



POLITÉCNICA

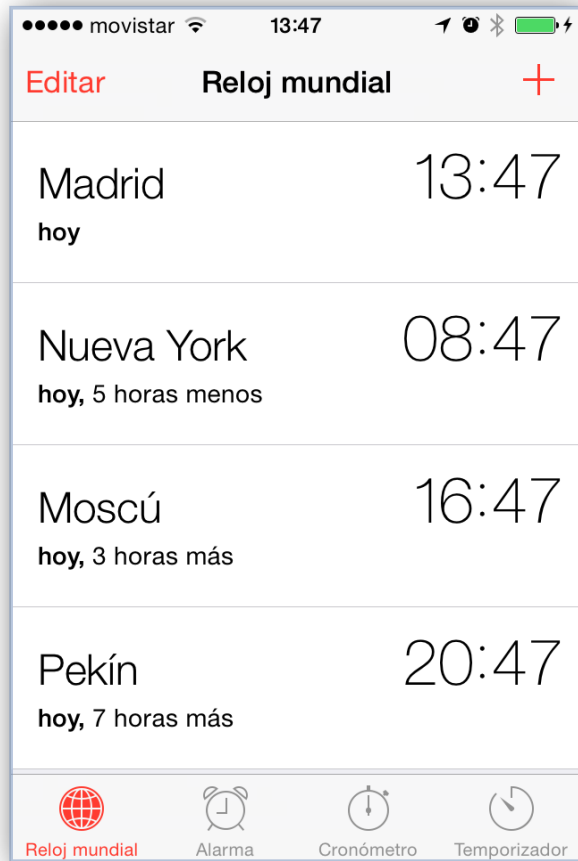
ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS Tab Bar Controller

