



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS Segues

IWEB 2019-2020
Santiago Pavón

ver: 2019.11.03

Tipos de Segues

- Un storyboard es un fichero donde se diseñan gráficamente las escenas (pantallas) de la aplicación.
 - Para pasar de una escena a otra se usan seguees.
 - Para indicar que escenas se muestran dentro de otros controladores se usan seguees.
- **Seguees de transición:**
 - Adaptativos: *(Los seguees no adaptativos están deprecados desde iOS 8)*
 - **Show**
 - Presentar un VC como el nuevo VC master en un Split View Controller, o como el siguiente VC en un Navigation Controller.
 - **Show Detail**
 - Presentar un VC como el nuevo VC detail en un Split View Controller, como el siguiente VC en un Navigation Controller.
 - **Present Modally**
 - Presentación de un VC de forma modal.
 - **Present as Popover**
 - Presentar un VC como un popover, o de forma modal.
 - **Custom**
 - Presentar un VC con transiciones personalizadas.
 - Para dar marcha atrás en las transiciones realizadas con seguees:
 - **Unwind**

- *Segues de relación:*
 - Establecer los VC que se mostrarán dentro de otros controladores:
 - **Root View Controller**
 - Establecer cuál es el VC raíz que mostrará un Navigation Controller.
 - **View Controllers**
 - Establecer los VC mostrados por un Tab Bar Controller.
 - **Master View Controller y Detail View Controller**
 - Establecer los VC master y detail de un Split View Controller.
 - **Embed**
 - Establecer el VC que mostrará en un Container View.
 - Un Container View es un UIView que muestra la vista de otro VC.

Crear un Segue con IB

- Editando un Storyboard en el Interface Builder:
 - Los segues se crean **ctrl-arrastrando**
 - **desde** controles, reconocedores de gestos, celdas de una tabla, desde el icono que representa al VC de una escena, controladores, views contenedoras, etc...
 - **hasta** el VC destino,
 - y seleccionando el **tipo** de segue.
 - Usar el inspector de atributos para asignar un identificador único a los segues creados.

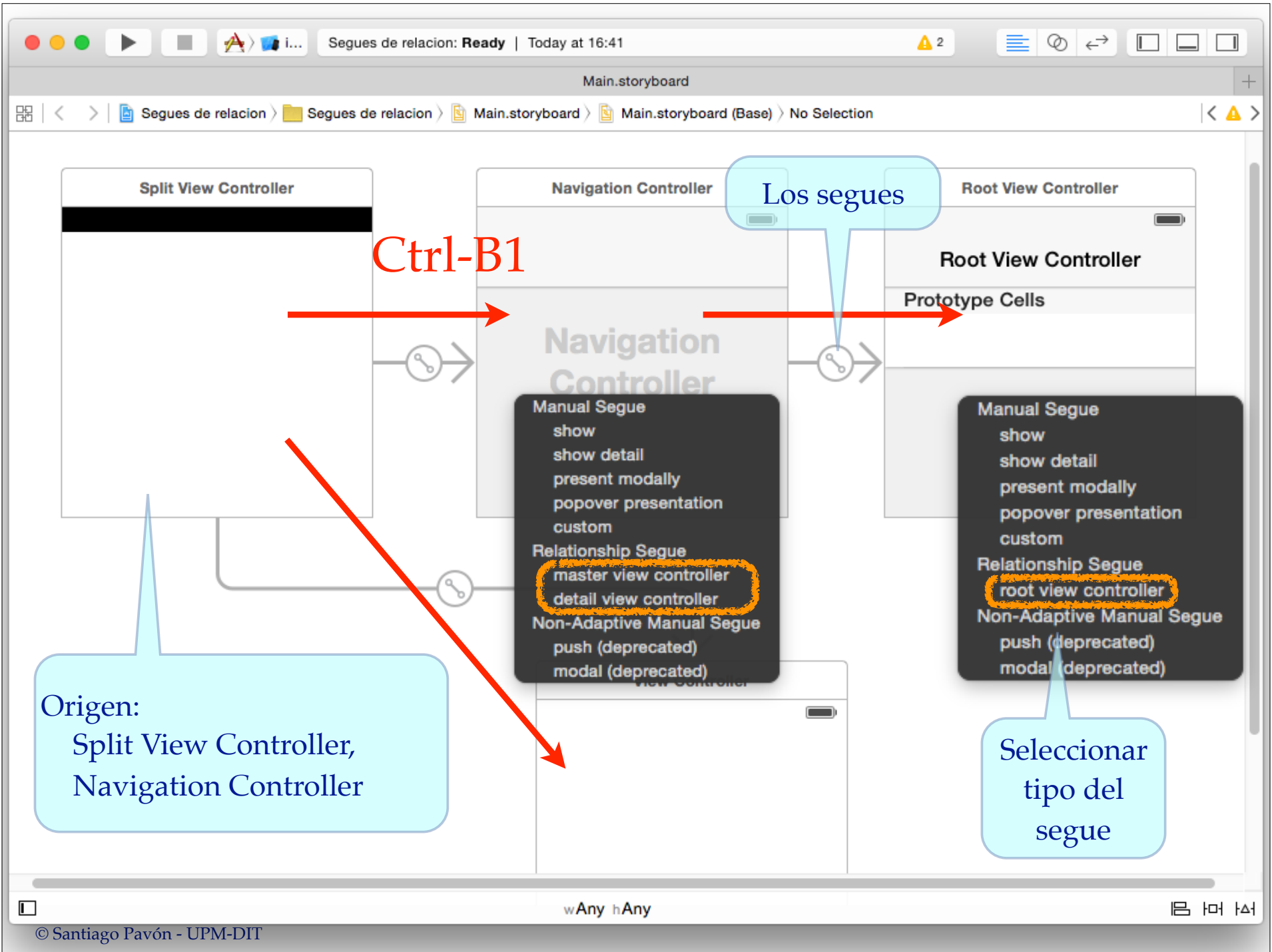
The image shows the Xcode Storyboard editor interface with several annotations in Spanish explaining how to create and configure segues:

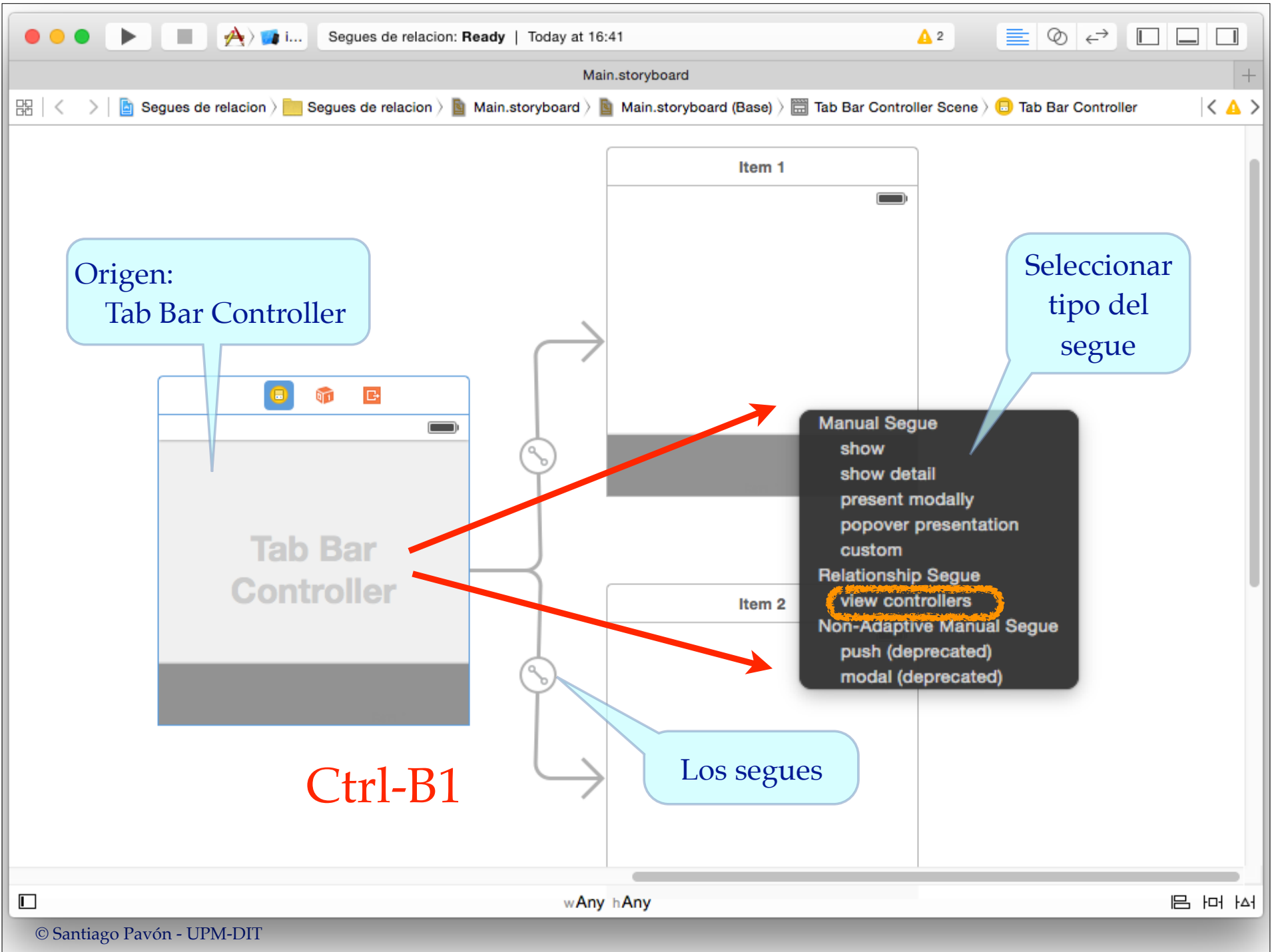
- Origen: el VC**: Points to the yellow square segue icon in the top toolbar.
- Origen: celda o accesorio**: Points to the blue square segue icon in the top toolbar.
- Origen: UIControl**: Points to the blue square segue icon in the top toolbar.
- Origen: reconocedor de gestos**: Points to the blue square segue icon in the top toolbar.
- Crear segues unwind**: Points to the red square segue icon in the top toolbar.
- Ctrl-B1**: A red text label with arrows pointing to the segue icons.
- Poner un identificador único a cada segue**: Points to the 'Identifier' field in the 'Storyboard Segue' inspector, which contains the text 'Continuar'.
- Seleccionar tipo del segue**: Points to the 'Segue' dropdown menu in the 'Storyboard Segue' inspector, which is set to 'Present Modally'.
- Los segues**: Points to the 'Segue' menu that appears when a segue icon is clicked, listing various segue types like 'Selection Segue', 'Accessory Action', and 'Non-Adaptive Selection Segue'.
- Los segues unwind solo se ven aquí**: Points to the 'Unwind segue to Scene...' options in the bottom-left toolbar.

