cantidad de elementos del arreglo creado:

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

29. En la segunda función implementada(**cellForRowAt**), coloque como texto de celda el elemento del arreglo correspondiente:

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
    UITableViewCell {
    let celda = UITableViewCell()
    celda.textLabel?.text = contenidoCeldas[indexPath.row]
    return celda
}
```


30. Ejecute el proyecto y verifique su funcionamiento



31. Adjunte capturas de la ejecución de su proyecto y comente los detalles mas importantes



CONSTRUCTOR O INIT DE TABLEVIEWCELL

1. Para ver las funciones ya establecidas de un **TableViewCell** en la segunda función creada, mantenga presionada la tecla **Command** o **Windows** ya sea en **MAC** o **Windows** respectivamente, y de clic a la clase **UITableViewCell**, elija **Jump to Definition**. Se abrirá la clase de la implementación de dicho objeto.

```
@available(iOS 3.0, *)  
public init(style: UITableViewCell.CellStyle, reuseIdentifier: String?)  
  
public init?(coder aDecoder: NSCoder)
```

2. Notará que uno de sus constructores recibe un estilo y un **identificador** en forma de **String** para asociarlo a un control ya creado en **Storyboard**.
3. Modifique el código de la creación del objeto Celda en **ViewController.swift** para que se cree con los indicadores en mención:

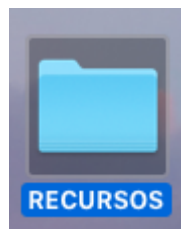
```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->  
    UITableViewCell {  
    let celda = UITableViewCell(style: UITableViewCell.CellStyle.default,  
        reuseIdentifier: "Celda")  
    celda.textLabel?.text = contenidoCeldas[indexPath.row]  