IWEB 2015-2016  
Santiago Pavón

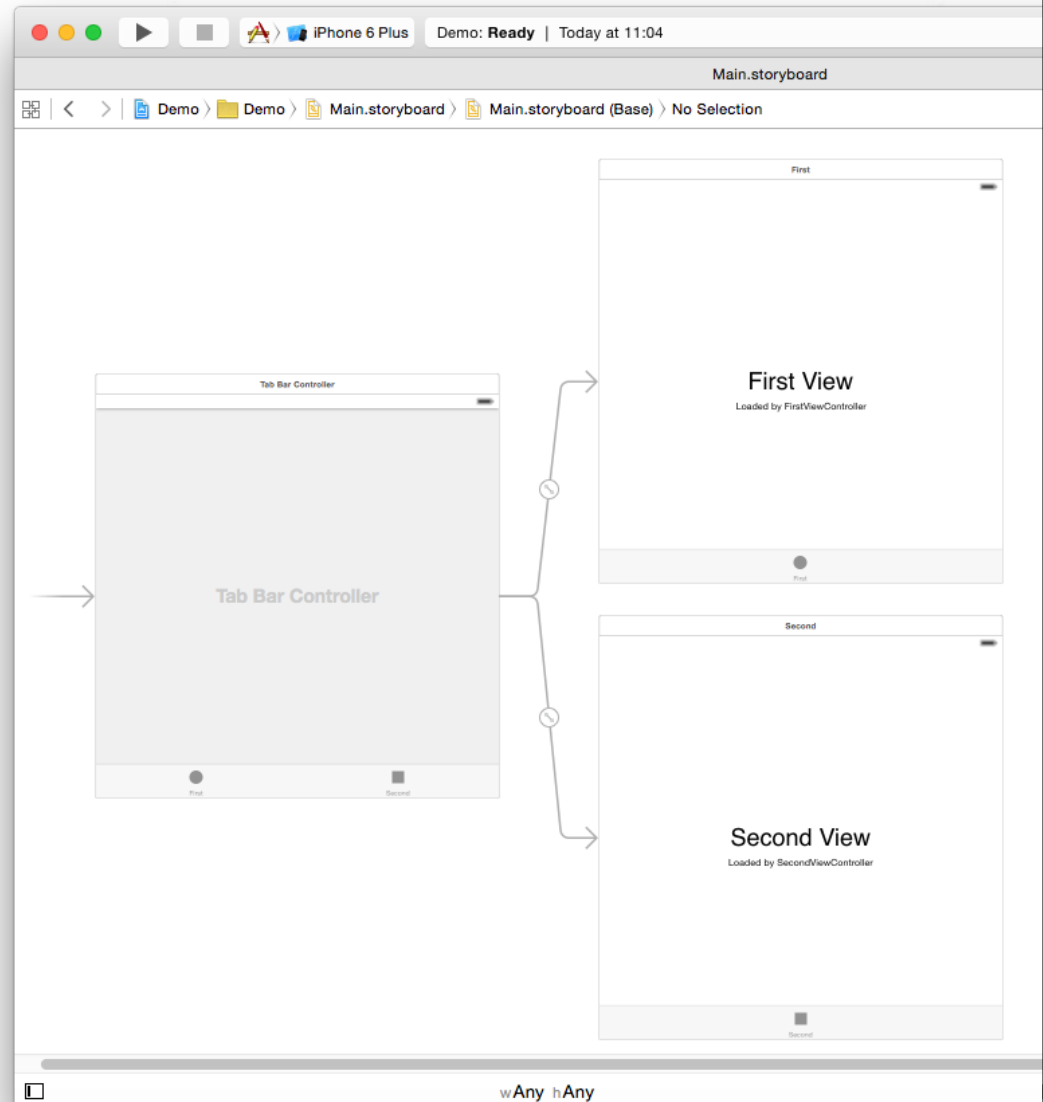
ver: 2015.10.30



La barra de pestañas inferior controla el VC mostrado

# Plantillas de Xcode

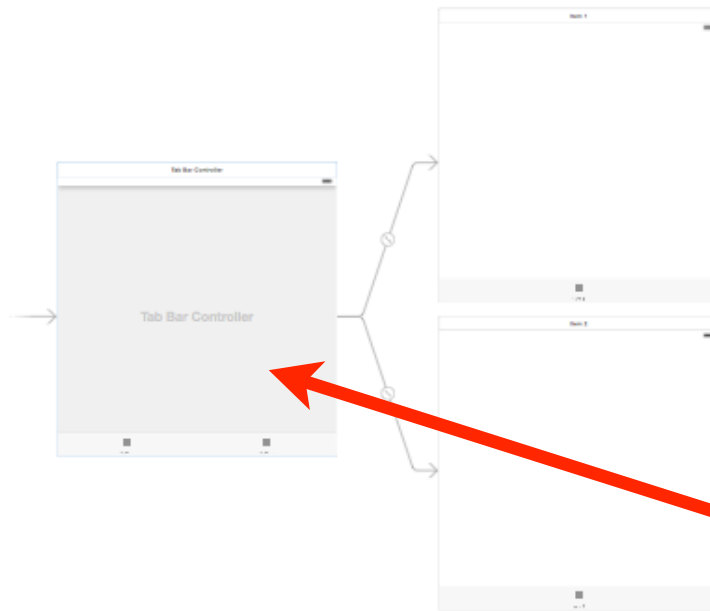
- Xcode tiene una plantilla para crear un proyecto nuevo usando un Tab Bar Controller.
  - Plantilla **Tabbed Application**.
- Esta plantilla crea:
  - Un storyboard con:
    - Un Tab Bar Controller.
    - Dos View Controller.
  - Ficheros .swift de las clases derivadas de UIViewController para los VC creados.
  - Iconos, ficheros de soporte, ...