The 'Storyboard Segue' inspector on the right shows the following configuration:

- Identifier: Continuar
- Segue: Present Modally
- Presentation: Default
- Transition: Default
- Animates:

The segue menu also shows options for 'Cancelar' and 'OK' at the bottom.





UIStoryboardSegue

- Es la clase que representa los segues.
- Los segues de tipo transición (**show**, **show detail**, **modal**, **popover**, y **custom**) crean una nueva instancia VC que se muestra tras la transición.
 - NO DEBEN usarse para volver hacia atrás a escenas anteriores.
 - Para volver hay que usar un segue de tipo **unwind** o hacerlo programáticamente.
- Propiedades de la clase **UIStoryboardSegue**
 - **identifier** - Identificador único para diferenciar los segue.
 - **destination** - VC destino de la transición.
 - **source** - VC origen de la transición.

Disparar un Segue

- Un segue se puede disparar:
 - al tocar un UIControl (ej: pulsar un botón)
 - al seleccionar una celda de una tabla.
 - al tocar el accesorio de la celda de una tabla.
 - cuando un reconocedor de gestos reconoce un gesto.
 - programáticamente llamando al siguiente método de los View Controllers:

```
func performSegue(withIdentifier identifier: String?,  
                  sender: Any?)
```

- Cuando se dispara un segue se invoca automáticamente en el VC origen el método

```
func prepare(for segue: UIStoryboardSegue,  
             sender: Any?)
```

- para que configuremos lo que sea necesario de la transición.