    return celda  
}
```

Compruebe que la aplicación no haya modificado su funcionamiento. Y adjunte capturas



DESCARGANDO RECURSOS

1. Cree una carpeta llamada **RECURSOS**.



2. Dentro de esta carpeta coloque 3 documentos PDF con los siguientes nombres

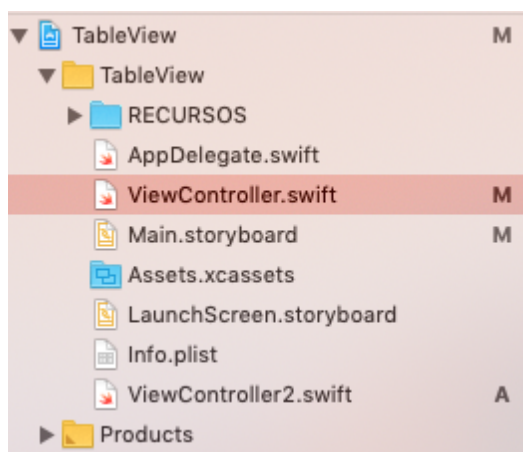
Nombre	Fecha...cación	Tamaño	Clase
recurso1	08:35	43 KB	Documento PDF
recurso2	08:38	91 KB	Documento PDF
recurso3	08:38	91 KB	Documento PDF

3. Arrastre la carpeta al proyecto como una subcarpeta de **TableView**. Y considere la siguiente configuración.

Destination: ☒ Copy items if needed

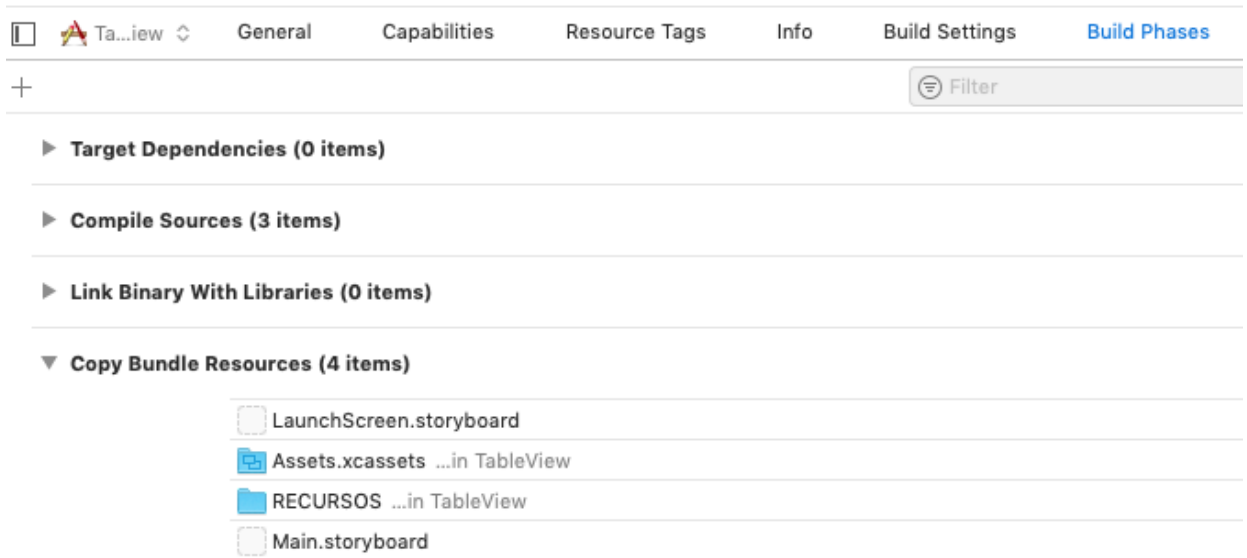
Added folders: ☐ Create groups
☒ Create folder references

Add to targets: ☒ TableView



NOTA: Para verificar que los archivos hayan sido copiados de manera correcta (porque cuando se trabajan con muchos archivos, existe la posibilidad de errores en uno que otro archivo), clic en el nombre del proyecto y seleccione la pestaña **Build Phases**, en la última categoría **Copy Bundle Resources** deberían figurar los archivos copiados.

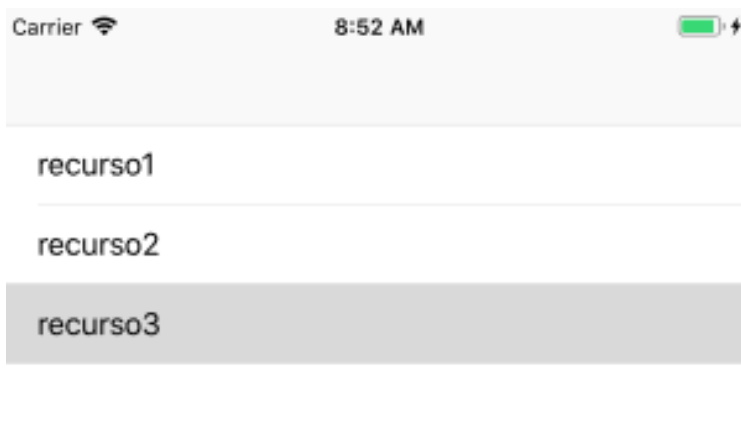
Si todos los archivos figuran en el **Bundle** del Proyecto, al momento de instalar la aplicación en cualquier dispositivo, los archivos serán copiados en el dispositivo sin ningún problema.



4. Ingrese al **ViewController** Principal y modifique el arreglo creado para que contenga los elementos: "recurso1", "recurso2", "recurso3".
5. Implemente la función **didSelectRowAt** que es propia del elemento **TableView** como parte del protocolo implementado en la clase **ViewController** con el siguiente código:


```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    print("Hizo clic en la fila: \(indexPath.row)")
}
```

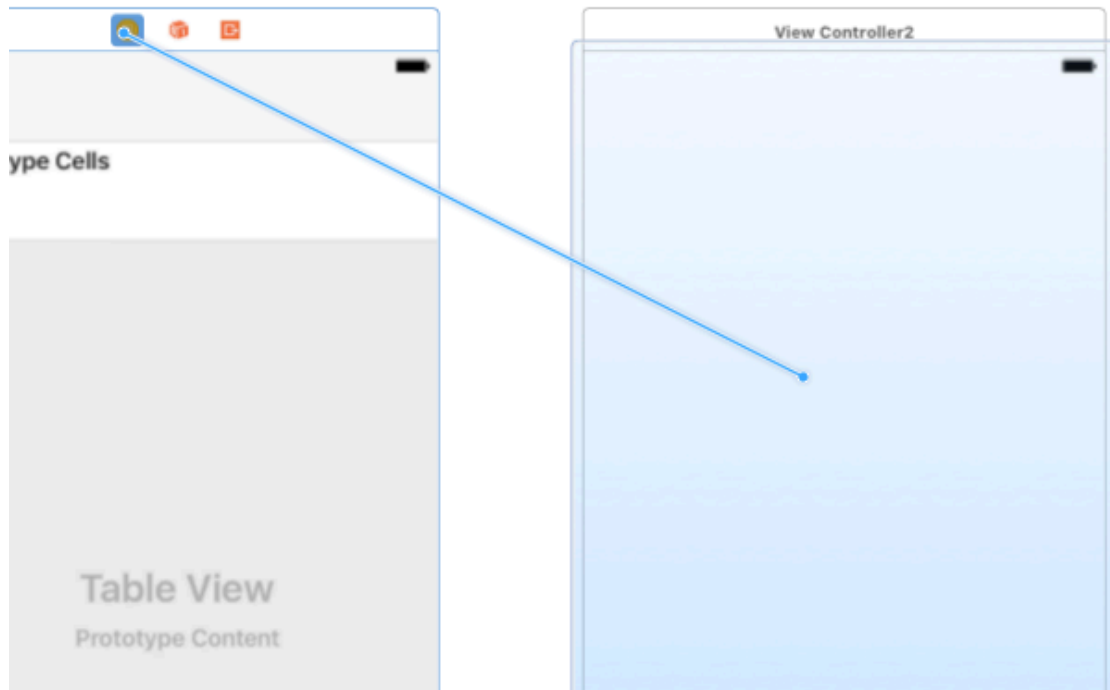
6. Ejecute la aplicación y compruebe que al presionar una de las **filas del TableView** se impriman los mensajes en consola de la siguiente manera:



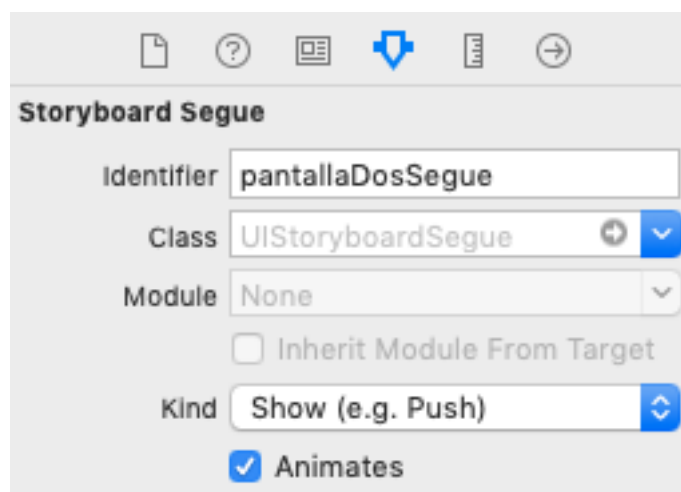
```
Hizo clic en la fila: 0
Hizo clic en la fila: 1
Hizo clic en la fila: 2
```

7. Cree un **Segue** del **ViewController** principal a **ViewController2**. Para ello, mantenga presionada la tecla

Ctrl en el ícono del **ViewController**  y arrástrelo hasta el **ViewController2**



8. En el menú contextual elija la opción **Show**.
9. Se creará el **Segue**.
10. Clic en el **Segue** creado y en sus atributos coloque el valor "**pantallaDosSegue**" en su atributo **Identifier**.



11. En **ViewController.swift**, modifique la función que obedece a la selección de un ítem dentro del **TableView** (**didSelectRowAt**) de la siguiente manera:

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    //print("Hizo clic en la fila: \(indexPath.row)")
    let recursoSeleccionado = indexPath.row
    self.performSegue(withIdentifier: "pantallaDosSegue", sender:
        recursoSeleccionado)
}
```