# Storyboard

- Interface Builder permite editar los ficheros storyboard para:
  - Incluir Tab Bar Controllers.
    - Arrastrando un objeto **Tab Bar Controller** desde la biblioteca de objetos.
      - Se crea un objeto Tab Bar Controller y dos objetos VC.
        - Si queremos seguir trabajando usando los VC creados, tendremos que usar el inspector para indicar cual es la clase real de estos VC.
        - También podemos borrar los dos VC creados y enganchar el Tab Bar Controller con otros VC ya existentes.
      - O seleccionando un VC existente en el storyboard y ejecutando:
        - Menú Editor > Embed In > Tab Bar Controller
  - Incluir un VC ya existente a un Tab Bar Controller ya existente.
    - Creando un segue Relationship desde el Tab Bar Controller hasta el VC.
  - Editar el título, el icono, el badge, ... de los Tab Bar Items que se muestran para cada VC añadido a un Tab Bar Controller.

Arrastrar un Tab Bar Controller desde la biblioteca de objetos. Se crean también dos objetos VC.



#### Simulated Metrics

- Size
- Orientation
- Status Bar
- Top Bar
- Bottom Bar

#### View Controller

- Title
- Is Initial View Controller
- Layout  Adjust Scroll View Insets
- Hide Bottom Bar on Push
- Resize View From NIB
- Use Full Screen (Depre...
- Extend Edges  Under Top Bars
- Under Bottom Bars
- Under Opaque Bars
- Transition Style
- Presentation

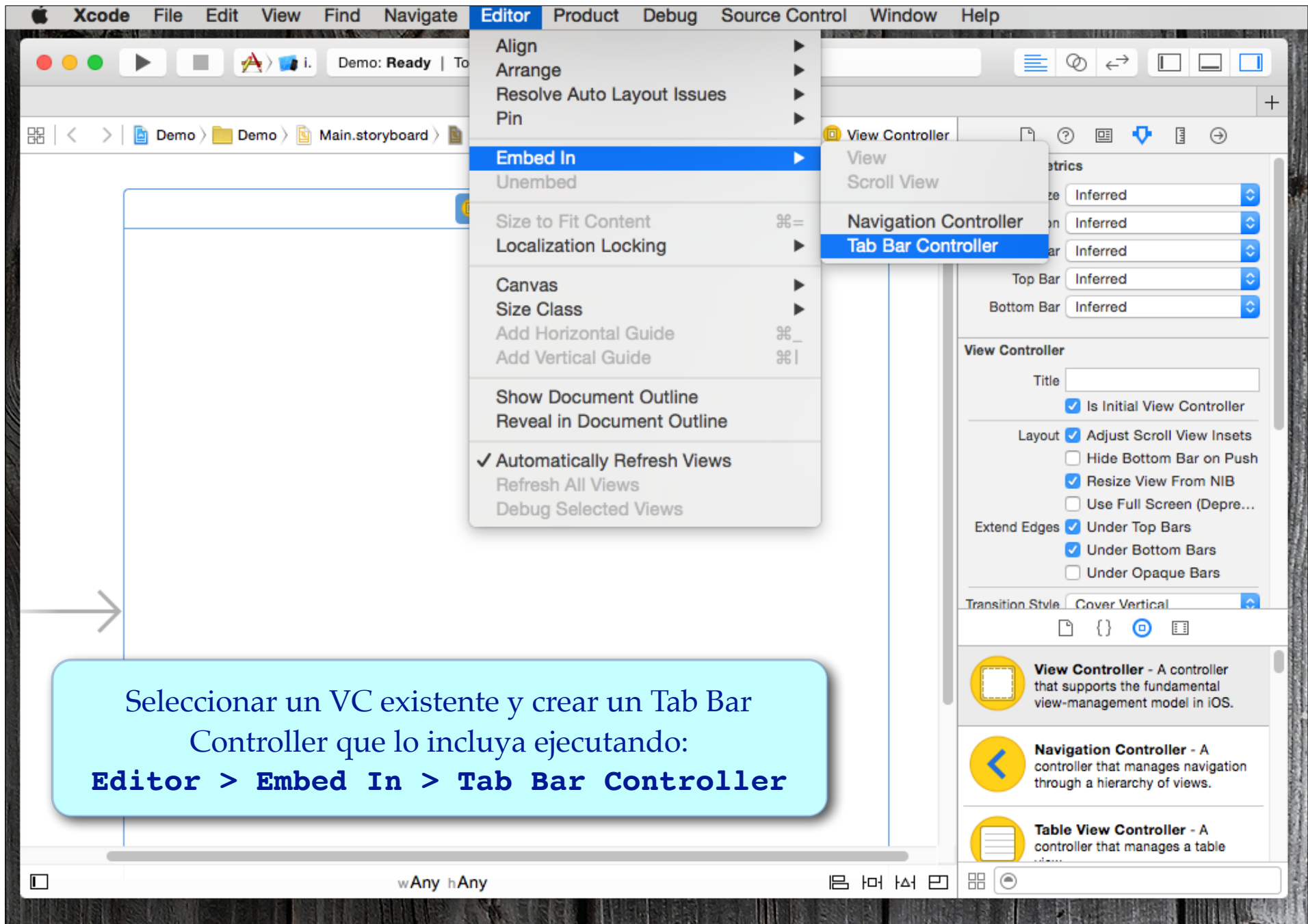
- Table View Controller** - A controller that manages a table view.
- Tab Bar Controller** - A controller that manages a set of view controllers that represent tab ba...
- Split View Controller** - A composite view controller that manages left and right view cont...