Prepararse para un Segue

- Cuando se dispara un segue, en el VC origen se ejecuta el método:

```
func prepare(for segue: UIStoryboardSegue,  
             sender: Any?)
```

- Este método lo sobrescribiremos para añadir las sentencias que necesitamos para configurar el VC destino (*o lo que necesitemos hacer*).
 - El argumento **segue** referencia al segue disparado.
 - Propiedades de **UINavigationController**:
 - **identifier** - String usado como identificador único del segue.
 - **source** - VC del que parte el segue.
 - **destination** - VC al que apunta el segue.
 - Consultar el valor de **segue.identifier** para identificar cuál es el segue disparado.
 - **No olvidar asignar un identificador diferente a cada segue para poder saber cuál ha sido el segue que se ha disparado.**
 - El argumento **sender** típicamente apunta al objeto que disparó el segue
 - Puede ser un `UIControl`, el VC origen, un reconocedor de gestos, ...

Muy Importante

- Cuando se invoca **prepare (for: sender:)**
 - Se crea automáticamente una instancia del VC destino, es decir, se crea un objeto nuevo.
 - Nosotros no tenemos que crear este objeto programáticamente.
 - Cada vez que se dispara el segue se crea un objeto VC destino nuevo y **distinto**.
 - Normalmente, en este método solo hay que escribir el código necesario para configurar el VC destino que nos han creado.

@IBSegueAction

- Introducido en Xcode 11.
 - Se necesita iOS 13.
- Identifica a un método que se llama durante el disparo de un segue para crear el segue destino.

@IBSegueAction

```
func showDestino(_ coder: NSCoder,  
                 sender: Any?,  
                 segueIdentifier: String?)  
                 -> DestinoViewController?
```

- Este método se usa para crear y configurar el view controller destino.
 - Si se usa este método, no es necesario usar `prepare(for:segue)`.
 - Aunque se sigue llamando.
 - Tampoco es necesario asignar un identificador a los segues.
- Este método devuelve el VC destino creado.
 - Si devuelve nil, UIKit llama a `init(coder:)` para crear el vc destino.

- La configuración del vc destino se puede hacer:

- asignando valores a las propiedades del vc destino.

```
@IBSegueAction func showDestino(_ coder: NSCoder, sender: Any?,
    segueIdentifier: String?) -> DestinoViewController? {
    let svc = DestinoViewController(coder: coder)
    svc?.titulo = "Hola"
    return svc
}
```

- Usando inicializadores personalizados (constructores) del VC destino.

```
@IBSegueAction func showDestino(_ coder: NSCoder, sender: Any?,
    segueIdentifier: String?) -> DestinoViewController? {
    DestinoViewController(coder: coder, titulo: "Hola")
}
```

- Hay que crear el inicializador personalizado en la clase vc destino:

```
let titulo: String
init?(coder: NSCoder, titulo: String) {
    self.titulo = titulo
    super.init(coder:coder)
}
required init?(coder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}
```

- El primer parámetro de inicializador debe ser `coder: NSCoder`.
- En el resto de parámetros se pasarán los valores para inicializar el vc destino.
 - Las propiedades donde se guardan los valores de configuración ya no tienen que ser optional porque su valor se asigna en el inicializador de la clase.
 - También pueden ser constantes let.
- El inicializador personalizado debe llamar a `super.init(coder: coder)`.

- Los métodos @IBSegueAction se crean interactivamente en el storyboard.
 - Estando en el modo Asistente del editor:
 - Arrastrar con Control-B1 desde el segue hasta el punto del código del VC origen en el que se quiere generar el método @IBSegueAction.
 - Arrastrar con Control-B1 desde el segue hasta encima de un método @IBSegueAction ya existente en el VC origen.
 - Arrastrar con Control-B1 desde el segue hasta el icono del VC origen para desplegar un menu con los métodos @IBSegueAction ya existentes, y seleccionar uno.
 - ...
- El método segue action tiene tres parámetros:
 - El primero es NSCoder y es obligatorio.
 - El segundo es el sender y el tercero es el identificador del segue.
 - Pueden omitirse si no se necesitan.

Cancelar un Segue

- La ejecución de un segue puede cancelarse sobrescribiendo en el VC origen el método:

```
func shouldPerformSegue(withIdentifier identifier: String?,  
                           sender: Any?) -> Bool
```

- Permite realizar algún tipo de validación antes de permitir que se realice el segue.
- Devolver false para evitar que se dispare el segue.

