



POLITÉCNICA

ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS

# Navigation Controller

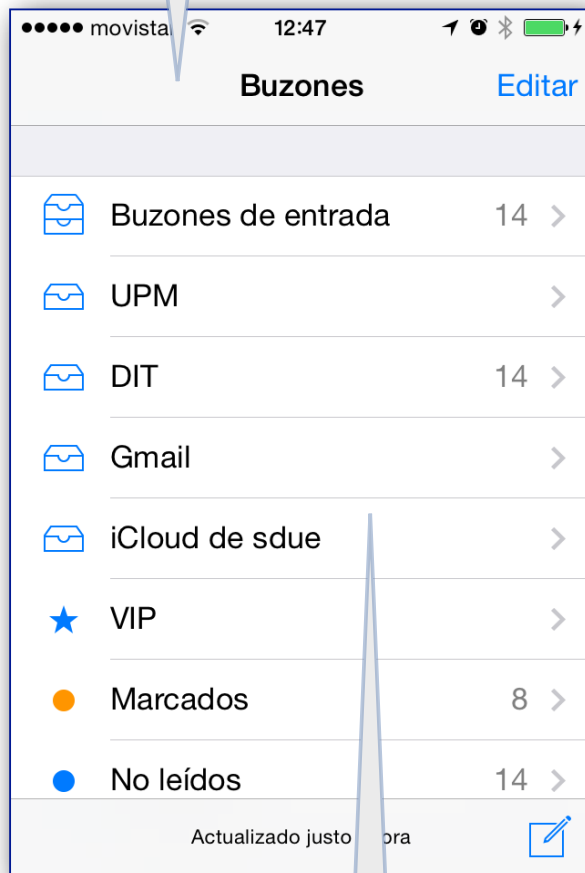
IWEB 2016-2017  
Santiago Pavón

ver: 2016.11.15

Barra de título creada por UINavigationController

Propiedad **title** del VC mostrado.

Botón de retroceso al VC anterior.  
Etiquetada con su title.



La vista (view) del VC actual.

Cada VC mostrado tiene su propia barra de herramientas con sus **toolbarItems**.

# Funcionamiento de UINavigationController

- La navegación con **UINavigationController** consiste en pasar por una serie de pantallas nuevas (View Controllers) con las que se crea una pila.
  - Para retroceder por la secuencia de pantallas por la que hemos pasado puede usarse el botón situado en la izquierda de la barra de navegación.
  - Opcionalmente, puede existir un botón en la derecha de la barra de navegación para pasar a la pantalla siguiente.
  - Y programáticamente se puede saltar a la pantalla que se desee.

- Un **UINavigationController** muestra:
  1. Una barra de navegación (**UINavigationController**) con:
    - un botón (**UIBarButtonItem**) para volver a la pantalla anterior.
      - Creado automáticamente con el título de la pantalla anterior, o añadido programáticamente.
    - el título de la pantalla actual.
      - asociado a la propiedad **title** del VC mostrado.
    - Otros botones (**UIBarButtonItem**) añadidos explícitamente en la derecha o la izquierda de la barra.
  2. La view del VC que esté actualmente en el **top** de la pila.
  3. Barra de herramientas (**UIToolBar**) que muestra el contenido de la propiedad **toolbarItems** del VC actual.
    - **toolbarItems** es un array de objetos **UIBarButtonItem**.

# UINavigationController

- Propiedades de UINavigationController :

- **topViewController** apunta al VC en el top de la pila.
- **visibleViewController** apunta al VC mostrado.
- **viewControllers** es el array con la pila de VCs.
- **navigationBar** es la barra de navegación mostrada.
- **toolBar** es la barra de herramientas mostrada.
- etc...

- Métodos de UINavigationController usados para navegar:

```
func popViewController(animated: Bool) -> UIViewController?
```

```
func popToRootViewController(animated: Bool)  
    -> [UIViewController]
```

```
func pushViewController(_ viewController: UIViewController,  
    animated: Bool)
```

etc...

# UINavigationController

- Los objetos **UINavigationController** gestionan una pila con los objetos UINavigationControllerItem por los que se va navegando.
  - Cada objeto UINavigationControllerItem apunta al VC que debe mostrar cuando es el primer elemento de la pila.
- Es usado por los navigation controllers para implementar su interface de navegación.
  - pero las barras de navegación también pueden usarse de forma independiente.

