



POLITÉCNICA

ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS

## Presentación Modal

IWEB,LSWC 2014-2015

Santiago Pavón

ver: 2014-10-29

# Presentación Modales

- Tradicionalmente, presentar un VC de forma modal en un iPhone significaba que el VC modal se presentaba delante del VC actual ocupando toda la pantalla, y hasta que no se eliminaba el VC modal no se podía volver a interactuar con el VC inicial.
  - En un iPad hay más opciones de presentar el VC modal:
    - presentar el VC modal sustituyendo solo a una subview del VC inicial.
    - presentar el VC modal ocupando una parte de la pantalla.
- Con iOS 8 se ha unificado la forma de presentar un VC de forma modal en todos los tipos de terminales.
  - La programación es la misma y dependiendo del tipo de terminal el VC modal se presenta de la forma más adecuada.
- Al presentar un VC puede configurarse:
  - el estilo de la **transición** entre los VC.
  - el estilo de **presentación** del VC modal.
- La presentación de un VC suele hacerse diseñando sobre el fichero storyboard y usando segues.
  - Pero también puede hacerse programáticamente.

# Programáticamente

- Un VC inicial presenta un segundo VC programáticamente ejecutando las siguientes sentencias:

```
// Crear el VC a presentar cargándolo desde el storyboard:  
//     self es el VC inicial. (puede omitirse)  
//     Su propiedad storyboard apunta al objeto storyboard.  
//     SecondViewController es la clase del VC a presentar.  
//     Configurar el Storyboard ID del segundo VC como "El Segundo VC".  
if let vc2 = self.storyboard?.instantiateViewControllerWithIdentifier(  
    "El Segundo VC") as? SecondViewController {  
    // Configurar los parámetros de entrada del VC a mostrar:  
    vc2.msg = "Hola Mundo"  
  
    // Presentar el segundo VC:  
    self.presentViewController(vc2, animated: true, completion: nil)  
}
```

- Para eliminar el segundo VC ejecutar:

```
self.dismissViewControllerAnimated(true, completion: nil)
```

# Métodos

- Método para hacer que un VC presente a otro VC:

```
func presentViewController(_ viewControllerToPresent: UIViewController,  
    animated flag: Bool,  
    completion completion: (() -> Void)?)
```

- Parámetros:

- El VC a presentar.
- Un booleano para indicar si se desea presentar de forma animada.
- Un closure que se llama después de ejecutar el método **viewDidAppear** del VC presentado.
  - Pasar **nil** si no se quiere usar.
- Este método lo llama el VC que desea presentar el nuevo VC.

- Método para quitar del interface el VC presentado antes:

```
func dismissViewControllerAnimated(_ flag: Bool,  
    completion completion: (() -> Void)?)
```

- Parámetros:

- Un boolean para indicar si se desea quitar de forma animada o no.
- Un closure que se ejecutará después del método **viewDidDisappear** del VC presentado.
  - Pasar **nil** si no se quiere usar.
- Este método se puede ejecutar sobre el VC presentado, o sobre el VC que presentó a este VC, o sobre el VC que se ocultó para mostrar a este VC.

# Propiedades

- Un VC puede acceder al VC que lo presentó modalmente accediendo a su propiedad:

```
var presentingViewController: UIViewController? { get }
```

- Un VC puede acceder al VC que ha presentado modalmente accediendo a su propiedad:

```
var presentedViewController: UIViewController? { get }
```

- Un VC guarda en esta propiedad la referencia del VC que ha pedido que se presente modalmente.
  - pero el valor de esta propiedad también se asigna en el VC que se ha ocultado para presentar el VC modal. Normalmente son el mismo.

# Estilo de la Transición

- El VC modal a mostrar puede aparecer / desaparecer de forma animada.
- El estilo de la animación se indica usando la propiedad:

```
var modalTransitionStyle: UIModalTransitionStyle
```

- Valores:
  - **UIModalTransitionStyle.CoverVertical**
    - Aparece y desaparece deslizándose desde la parte inferior de la pantalla
  - **UIModalTransitionStyle.FlipHorizontal**
    - La pantalla rota sobre su eje vertical central.
  - **UIModalTransitionStyle.CrossDissolve**
    - La pantalla se desvanece cuando aparece o desaparece el VC modal.
  - **UIModalTransitionStyle.PartialCurl**
    - Una esquina de la pantalla simula el efecto de pasar página. Al tocar el borde doblado se deshace la presentación modal.

# Estilo de la Presentación

- Por defecto, el VC presentado modalmente ocupa toda la pantalla.
  - pero podemos configurar otros estilos de presentación.
- El estilo de presentación se indica usando la propiedad:

```
var modalPresentationStyle: UIModalPresentationStyle
```

- Valores:

```
.FullScreen           // ocupa toda la pantalla.  
.OverFullScreen       // sin quitar el presenting VC.  
.PageSheet           // no ocupa todo el ancho de la pantalla.  
.FormSheet           // ocupa solo el centro de la pantalla.  
.CurrentContext      // lo que ocupe la view del presenting VC.  
.OverCurrentContext  // sin quitar la presenting VC.  
.Popover              // presentar como un popover.  
...
```

- También puede usarse la propiedad **definesPresentationContext** para indicar que VC por encima en la jerarquía proporciona el contexto para presentar el VC modal.

# Presentación Adaptativa

- En función del Size Class se puede:
  - indicar que estilo de presentación se desea tener.
  - o presentar modalmente un VC distinto.
- La presentación adaptativa se realiza usando un objeto **UIPresentationController** y el delegado **UIAdaptivePresentationControllerDelegate**.



# Presentación con Segues

- Para presentar modalmente un VC usar un segue **Present Modally**:
  - Crear un segue de tipo **Present Modally** que conecte el VC origen con el VC destino.
    - El origen del segue será normalmente un UIControl o un reconocedor de gestos perteneciente al VC origen.
      - También podría ser el propio objeto VC origen si el segue se va a disparar programáticamente.
    - El destino del segue es el propio VC a mostrar de forma modal.
- Para volver hacia atrás puede usarse un segue de tipo **unwind**:
  - Crear un segue en el VC mostrado de forma modal.
    - El origen del segue será un UIControl, un reconocedor de gestos, el propio VC, ...
    - El destino del segue es el icono Exit y seleccionando un método de vuelta perteneciente a un VC anterior.
- Consultar el tema sobre Segues para profundizar en los detalles de su uso.

# Flujo de Datos

- Los view controllers por los que navego pueden necesitar 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, ...
    - No 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.

# Demos

# Demos

- 1- Color del pelo:
  - v1: Sin segues.
  - v2: Con segues modal y unwind.
- 2 - Días de vida:
  - Segues modal y unwind.
- 3 - Desayuno
  - Con segues modal y unwind, y con delegación.