The image shows the Xcode interface for editing a storyboard. On the left, a project browser shows a 'Demo' project with files like AppDelegate.swift, FirstViewController.swift, and SecondViewController.swift. The main canvas displays a storyboard with a 'Tab Bar Controller' and two child view controllers. On the right, the 'Custom Class' panel shows the class set to 'UIViewController', which is circled in red. Below it, the 'Identity' panel shows 'Xcode Specific Label' and 'NBN-as-9yD'. At the bottom right, a library of view controller classes is visible, including 'Table View Controller', 'Tab Bar Controller', and 'Split View Controller'. A light blue callout box with a pointer to the 'Class' dropdown contains the following text:

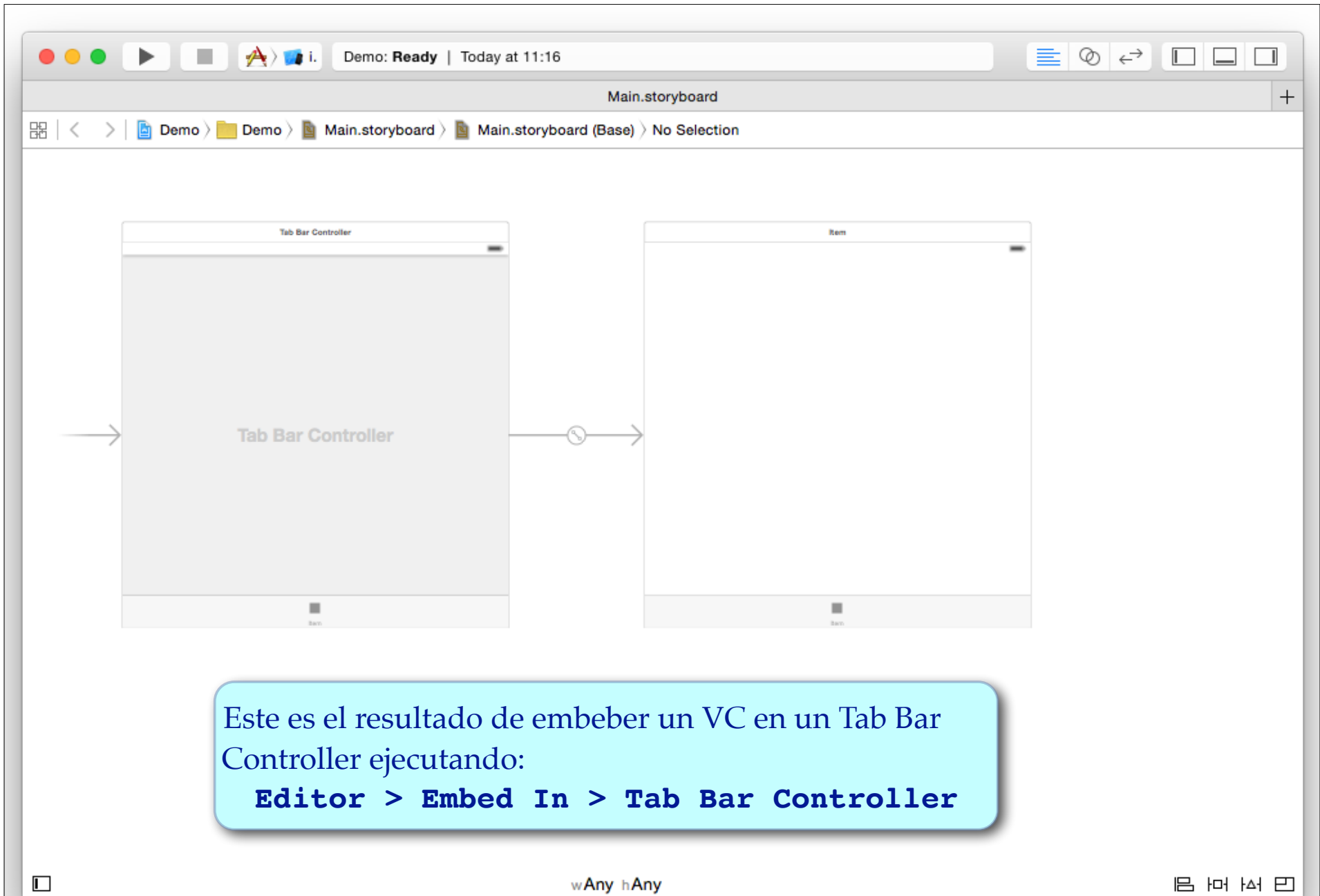
Los VC creados son instancias de la clase **UIViewController**. Hay que cambiar su clase a una clase derivada que habremos creado antes.

# Storyboard

- Interface Builder permite editar los ficheros storyboard para:
  - Incluir Tab Bar Controllers.
    - Arrastrando un objeto Tab Bar Controller desde la biblioteca de objetos.
      - Se crea un objeto Tab Bar Controller y dos objetos VC.
        - Si queremos seguir trabajando usando los VC creados, tendremos que usar el inspector para indicar cual es su clase real de estos VC.
        - También podemos borrar los dos VC creados y enganchar el Tab Bar Controller con otros VC ya existentes.
      - O seleccionando un VC existente en el storyboard y ejecutando:  
**Menú Editor > Embed In > Tab Bar Controller**
  - Incluir un VC ya existente a un Tab Bar Controller ya existente.
    - Creando un segue Relationship desde el Tab Bar Controller hasta el VC.
  - Editar el título, el icono, el badge, ... de los Tab Bar Items que se muestran para cada VC añadido a un Tab Bar Controller.







Este es el resultado de embeber un VC en un Tab Bar Controller ejecutando:

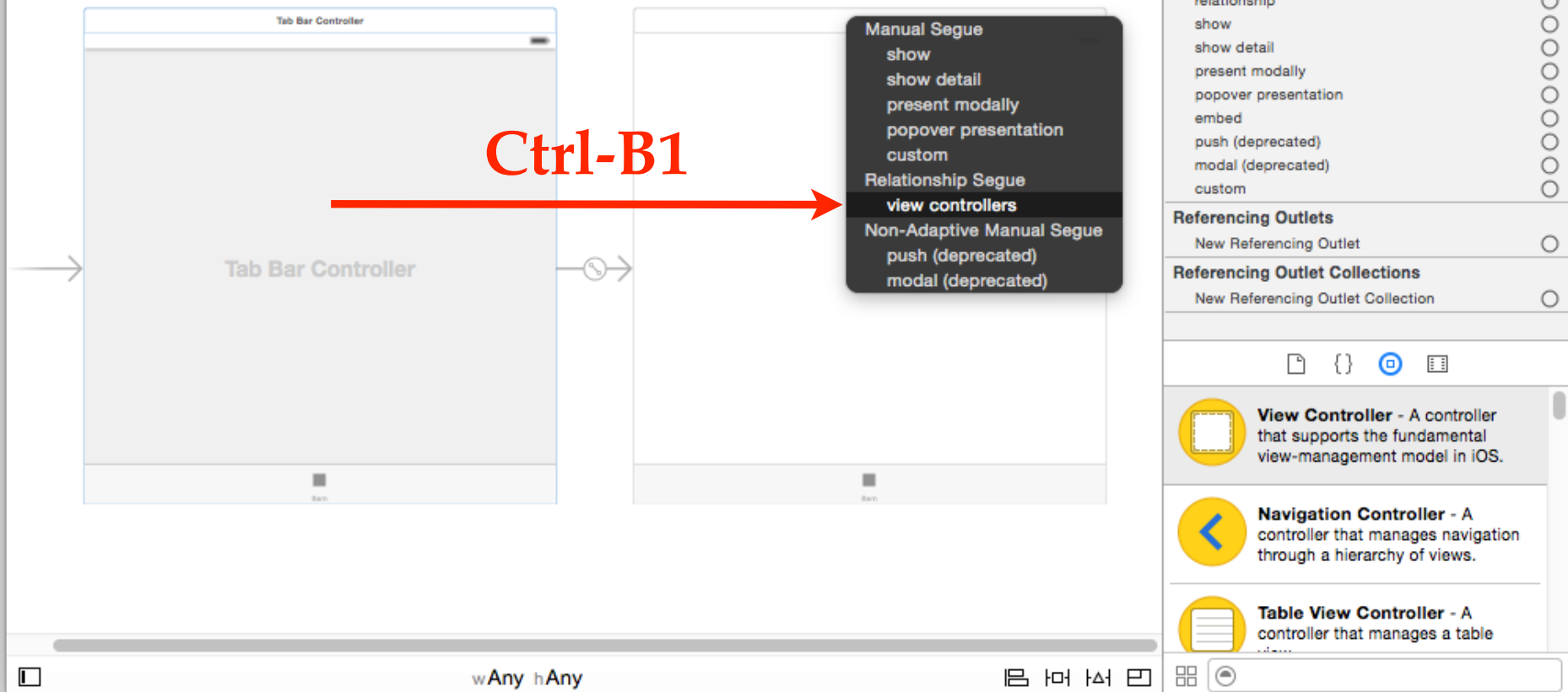
**Editor > Embed In > Tab Bar Controller**

# Storyboard

- Interface Builder permite editar los ficheros storyboard para:
  - Incluir Tab Bar Controllers.
    - Arrastrando un objeto Tab Bar Controller desde la biblioteca de objetos.
      - Se crea un objeto Tab Bar Controller y dos objetos VC.
        - Si queremos seguir trabajando usando los VC creados, tendremos que usar el inspector para indicar cual es su clase real de estos VC.
        - También podemos borrar los dos VC creados y enganchar el Tab Bar Controller con otros VC ya existentes.
      - O seleccionando un VC existente en el storyboard y ejecutando:
        - Menú Editor > Embed In > Tab Bar Controller
  - Incluir un VC ya existente a un Tab Bar Controller ya existente.
    - Creando un **segue Relationship** desde el Tab Bar Controller hasta el VC.
  - Editar el título, el icono, el badge, ... de los Tab Bar Items que se muestran para cada VC añadido a un Tab Bar Controller.

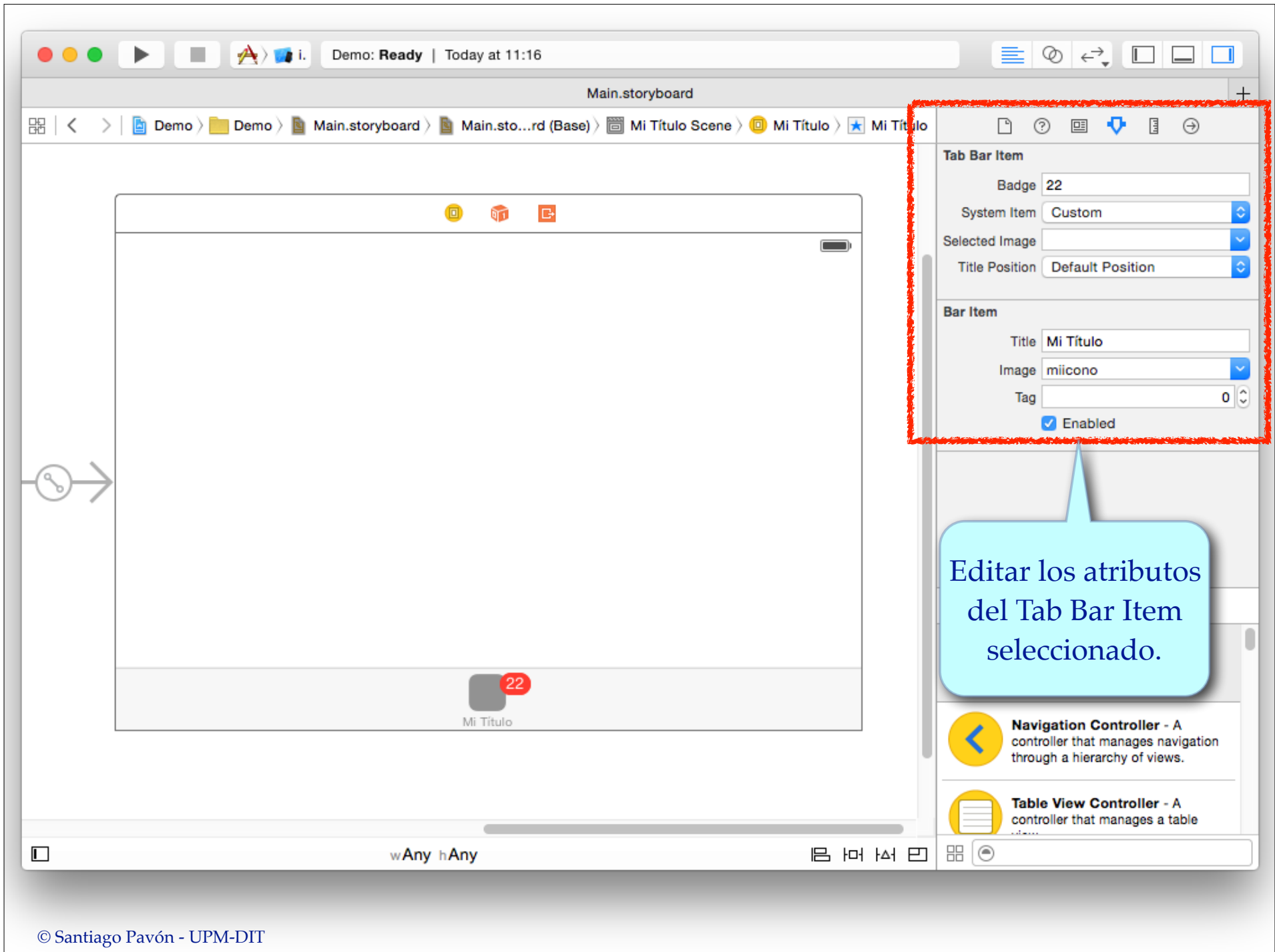
Un VC se añade a un Tab Bar Controller creando un segue.  
Seleccionar el tipo del segue como:

**Relationship Segue > view controllers**



# Storyboard

- **Interface Builder permite editar los ficheros storyboard para:**
  - Incluir Tab Bar Controllers.
    - Arrastrando un objeto Tab Bar Controller desde la biblioteca de objetos.
      - Se crea un objeto Tab Bar Controller y dos objetos VC.
        - Si queremos seguir trabajando usando los VC creados, tendremos que usar el inspector para indicar cual es su clase real de estos VC.
        - También podemos borrar los dos VC creados y enganchar el Tab Bar Controller con otros VC ya existentes.
      - O seleccionando un VC existente en el storyboard y ejecutando:
        - Menú Editor > Embed In > Tab Bar Controller
  - Incluir un VC ya existente a un Tab Bar Controller ya existente.
    - Creando un segue Relationship desde el Tab Bar Controller hasta el VC.
  - Editar el título, el icono, el badge, ... de los Tab Bar Items que se muestran para cada VC añadido a un Tab Bar Controller.



# UITabBarController

- Este controlador es un objeto de la clase **UITabBarController**.

- Algunos métodos y propiedades:

- Array con todos los view controllers que gestiona.

```
var viewControllers: [UIViewController]?
```

- Los VC de este array, y todos los VC descendientes de estos, tienen una propiedad, llamada **tabBarController**, que apunta al objeto Tab Bar Controller que los contiene.

- Esta propiedad es **nil** si el VC no está gestionado por un Tab Bar Controller.

- El objeto Tab Bar Controller también es el **parentViewController** de los VC que contiene.

- Programáticamente los VCs pueden cambiarse por otros con el método:

```
func setViewControllers(_ viewControllers: [UIViewController]?,  
                        animated animated: Bool)
```

- El VC seleccionado se indica con:

```
unowned(unsafe) var selectedViewController: UIViewController?
```

```
var selectedIndex: Int
```

- Un delegado **UITabBarControllerDelegate** con el que se puede prohibir la selección de un VC, avisar cuando se selecciona un VC, etc...

# UITabBar y UITabBarItem

- La barra de control es un objeto **UITabBar** que contiene elementos **UITabBarItem**.
- Por defecto, cada **UITabBarItem** muestra un icono y un título.
  - El título y el icono de cada item puede editarse en el storyboard.
  - o pueden configurarse programáticamente mediante la propiedad `tabBarItem` de los VC.
- Cada View Controller tiene una propiedad **tabBarItem** para controlar la apariencia del item que lo representa en la UITabBar.

```
var tabBarItem: UITabBarItem!
```

## - Propiedades de **UITabBarItem**:

- **title**: El texto mostrado en la Tab Bar para este item.
  - Por defecto, es el valor de la propiedad **title** del VC asociado.
- **image**: El icono mostrado en la Tab Bar cuando no esta seleccionado el VC asociado.
  - Es un icono PNG de 30x30, usado una máscara con el color **tintColor** de la Tab Bar.
  - Para usar la imagen original, sin tratarla como una máscara, usar una imagen cuyo modo de renderizarse sea **.AlwaysOriginal** (crear una subclase de **UIImage** y sobrescribir **imageWithRenderingMode:**, o usar el xcassets y seleccionar el valor **"Original Image"** en la propiedad **"Render As"** de la imagen).
- **selectedImage**: icono mostrado cuando el VC está seleccionado.
- **badgeValue**: el badge (numerito) mostrado en el item.
- **enabled**: habilitar o deshabilitar el item.

# Demo

- Crear un proyecto con la plantilla Tabbed Application.
- Crear nuevas pantallas.
  - De distinto tipo.



# Combinar Navigation y Tab Bar Controllers

# ¿Cómo se hace?

- Con storyboards:
  - Cree varios UINavigationController independientes.
    - Cada uno con su propia jerarquía de VC.
  - Añada los Navigation Controller dentro de un único Tab Bar Controller.
- Programáticamente:
  - Cree varios UINavigationController independientes.
    - Cada uno con su propia jerarquía de VC.
  - Méталos en un array.
  - Este array se usará para definir las pestañas de un único UITabBarController.