# UINavigationController

- Los objetos UINavigationController contienen los botones y views que hay que mostrar en una UINavigationControllerBar.
- Cada objeto View Controller usado en este tipo de navegación tiene asociado un objeto UINavigationControllerItem donde configura los elementos a mostrar en la barra de navegación cuando es visible.
  - **title** es el título mostrado en el centro de la barra de navegación.
  - **backBarButtonItem** es el botón para retroceder a la pantalla anterior.
  - **prompt** es una línea de texto situada en la parte superior de la barra de navegación
  - **titleView**, **leftBarButtonItem**, **leftBarItem**, **rightBarButtonItem**, **rightBarItem** son views y botones personalizados.
  - ...

# UIToolbar

- **UIToolbar** es una barra de herramientas.
  - Por herramientas queremos decir botones **UIBarButtonItem**.
- Los **UINavigationController** muestran en una barra de herramientas los elementos contenidos en la propiedad **toolbarItems** del VC actual.
  - Cada VC tiene su propia propiedad **toolbarItems**.
  - **toolbarItems** es un array de objetos **UIBarButtonItem**.
    - Los **UIBarButtonItem** pueden ser:
      - botones que usan target/ action o segues:
        - estilos: con borde, planos, con texto, con imágenes predefinidas, texto personalizado o predefinido, etc.
      - botones usados para crear separaciones fijas o flexibles.
  - La visibilidad de esta barra se controla con la propiedad **isToolbarHidden**.
- También podemos añadir nuestras propias **UIToolbars** a nuestras GUIs.
  - Típicamente en la parte superior o inferior de la pantalla.



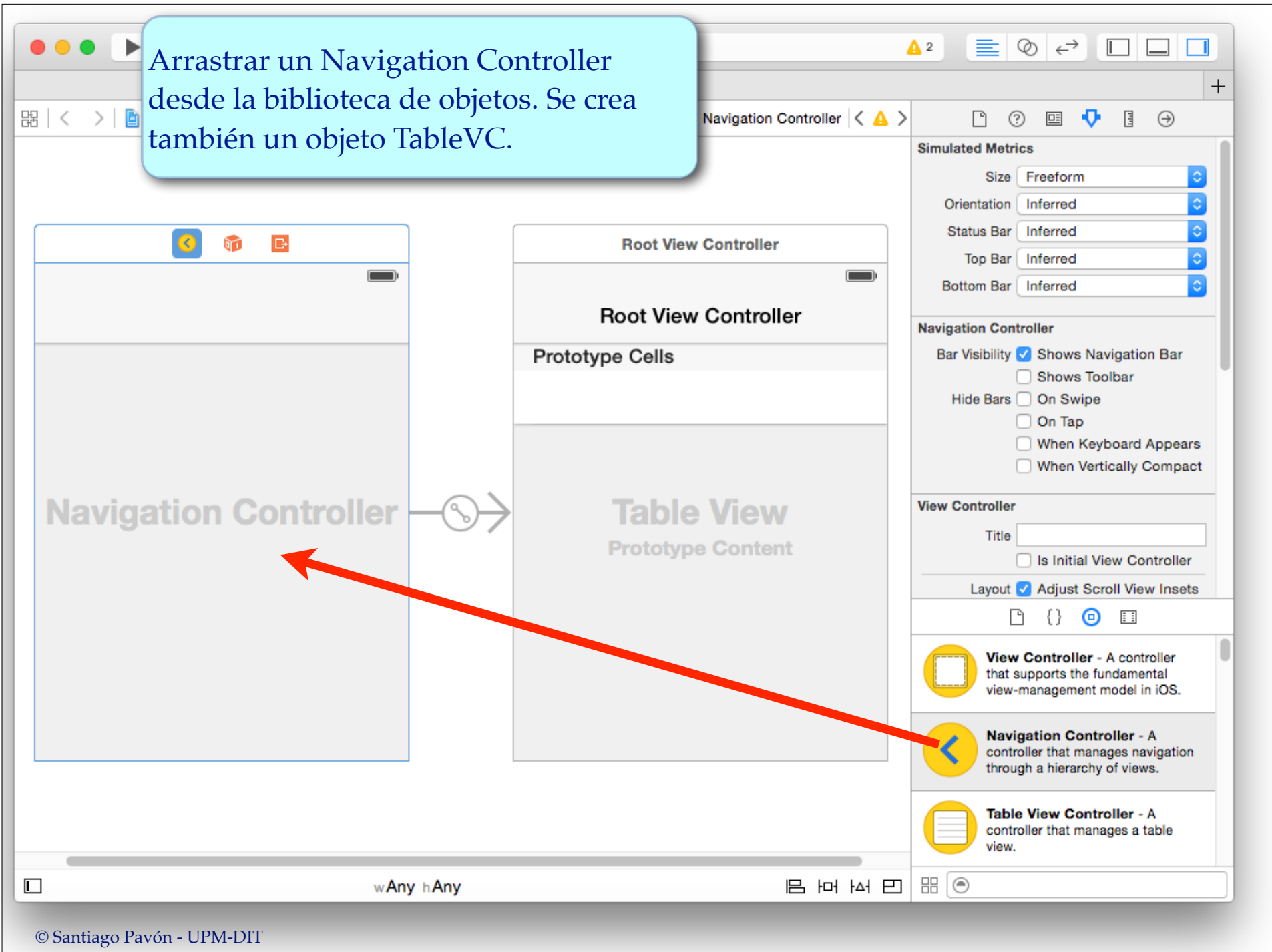
# UIViewController

- Los objetos **ViewControllers** tienen:
  - Propiedad **navigationController** que apunta al **UINavigationController** que los controla.
    - Y la propiedad **parent** también apunta al Navigation Controller.
  - Propiedad **title** usada en la barra de navegación para poner el título.
  - Propiedad **navigationItem** gestiona los botones y vistas que se muestran en la barra de navegación.
  - Propiedad **toolbarItems** es un array con los items a poner en la toolbar.
  - ...

# Demo: Añadir NavController

- Usando Interface Builder para editar un Storyboard:
  - Arrastrar un objeto Navigation Controller desde la biblioteca de objetos.
  - Se crea un objeto Navigation Controller y un objeto TableVC.
    - Si queremos seguir trabajando usando el TableVC creado, tendremos que usar el inspector de identidad para indicar cuál es la clase real del TableVC.
    - También podemos borrar el TableVC creado y enganchar el Navigation Controller con algún VC ya existente.

Arrastrar un Navigation Controller desde la biblioteca de objetos. Se crea también un objeto TableVC.



The screenshot shows the Xcode storyboard editor with a storyboard named "Main.storyboard". The storyboard contains a "Navigation Controller" and a "Table View Controller". A callout box points to the "Table View Controller" and contains the following text:

El TableVC creado es una instancia de la clase UITableViewController.  
Hay que cambiar su clase a una clase nuestra (será una clase derivada de UITableViewController) que habremos creado antes.

The right-hand side of the screenshot shows the "Custom Class" inspector, where the "Class" is set to "UITableViewController" and the "Module" is set to "None". Below the inspector, there is a list of components with their descriptions:

- View Controller** - A controller that supports the fundamental view-management model in iOS.
- Navigation Controller** - A controller that manages navigation through a hierarchy of views.
- Table View Controller** - A controller that manages a table view.