12. Ingrese al **Main.storyboard** y coloque un **Label** en el **ViewController2** con un texto **"Pantalla 2"**.
13. Cree una **conexión de tipo outLet** del **label** creado en **ViewController2** como variable de clase con el nombre de **"labelTitulo"**.
14. En **ViewController2**, cree una variable de clase de tipo **String Optional** de nombre **recursoRecibido** justo después de la conexión Outlet.
15. En la función **viewDidLoad()** del **ViewController2** coloque el siguiente código:

```
@IBOutlet weak var labelTitulo: UILabel!
var recursoRecibido:String?

override func viewDidLoad() {
    super.viewDidLoad()
    labelTitulo.text = recursoRecibido!
    // Do any additional setup after loading the view.
}
```

16. Regrese al **ViewController principal** y cree la **función prepare** de la siguiente manera:

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "pantallaDosSegue"{
        let idrecursoRecibido = sender as! Int
        let pantalla2:ViewController2 = segue.destination as! ViewController2
        pantalla2.recursoRecibido = contenidoCeldas[idrecursoRecibido]
    }
}
```

17. Ejecute la aplicación y verifique que la segunda pantalla reciba el nombre del elemento seleccionado en el **TableView**.

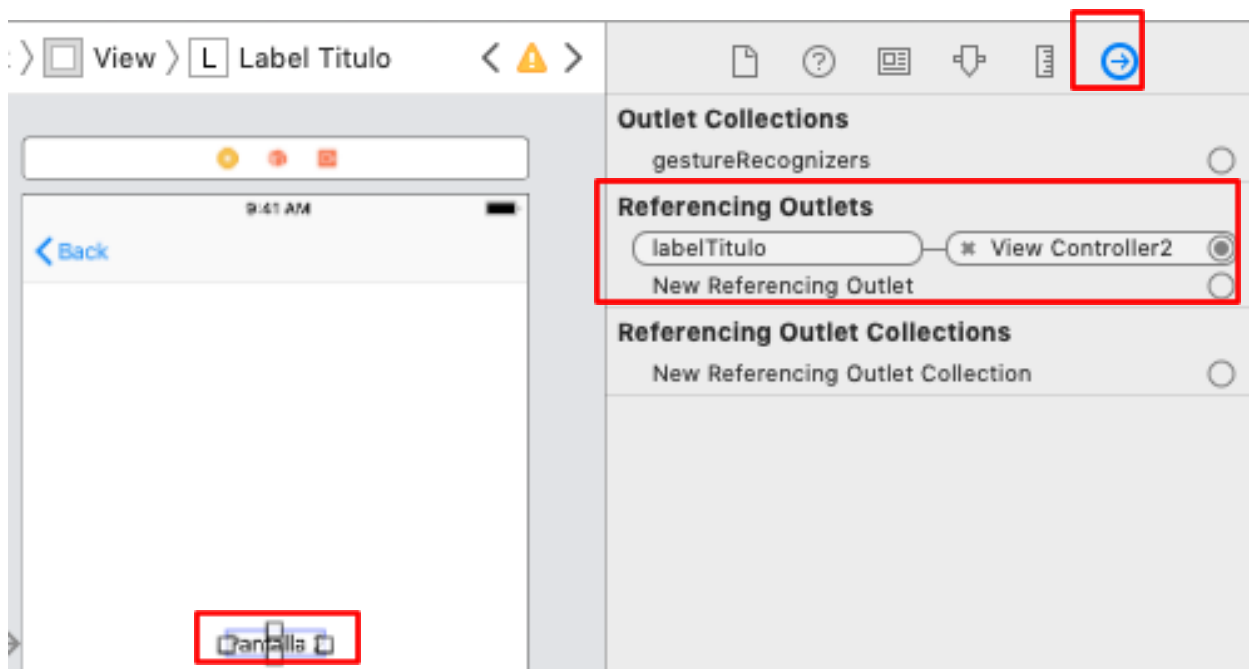
[< Back](#)



recurso1

18. Indique los detalles mas importantes de la ejecución del proyecto

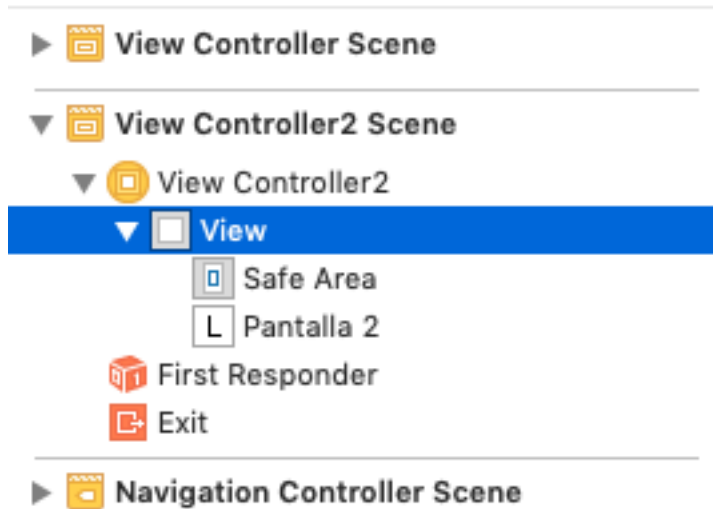
19. En el **inspector de conexiones** para **ViewController2**, elimine la **conexión Outlet** que la **etiqueta** del **ViewController2** tiene con su respectiva vista, luego elimine la etiqueta y el código que relacione a dicha etiqueta.



