



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS Storyboards

IWEB,LSWC 2013-2014
Santiago Pavón

ver: 2012.10.19 p1

Storyboard

iOS 5

Storyboard

- Es un fichero donde se describen todas las pantallas de la aplicación.
 - Es como tener todos los XIB juntos.
- Cada pantalla (VC u otro controller) se llama escena.
- Paso de una escena a otra usando segue.
 - Modal (Presentación modal)
 - Push (Navigation Controller)
 - Custom
- Las escenas se pueden embeber en un Navigation Controller o Tab Bar Controller:
 - Seleccionar la opción deseada en *Menu Editor > Embed In >*
 - Se muestran relaciones entre los controladores involucrados
 - Son segue de relación:
 - Relationship - Varios tipos (Navigation, Tab, Master/Details, ...)
- La aplicación comienza con el View Controller inicial.
 - Se identifica con una flecha que lo señala.

Crear Tablas

- Las Tablas pueden crearse para que usen
 - Celdas prototipo
 - las celdas de la tabla se crean copiando estas celdas.
 - pueden reutilizarse celdas ya usadas.
 - Celdas estáticas
 - la tabla sólo tiene las celdas que hemos creado con IB.
- Si el estilo de celda es personalizado, para acceder a las views incluidas en la celda, podemos:
 - usar un tag para identificar cada subview.
 - crear una clase derivada de UITableViewCell y crear outlets para las subviews

UIStoryboardSegue

- Un segue representa la transición desde un VC a otro VC.
 - No permiten volver al VC inicial.
 - Debe hacerse programáticamente.
- Propiedades de la clase UIStoryboardSegue
 - identifier
 - destinationViewController
 - sourceViewController
- Clases derivadas:
 - UIStoryboardSeguePopoverSegue
 - Propiedades: popoverController

Disparar un Segue

- Un segue se puede disparar:
 - al tocar un UIControl (ej: botón)
 - al seleccionar una celda de una tabla.
 - cuando un reconocedor de gestos reconoce un gesto.
 - programáticamente llamando a **performSegueWithIdentifier:sender:.**
- Cuando se dispara un segue se invoca automáticamente el método **prepareForSegue:sender:** para que configuremos lo que sea necesario.
 - El VC destino no tenemos que crearlos nosotros, nos lo dan creado.
- En IB podemos crear los segues **ctrl-arrastrando** desde controles, reconocedores de gestos, celdas de una tabla.
- Los segues pueden crearse con el VC como origen.
 - Ejemplo donde es necesario crear el segue en un VC:
 - Si queremos crear un segue que se origine en un sitio donde no puede ponerse un segue
 - por ejemplo, el accesorio de las celdas de una tabla
 - hay que crear el segue ctrl-arrastrándolo sobre el propio VC
 - y cuando se quiera disparar el segue
 - en el ejemplo, se ha tocado el accesorio de la celda
 - se debe llamar a **performSegueWithIdentifier:sender:.**

Carga Manual

- Cargar un storyboard:

```
UINavigationController * storyboard =  
    [UINavigationController storyboardWithName: @"a" bundle:nil]
```

- Cargar el VC inicial:

```
UIViewController * vc =  
    [storyboard instantiateInitialViewController];
```

- Cargar un VC por su nombre:

```
UIViewController * vc =  
    [storyboard instantiateViewControllerWithIdentifier: @"x"];
```

Storyboard iOS 6

Novedades en iOS 6

- Segues de tipo **unwind**.
 - Dar marcha atrás en la transición de un segue.
- Añadido soporte para Container Views.
 - para embeber unos VC dentro de otros VC.
 - Usa segues de tipo **embed**.
- Se puede añadir un segue desde el accesorio de una celda de tabla.
- Auto Layouts.
- Gestión de la autorotación.

Container View

- Un VC puede mostrar varias views:
 - labels, botones, image views, ...
 - y también puede mostrar Container Views.
 - Un Container View muestra como contenido la view del VC al que apunta.
 - Usa un segue de tipo embed para apuntar al VC a mostrar.
- Cuando se carga el VC padre, se disparan los segues de tipo embed de los Container Views que contiene.
 - Podemos añadir un método `prepareForSegue:sender:` en el VC padre para interceptar los segues embed y configurar lo que necesitamos.
 - No olvidad dar un nombre identificador a cada uno de los segues embed a interceptar.
- Si un VC padre contiene varios Container Views, no debemos depender 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.
 - Este método se llama después de los `prepareForSegue` y `viewDidLoad` de todos los VC hijos.

Unwind Segues

- ¿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, e invoca algún método del protocolo en él cuando ha terminado.
 - El método invocado actualiza datos y destruye el VC destino.
 - Los unwind segues son un mecanismo más sencillo de realizar esta misma tarea.
 - Volver a la escena anterior, devolviendo algún dato.
- Se crean usando el icono Exit del dock de los VC.

- Para crear un segue unwind.
 - Crear en los VCs que se desee, métodos para las acciones unwind.
 - Estos métodos son IBActions con un argumento de tipo UIStoryboardSegue.
 - (IBAction)methodName:(UIStoryboardSegue *) segue;
 - Una vez creados estos métodos, pueden crearse unwind segues mediante Ctrl-Drag desde un control hasta el icono Exit.

- ¿Qué ocurre cuando se dispara un unwind segue?
 - Primero se determina cual es el VC destino.
 - El VC más cercano que tiene una acción unwind con el nombre adecuado.
 - Después se ejecuta `prepareForSegue` en el VC origen.
 - Se ejecuta la acción unwind en el VC destino.
 - Y finalmente se realiza el segue.
 - Se ve la animación de un VC al otro.
- La realización de un segue puede cancelarse sobrescribiendo en el VC origen el método
 - (BOOL) **shouldPerformSegueWithIdentifier:(NSString *)identifier sender:(id)sender**
 - Permite realizar algún tipo de validación antes de permitir que se realice el segue.