Unwind Segues

- ¿Cómo se retrocede para mostrar uno de los VC anteriores por los que hemos pasado?
 - Programáticamente o con segues **unwind**.
- ¿Cómo se devuelve información desde un VC destino al VC origen?
 - Tradicionalmente se hace usando un protocolo y el patrón delegado.
 - El VC destino tiene al VC origen como su delegado,
 - Cuando hay que volver, el VC destino invoca en su delegado algún método del protocolo delegado.
 - El delegado (el VC origen) ejecuta el método invocado.
 - En este método actualizará algunos datos y destruirá el VC destino.
 - Los unwind segues son un mecanismo más sencillo de realizar esta misma tarea.
 - Permiten volver a alguna de las escenas anteriores por las que se ha ido avanzando hasta llegar a la escena actual,
 - y también permiten devolver datos.
 - Recuérdese que otra posibilidad (más sencilla y moderna) de obtener los datos, o realizar cualquier tarea, entre distintos VC es usar closures.
- Los segues unwind se crean enlazando con los IBActions que se ofrecen en el icono **Exit** del *scene docs* de los VC.



- Para crear segues de tipo unwind:

- Primero hay que crear en los VCs a los que se desee volver, métodos IBActions con un argumento de tipo UIStoryboardSegue.

```
@IBAction func methodName(_ segue: UIStoryboardSegue)
```

- o usando la siguiente signature alternativa:

```
@IBAction func methodNameWithSegue(segue: UIStoryboardSegue)
```

- Estos métodos simplemente crean puntos a los que puede volver.

- Una vez creados estos métodos, cada segue unwind se crea:

1. **Ctrl-Arrastrando** desde el elemento que va a disparar el segue hasta el icono **Exit** de su propio VC.



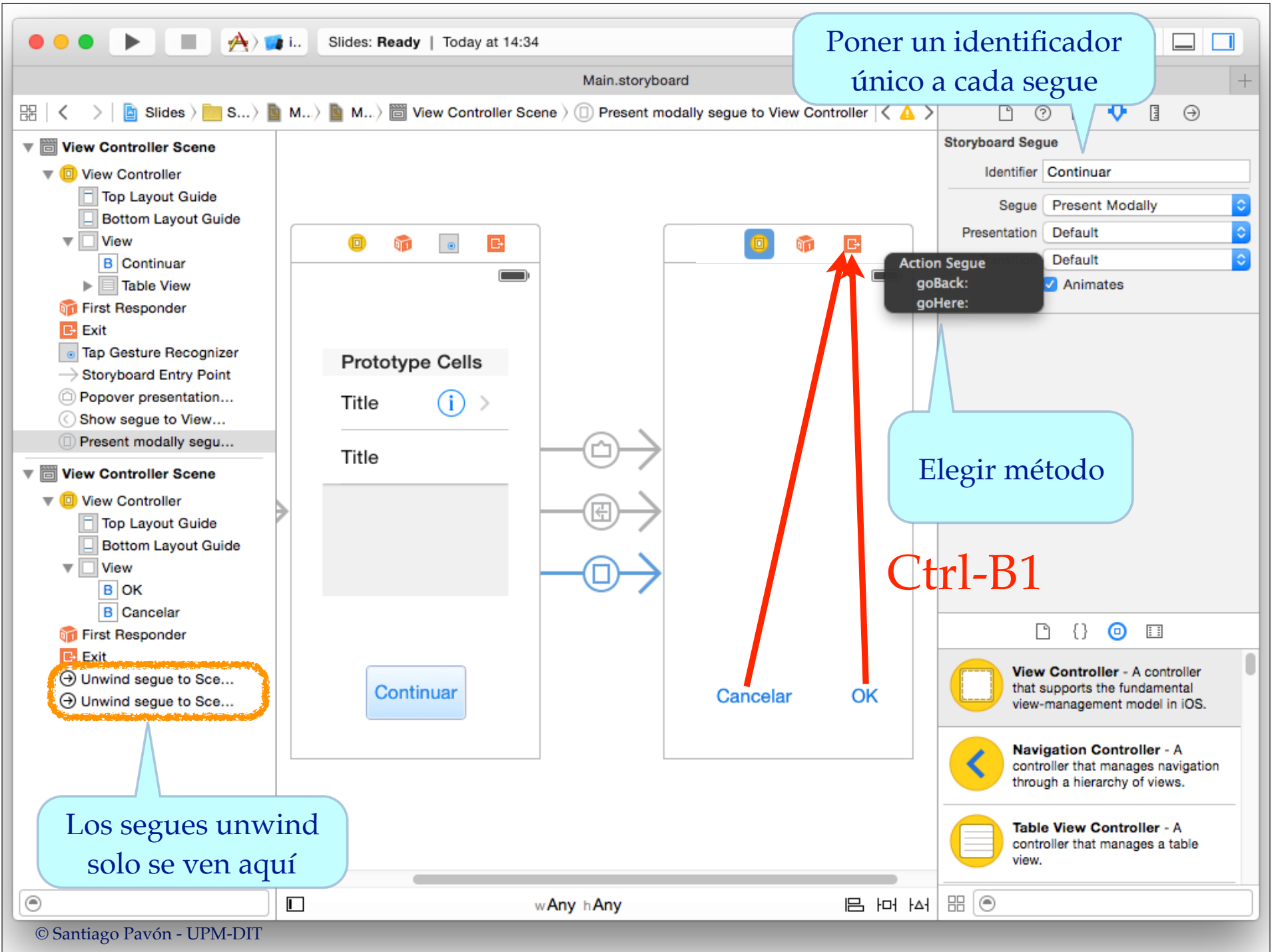
2. Entonces aparece un menú con todas las IBActions creadas anteriormente en los VCs.