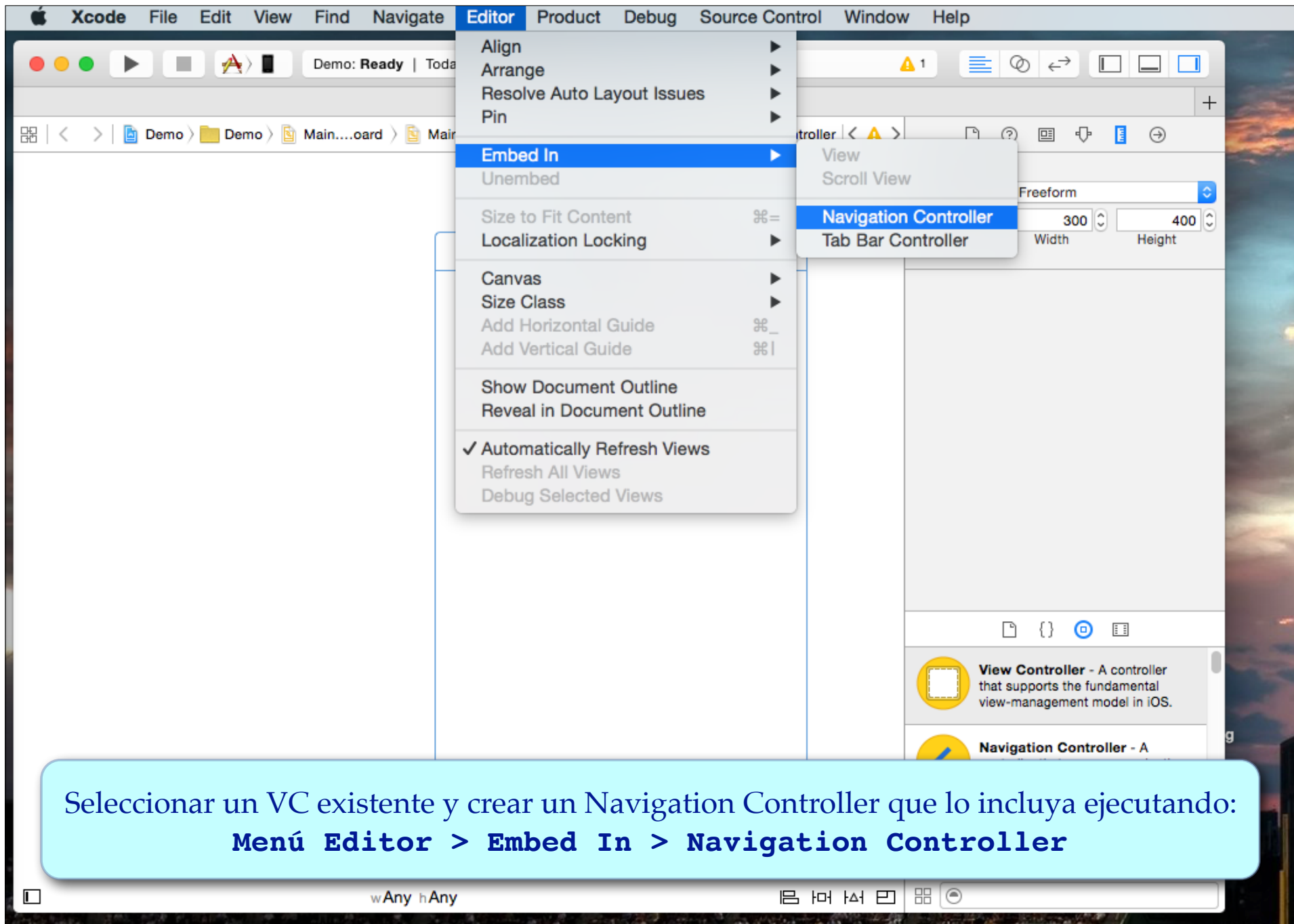
# Demo: Embeber en un NavController

- Usando Interface Builder para editar un Storyboard:
  - Seleccionar un VC ya existente en el storyboard y ejecutar:

Menú Editor >

Embed In >

Navigation Controller



Seleccionar un VC existente y crear un Navigation Controller que lo incluya ejecutando:  
**Menú Editor > Embed In > Navigation Controller**

The screenshot shows the Xcode storyboard editor interface. The main canvas displays a storyboard named 'Main.storyboard' with a 'Navigation Controller Scene' selected. Inside this scene, a 'Navigation Controller' widget is shown with a 'View Controller' widget embedded within it. An arrow points from the 'View Controller' widget to the right-hand 'Properties' panel. The 'Properties' panel is divided into several sections: 'Simulated Metrics' (Size: Freeform, Orientation: Inferred, Status Bar: Inferred, Top Bar: Inferred, Bottom Bar: Inferred), 'Navigation Controller' (Bar Visibility:  Shows Navigation Bar, Shows Toolbar: , Hide Bars:  On Swipe,  On Tap,  When Keyboard Appears,  When Vertically Compact), and 'View Controller' (Title: , Is Initial View Controller: , Layout:  Adjust Scroll View Insets). Below the properties panel, there are two informational cards: one for 'View Controller' and one for 'Navigation Controller'. A light blue callout box at the bottom of the screen contains the text: 'Este es el resultado de embeber un VC en un Navigation Controller ejecutando: **Menú Editor > Embed In > Navigation Controller**'.

# Demo: Crear Segues

- Editar un fichero storyboard para:
  - Enganchar un VC ya existente a un Navigation Controller ya existente.
    - Hay que crear un segue **Relationship** de tipo **root view controller** desde el Navigation Controller hasta el VC.
  - Enganchar entre sí los VC existentes para formar cadenas de navegación.
    - Hay que crear un **segue** de tipo **Show** para cada transición.
      - **Origen:** botón, celda de tabla, accesorio de la celda de una tabla, vc, reconocedor de gestos, ...
      - **Destino:** el VC destino.
      - **Identificador:** valor único que identifique el segue.

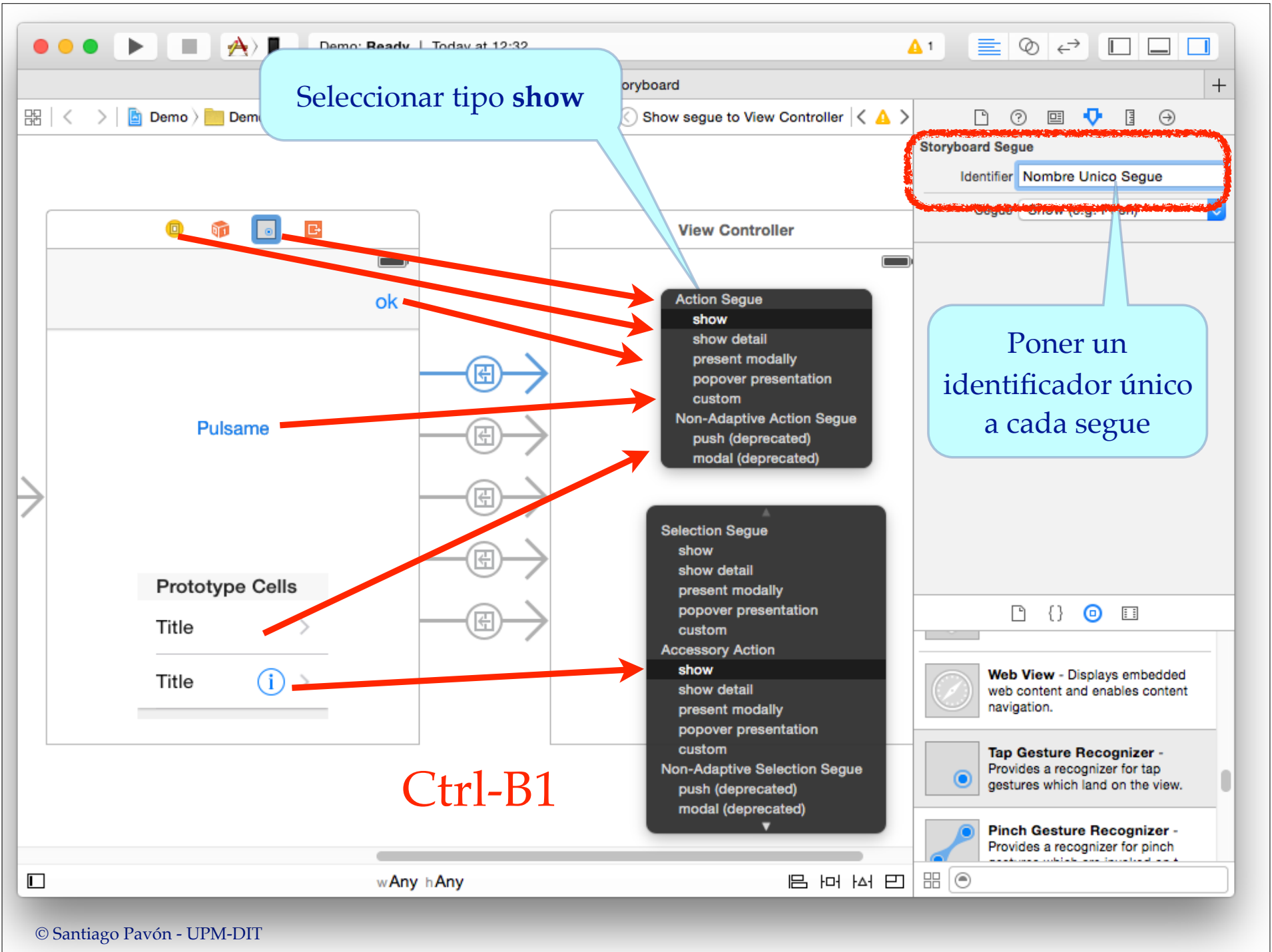


The screenshot shows the Xcode storyboard editor for 'Main.storyboard'. The breadcrumb trail indicates the current location: Demo > Demo > Main.storyboard > Main.storyboard (Base) > Navigation Controller Scene > Navigation Controller. On the left is a 'Navigation Controller' widget, and on the right is a 'View Controller' widget. A red arrow labeled 'Ctrl-B1' points from the Navigation Controller to the View Controller. A segue menu is open over the View Controller, listing options: Manual Segue (show, show detail, present modally, popover presentation, custom), Relationship Segue (root view controller), and Non-Adaptive Manual Segue (push (deprecated), modal (deprecated)). The 'root view controller' option is highlighted.

El VC raíz del Navigation Controller se engancha creando un segue.  
Seleccionar el tipo del segue como:  
**Relationship Segue > root view controller**

The image shows a screenshot of the Xcode IDE. The main storyboard area displays a 'Navigation Controller' with a red curved arrow pointing from a 'View Controller' to it. A light blue callout bubble points to the 'Navigation Controller' with the text 'Cambiar la escena inicial al Navigation Controller'. Another light blue callout bubble points to the 'View Controller' with the text 'Se pinta una barra de navegación automáticamente.' The right-hand 'Property Inspector' is open, showing the 'View Controller' section. The checkbox 'Is Initial View Controller' is highlighted with a red dashed border. Below the property inspector, there is a list of controller types: 'View Controller', 'Navigation Controller', and 'Table View Controller', each with a brief description.

