



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS

Navigation Controller

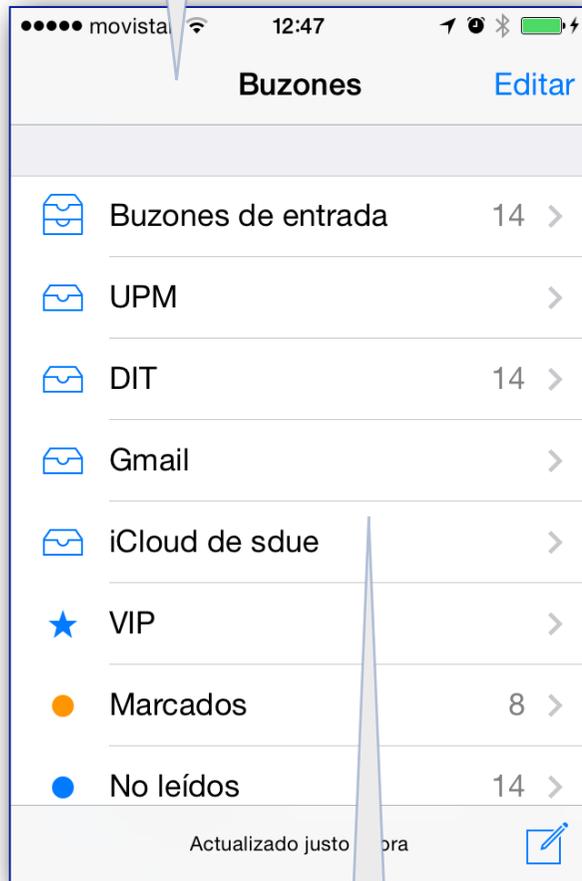
IWEB,LSWC 2013-2014
Santiago Pavón

ver: 2014.03.15

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.

Barra de herramientas con los **toolbarItems** del VC mostrado.

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.

Barra de herramientas con los toolbarItems del VC mostrado.

Capturas con iOS 6

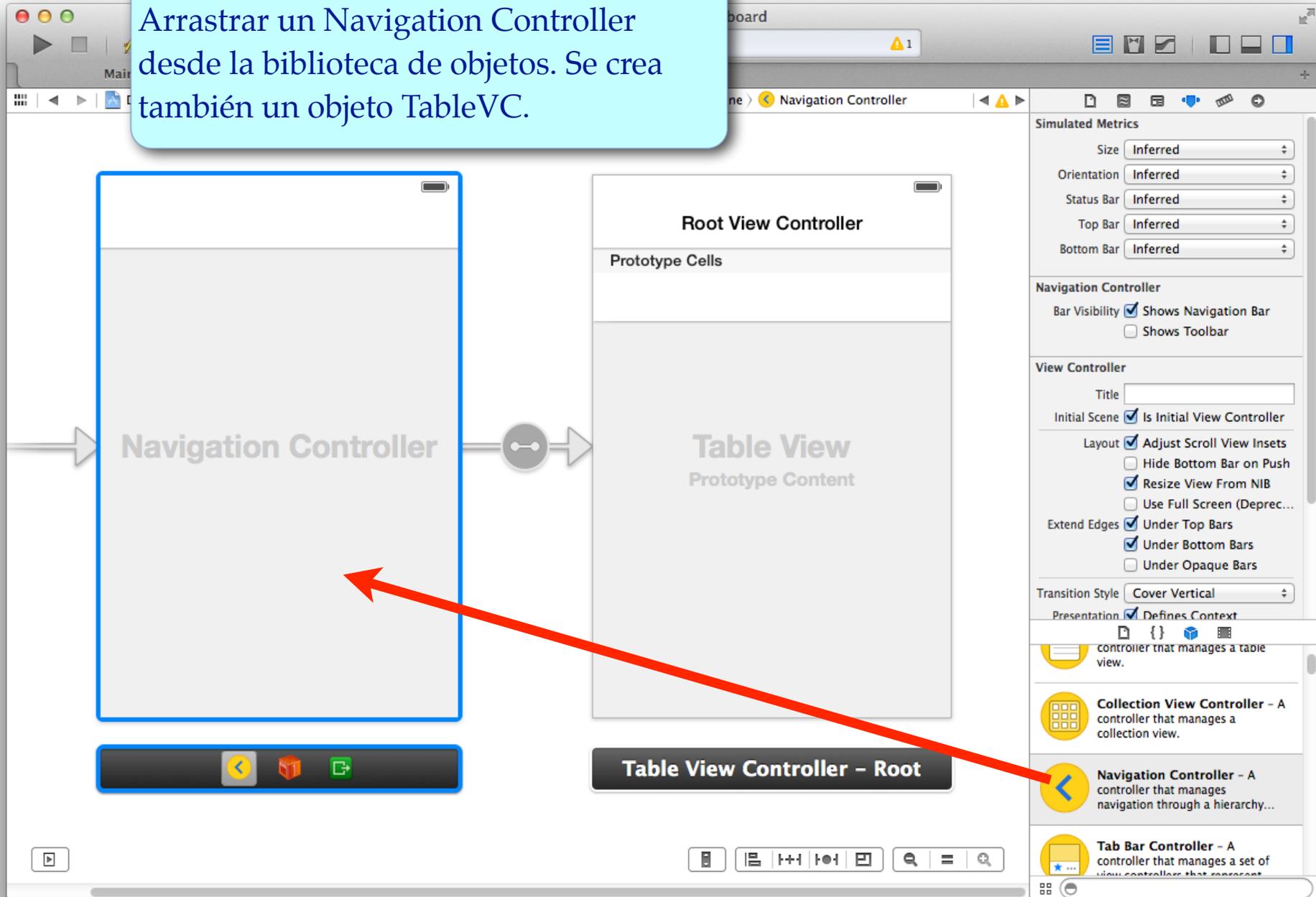
Funcionamiento de UINavigationController

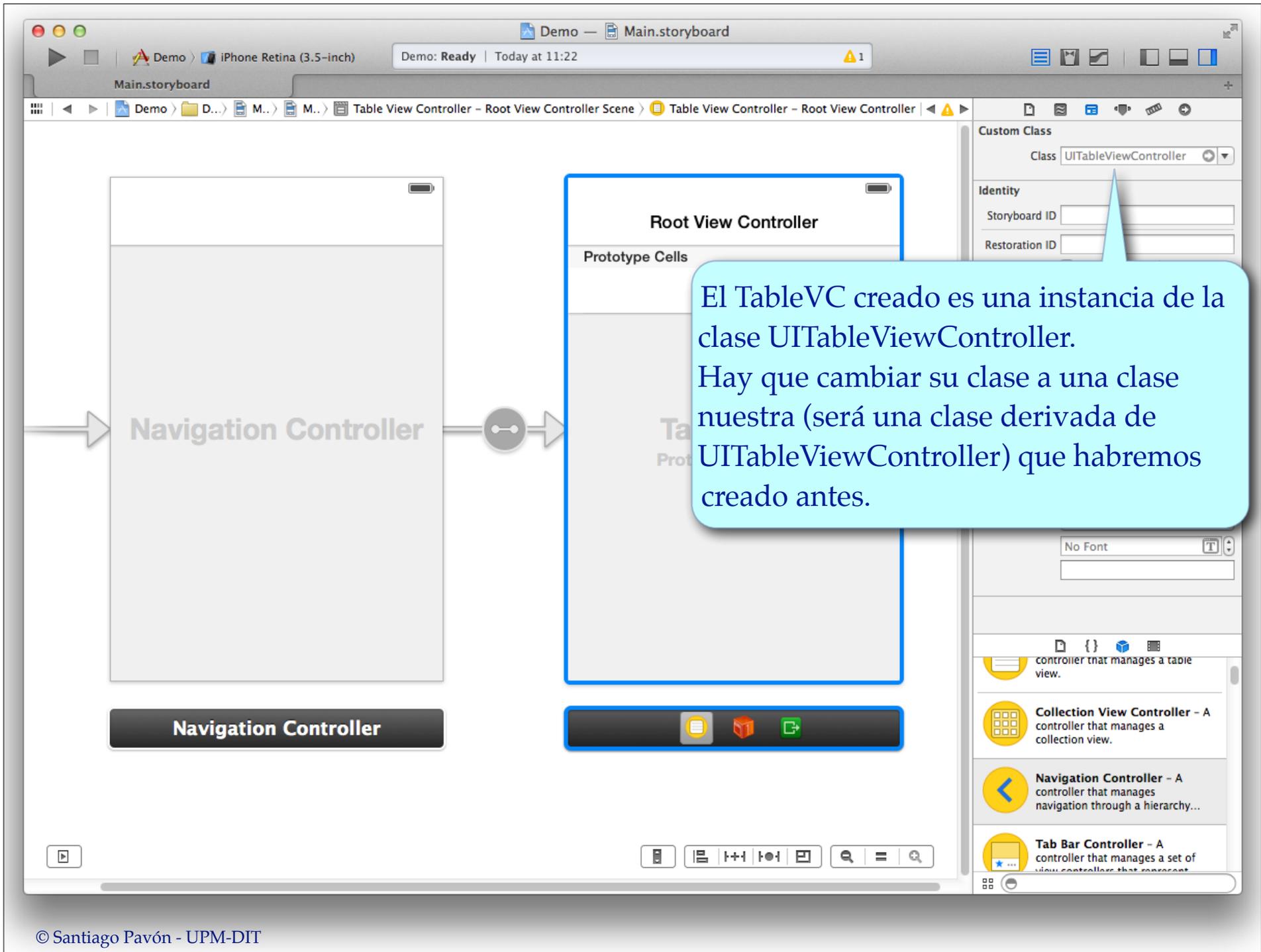
- La navegación con **UINavigationController** consiste en pasar por una serie de pantallas nuevas (ViewControllers) con las que se crea una pila.
 - Esta pila nos permite retroceder en la secuencia de pantallas por la que hemos pasado:
 - usando un botón de retroceder.
 - o programáticamente retroceder a la pantalla que se desee.
- Un **UINavigationController** muestra:
 - Una barra de navegación (**UINavigationController**) con:
 - el título de la pantalla actual.
 - asociado a la propiedad **title** del VC mostrado.
 - un botón (**UIBarButtonItem**) para volver a la pantalla anterior.
 - etiquetado con el título de la pantalla anterior.
 - La view del VC que esté actualmente en el **top** de la pila.
 - Barra de herramientas (**UIToolBar**) que muestra el contenido de la propiedad **toolbarItems** del VC actual.
 - **toolbarItems** es un NSArray de objetos **UIBarButtonItem**.