```
class ViewController2: UIViewController {  
  
    var recursoRecibido:String?  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
    }  
}
```

WEBVIEW

1. Seleccione la segunda vista y en el cuadro de escenas elimine la vista **VIEW** del **ViewController2** presionando la tecla **Suprimir**.



2. El **ViewController2**, debería quedar vacío.
3. En el **explorador de objetos** busque la vista **WebView** y arrástrelo al **ViewController2**
4. Cree una conexión de tipo **Outlet** de la **vista web** creada al archivo **ViewController2.swift** asigne como nombre de variable **"vistaWeb"**
5. En el **ViewController2** cree una función de nombre **"mostrarRecurso"** e inyecte el siguiente código para crear la ubicación del recurso a leer.

```
func mostrarRecurso(){
    let direccionRecurso = URL(fileURLWithPath:
        Bundle.main.path(forResource: recursoRecibido!, ofType: "pdf",
            inDirectory: "RECURSOS")!)
}
```

6. Ahora aumente el código para transformar la dirección creada en el tipo de dato **DATA**.

```
func mostrarRecurso(){
    let direccionRecurso = URL(fileURLWithPath:
        Bundle.main.path(forResource: recursoRecibido!, ofType: "pdf",
            inDirectory: "RECURSOS")!)
    let datos = try? Data(contentsOf: direccionRecurso)
}
```

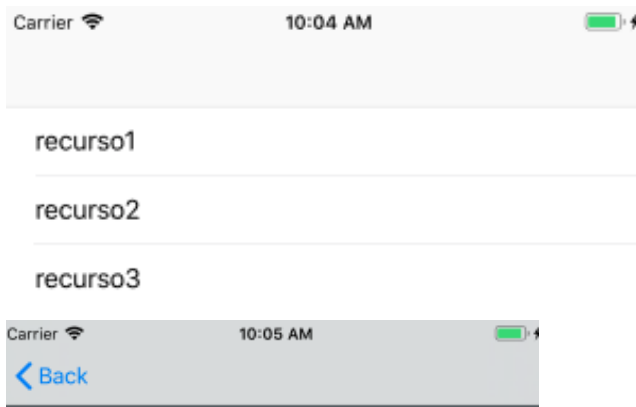
7. Cargue el documento PDF en el **WebView**:


```
func mostrarRecurso(){
    let direccionRecurso = URL(fileURLWithPath:
        Bundle.main.path(forResource: recursoRecibido!, ofType: "pdf",
            inDirectory: "RECURSOS")!)
    let datos = try? Data(contentsOf: direccionRecurso)
    vistaWeb.load(datos!, mimeType: "application/pdf", textEncodingName: "utf-8",
        baseUrl: direccionRecurso)
}
```

8. Asegúrese de llamar a la función **mostrarRecurso()** desde la función principal **viewDidLoad()**

```
override func viewDidLoad() {
    super.viewDidLoad()
    mostrarRecurso()
}
```

9. Pruebe la aplicación para ver el contenido de los documentos PDF



10. Para terminar el proyecto, agregue la función **habilitarZoom** y llámela de la función **didViewLoad()** para habilitar el zoom en los archivos PDF.

```
func habilitarZoom(){  
    vistaWeb.scalesPageToFit = true  
}  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
    mostrarRecurso()  
    habilitarZoom()  
}
```

11. Vuelva a ejecutar el proyecto y Anote los detalles más importantes sobre los pasos anteriores.

TAREA:

1. Desarrolle una aplicación que permita mostrar una lista de países con destinos turísticos. Al seleccionar un país debe dirigir a otra vista donde muestre algunos destinos turísticos presentes en dicho país, al seleccionar un destino X, debe mostrar una imagen del destino.
2. Debe permitir volver al viewController anterior

OBSERVACIONES (5 mínimo):

(Las observaciones son las notas aclaratorias, objeciones y problemas que se pudo presentar en el desarrollo del laboratorio)

CONCLUSIONES (5 mínimo):

(Las conclusiones son una opinión personal sobre tu trabajo, explicar como resolviste las dudas o problemas presentados en el laboratorio. Además de aportar una opinión crítica de lo realizado)