© Santiago Pavón - UPM-DIT

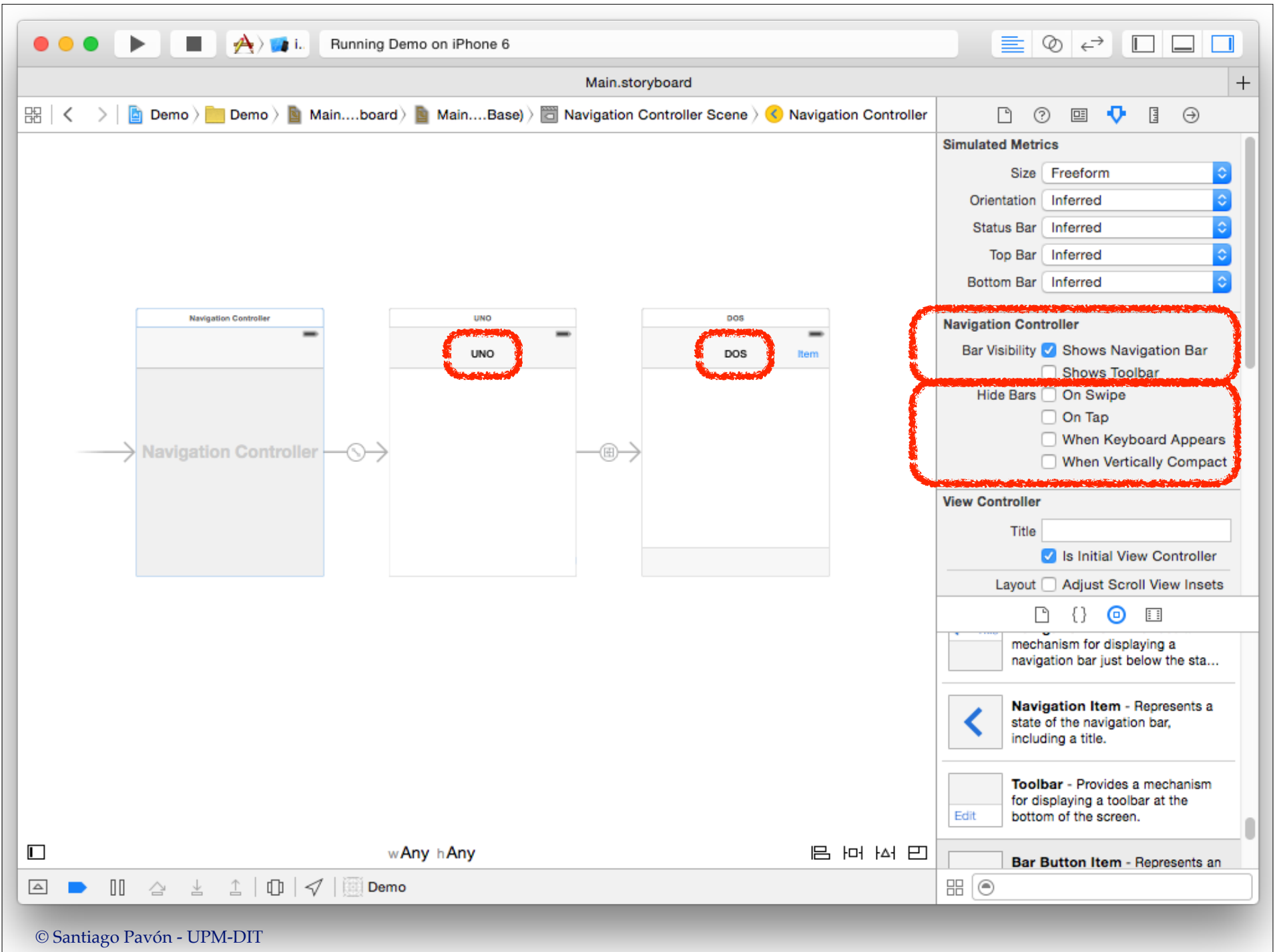


# Demo: IB - Configurar Un NavController

- Desde IB, editar un storyboard para configurar los Navigation Controller y los VC que presenta:
  - Añadir objetos Navigation Item a los VC que lo necesiten.
  - Poner un título en la barra de navegación de cada VC.
  - Estilo de la Navigation Bar.
  - Visibilidad de la Navigation Bar y de la Tool Bar.
  - Ocultar las barras con un tap, swipe, al mostrar el teclado, o al pasar a un Size Class vertical compacto.
  - ...

Al añadir un segue Show a un View Controller:

- Se muestra y se reserva el espacio que ocupará la barra de navegación.
- Para poner el título o añadir botones en la barra de navegación interactivamente usando IB, antes hay que arrastrar un objeto Navigation Item desde la biblioteca de objetos hasta el hueco que ocupa la barra de navegación.



# Ejemplo: Añadir un Botón

- Los botones (UIBarButtonItem) se pueden añadir a la barra de navegación usando IB o programáticamente.
  - **Con Interface Builder:**
    - Añadir un botón a la barra de navegación:
      - Editar storyboard con Interface Builder.
      - Arrastrar un UIBarButtonItem desde la biblioteca de objetos a la barra de navegación.
      - Configurar propiedades del botón.
      - Crear IBActions o segues

- **Programáticamente:**

```
let b = UIBarButtonItem(title: "acercar",  
                        style: .done,  
                        target: self,  
                        action: #selector(acercar))  
vc.navigationItem.leftBarButtonItem = b
```

- Existen más métodos inicializadores que permiten especificar:
  - imágenes, símbolos del sistema, views personalizadas.

# Flujo de Datos

- Normalmente, los view controllers por los que se navega necesitan pasarse datos entre ellos.
  - Ejemplo: Desde el view controller actual se pasan valores al siguiente view controller a mostrar asignándolos en propiedades del segundo View Controller.
- Buen diseño: poca cohesión entre los VC:
  - Evitar efectos laterales, facilitar la reutilización, VC independientes, ...
    - Evitar usar variables globales, objetos singletons, el delegado de la aplicación, etc. para pasar datos entre los VC.
- Para pasar datos al siguiente view controller:
  - ej: usar parámetros de entrada, protocolo data source.
- Para devolver datos al view controller anterior.
  - ej: usar delegación, unwind segues, closures.