Storyboard

- Editar los ficheros storyboard para incluir Navigation Controllers:
 - 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.
 - Seleccionar un VC existente en el storyboard y ejecutar:
Menú Editor > Embed In > Navigation Controller

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



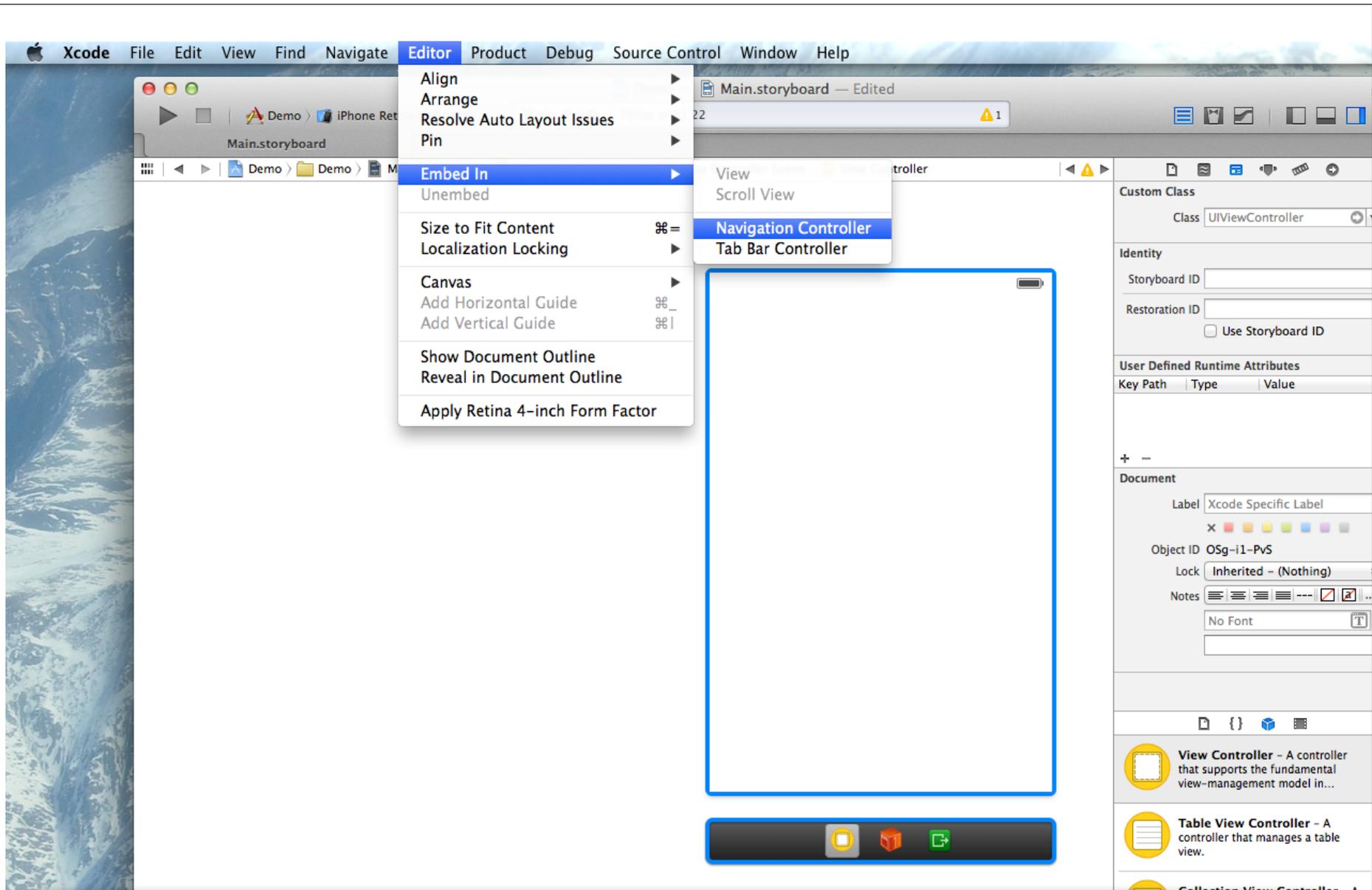


Storyboard

- Editar los ficheros storyboard para incluir Navigation Controllers:

- 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 cual es la clase real del TableVC.
 - También podemos borrar el TableVC creado y enganchar el Navigation Controller con algún VC ya existente.
- Seleccionar un VC 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

Este es el resultado de embeber un VC en un Navigation Controller ejecutando:
Menú Editor > Embed In > Navigation Controller

Storyboard

- Editar los ficheros storyboard para:
 - Enganchar un VC ya existente a un Navigation Controller ya existente.
 - Creando un segue de tipo **Relationship root view controller** desde el Navigation Controller hasta el VC.
 - Enganchar entre sí los VC existentes para formar las cadenas de navegación.
 - Creando un segue de tipo Push 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 interface for a storyboard named 'Main.storyboard'. The storyboard contains a 'Navigation Controller' and a 'Demo View Controller'. A red arrow labeled 'Ctrl-B1' points to the segue menu, which is open and shows 'Relationship Segue' selected. The right sidebar shows the Attributes Inspector for the selected segue, with options like 'Bar Visibility', 'View Controller', and 'Transition Style'.

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

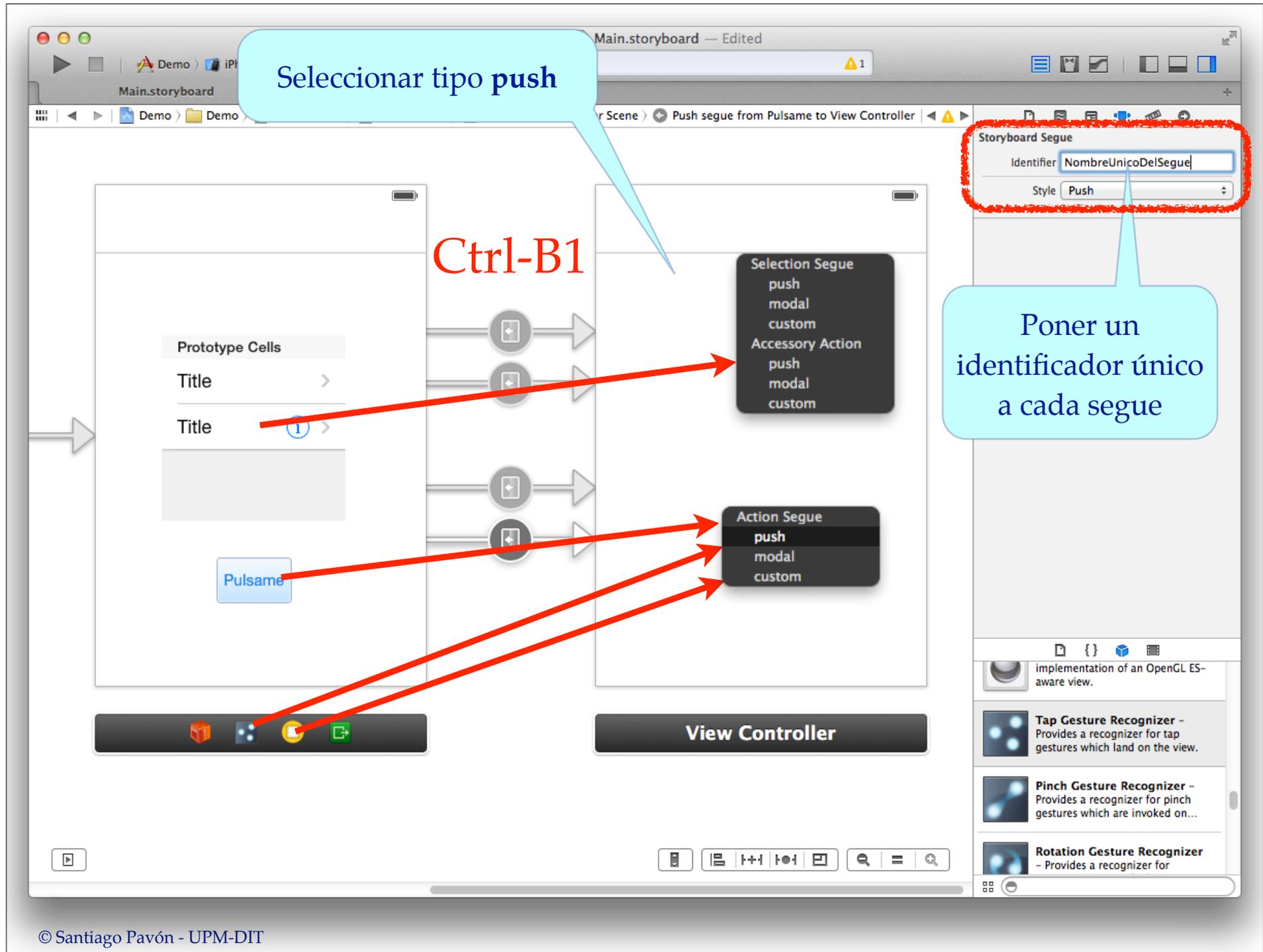
Se pinta una barra de navegación automáticamente.

Cambiar la escena inicial al Navigation Controller

© Santiago Pavón - UPM-DIT

Storyboard

- Editar los ficheros storyboard para:
 - Enganchar un VC ya existente a un Navigation Controller ya existente.
 - Creando un segue de tipo **Relationship root view controller** desde el Navigation Controller hasta el VC.
 - Enganchar entre sí los VC existentes para formar las cadenas de navegación.
 - Creando un segue de tipo **Push** 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.



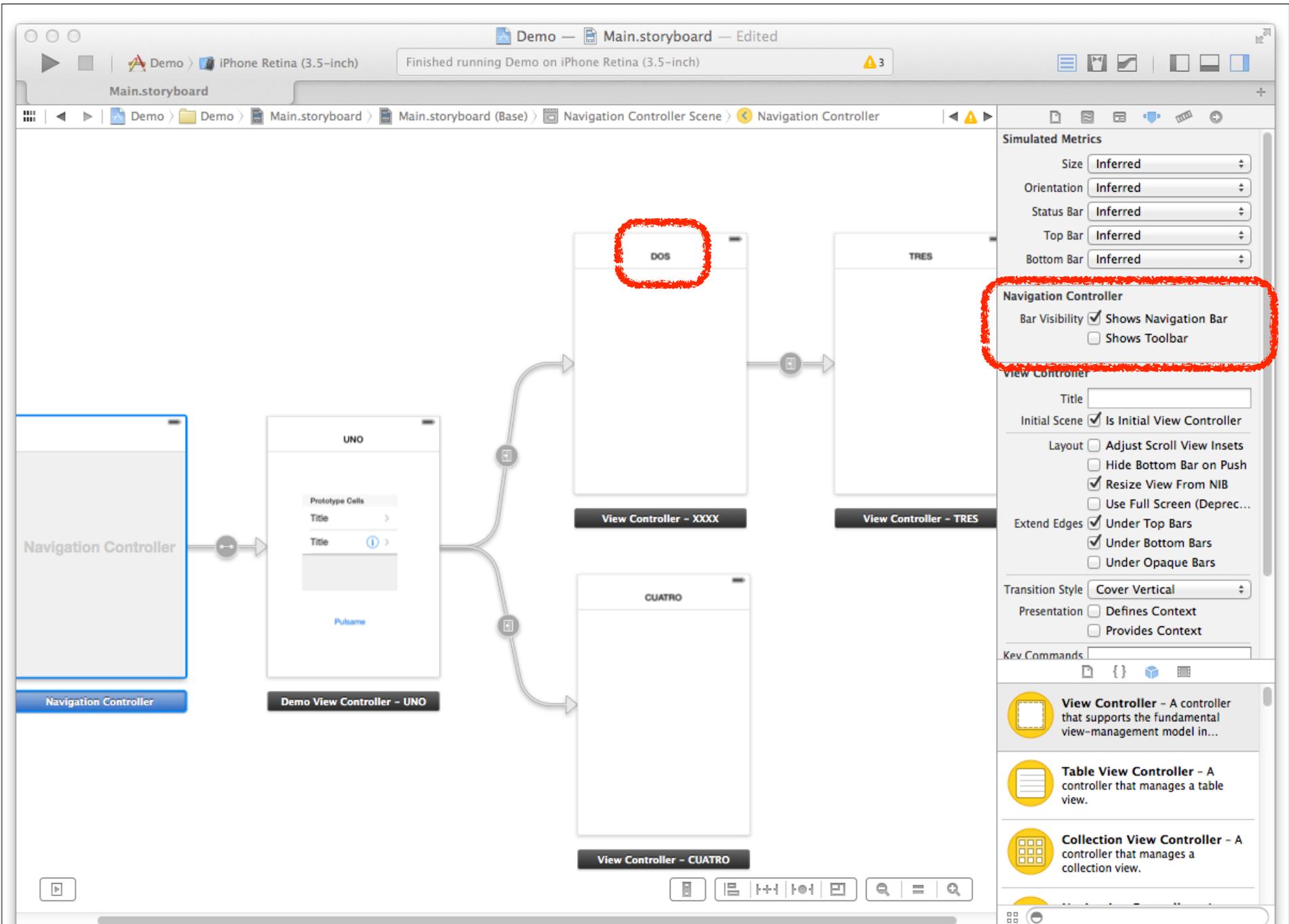
Seleccionar tipo push

Ctrl-B1

Poner un identificador único a cada segue

Storyboard

- Editar los ficheros storyboard para:
 - Poner un título en la barra de navegación de cada VC.
 - Otros ajustes:
 - Estilo de la Navigation Bar.
 - Visibilidad de la Navigation Bar y de la Tool Bar.
 - ...



Propiedades y Métodos

- Propiedades de **UINavigationController** :

- **topViewController** apunta al VC en el top de la pila.
- **visibleViewController** apunta al VC mostrado.
- **viewControllers** es el NSArray 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:

`-(UIViewController*)popViewControllerAnimated:(BOOL)animated`

`-(void)pushViewController:(UIViewController*)viewController
animated:(BOOL)animated`

etc...

Propiedades y Métodos

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

Personalizar la Barra de Navegación

- Los view controller poseen una propiedad que controla como debe pintarse la barra de navegación.
 - título
 - botón derecho, botón izquierdo, botón back
 - prompt
 - views personalizadas
 - ...

```
@property(nonatomic, readonly, retain)  
    UINavigationController *navigationItem;
```

Ejemplo: Poner un Botón

```
UIBarButtonItem *b =  
    [[UIBarButtonItem alloc]  
     initWithTitle:@"acercar"  
     style:UIBarButtonItemStyleBordered  
     target:self  
     action:@selector(acercar:)];  
  
vc.navigationItem.rightBarButtonItem = b;
```

- Otros métodos `initWith*` permiten especificar:
 - imágenes, símbolos del sistema, views personalizadas.

UIToolbar

- **UIToolbar** es una barra de herramientas creada por un **UINavigationController** para mostrar los elementos contenidos en la propiedad **toolbarItems** del VC actual.
 - Cada VC tiene su propia propiedad **toolbarItems**.
 - **toolbarItems** es un NSArray 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 **toolbarHidden**.
- También podemos añadir nuestras propias **UIToolbars** a nuestras GUIs.
 - Típicamente en la parte superior o inferior de la pantalla.

Flujo de Datos

- Normalmente, los view controllers por los que navego necesitan pasarse datos entre ellos.
 - Desde el view controller actual quiero parametrizar el siguiente view controller a mostrar.
- 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.