1. Y hay que elegir en este menú la IBAction del VC al que vamos a volver.

3. No olvidar asignar un identificador al segue creado:

1. Seleccionar el segue creado en el Document Outline del storyboard

2. Y usar el inspector de atributos para introducir el valor del identificador.




- ¿Qué ocurre cuando se dispara un segue unwind?
 - Primero se busca cuál es el VC destino.
 - El VC más cercano que tiene una acción unwind con el nombre adecuado.
 - Después se ejecuta **prepare(for segue:sender:)** en el VC origen.
 - Después se ejecuta la **IBAction unwind** en el VC destino.
 - Y finalmente se realiza el segue:
 - Se destruyen los VC que abandonamos y se muestra el VC destino
 - Se visualiza una animación con el cambio de VCs.
- **Cuidado: No confundir el VC origen y destino en un segue unwind:**
 - El **VC origen** es el VC desde el que parte el segue unwind
 - es el VC que hizo el papel de VC destino en un segue anterior.
 - El **VC destino** es alguno de los VC anteriores por los que fuimos pasando
 - y que en algún instante fue el VC origen de otro segue.

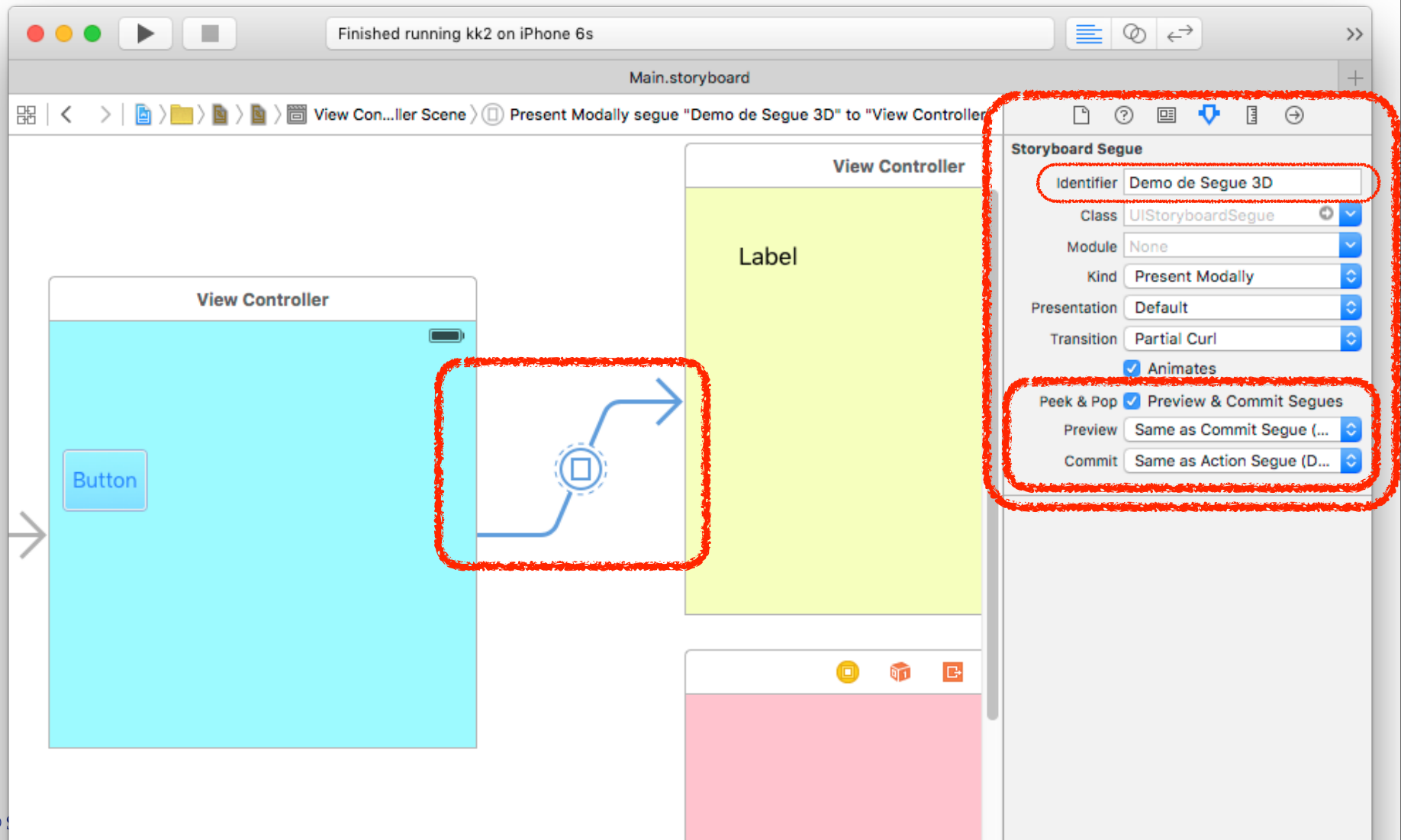
Container View

- Un VC puede mostrar en su vista varias subviews:
 - labels, botones, image views, ...
 - y también puede mostrar Container Views.
 - Un Container View es una UIView que muestra la view del VC al que apunta.
 - Es decir: lo que vemos en un Container View es la view de otro VC.
 - Un Container View usa un segue de tipo **Embed** para apuntar al VC a mostrar.
- Si un VC tiene Container View, entonces cuando se carga este VC, se disparan los segues de tipo embed de los Container Views que contiene.
 - Podemos añadir un método **prepare(for segue:sender:)** en el VC padre para interceptar los segues embed y configurar lo que necesitamos.
 - No olvidad poner un identificador a cada uno de los segues embed a interceptar para identificarlos.
- Si un VC tiene varios Container Views, no se sabe en orden se dispararán los segues embed.
 - No debemos hacer nada que dependa del orden en que se disparan los segues embed.
 - Si necesitamos configurar algo una vez que se hayan disparado todos los segues embed, podemos hacerlo en el método `viewDidLoad` del VC padre.
 - `viewDidLoad` se llama después de ejecutar los `prepare(for segue:sender:)` y `viewDidLoad` de todos los VC hijos.

Toques 3D en Segues: Peek y Pop

- Los elementos de un VC pueden reaccionar a la presión que se ejerce sobre ellos.
 - Dependiendo de la presión ejercida se mostrará inicialmente la previsualización de un contenido (Peek), una lista de acciones rápidas que pueden ejecutarse, o realizar una transición a otro VC (Pop).
- Este comportamiento 3D puede añadirse a los segues existentes de forma programática o desde el inspector de atributos del IB.
 - Con IB, en el inspector de atributos hay que activar la opción Peek & Pop
 - Preview & Commit Segues.
 - El segue se muestra rodeado de un círculo de rayas: 
 - Notas:
 - Soportado en iOS 9.1 y posteriores.
 - No soportado en el simulador Xcode 7.1. ¿Cuándo se soportará?

- Por defecto, el comportamiento y el identificador del segue Pop-Commit es el mismo que el del segue original, y comportamiento y el identificador del segue Peek-Preview es el mismo que el del segue Pop-Commit:



- Pero también puede personalizarse cuáles serán los VC usados por los segues Peek y Pop:

