



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS

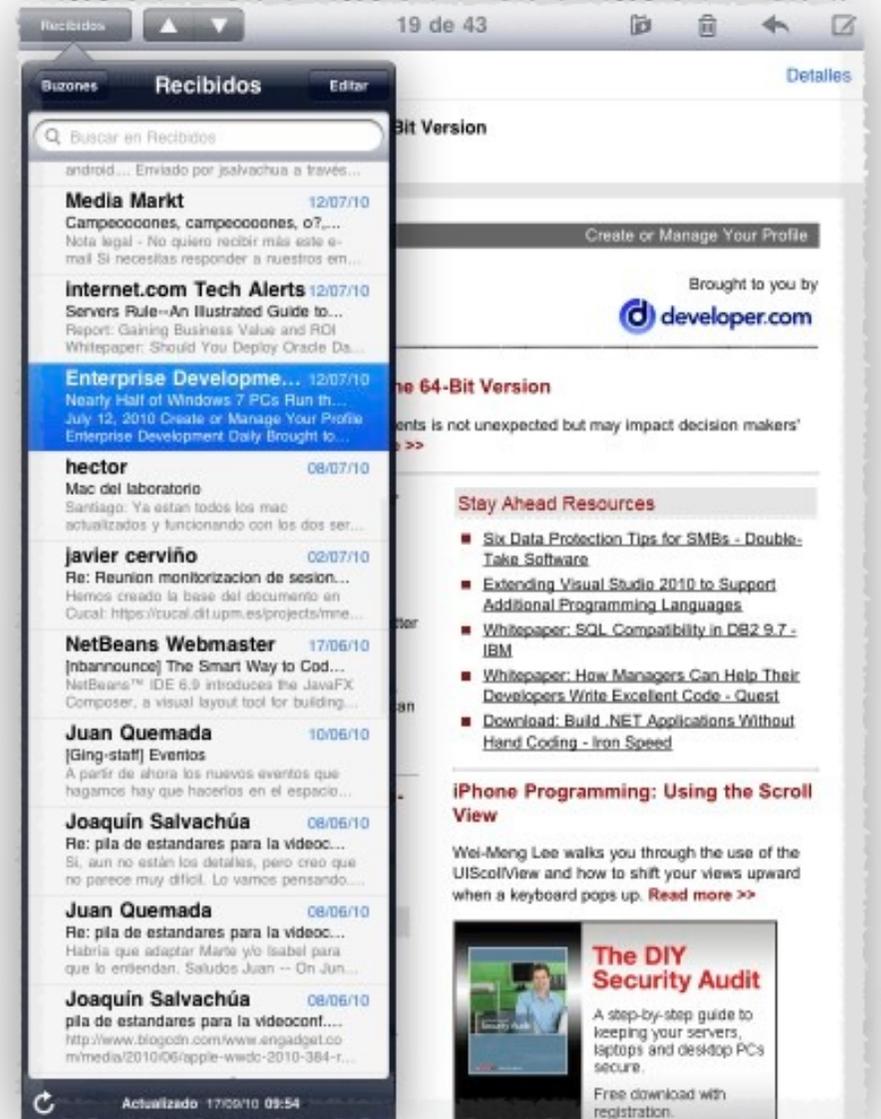
Popover Presentation Controller

IWEB 2016-2017
Santiago Pavón

ver: 2016.09.02

¿Qué es un Popover?

- Es una ventana que se crea sobre la pantalla y en la que se muestra un View Controller.
 - La ventana tiene una flecha que apunta al elemento que la hizo aparecer.
 - Por defecto, el popover desaparece al tocar fuera de él.



Mostrar un VC como un Popover

Programáticamente

- Para gestionar la presentación de un VC en un popover se usa un objeto de la clase **UIPopoverPresentationController**.
- Se programa igual que la presentación modal de un VC, pero usando el estilo **UIModalPresentationStyle.popover**.
 - La presentación de VC se adaptará dependiendo del size class horizontal del terminal:
 - si es **Regular**, el VC se muestra usando un popover.
 - si es **Compact**, el VC se muestra modalmente en modo Full Screen.
 - No es necesario crear manualmente el objeto **UIPopoverPresentationController**.
 - Se crea automáticamente, pero SOLO cuando la presentación usa un popover, es decir, cuando el size class horizontal es Regular. Es un Optional.
 - Se guarda en la propiedad **popoverPresentationController** del VC a presentar.
 - Finalmente, si existe la propiedad **popoverPresentationController** del VC a mostrar, es obligatorio asignar valores a sus propiedades **sourceView** y **sourceRect**, o a su propiedad **barButtonItem**.
 - Si no se asignan estos valores, la aplicación se muere.

```

class ViewController: UIViewController {

    @IBAction func showPopover(sender: UIButton) {

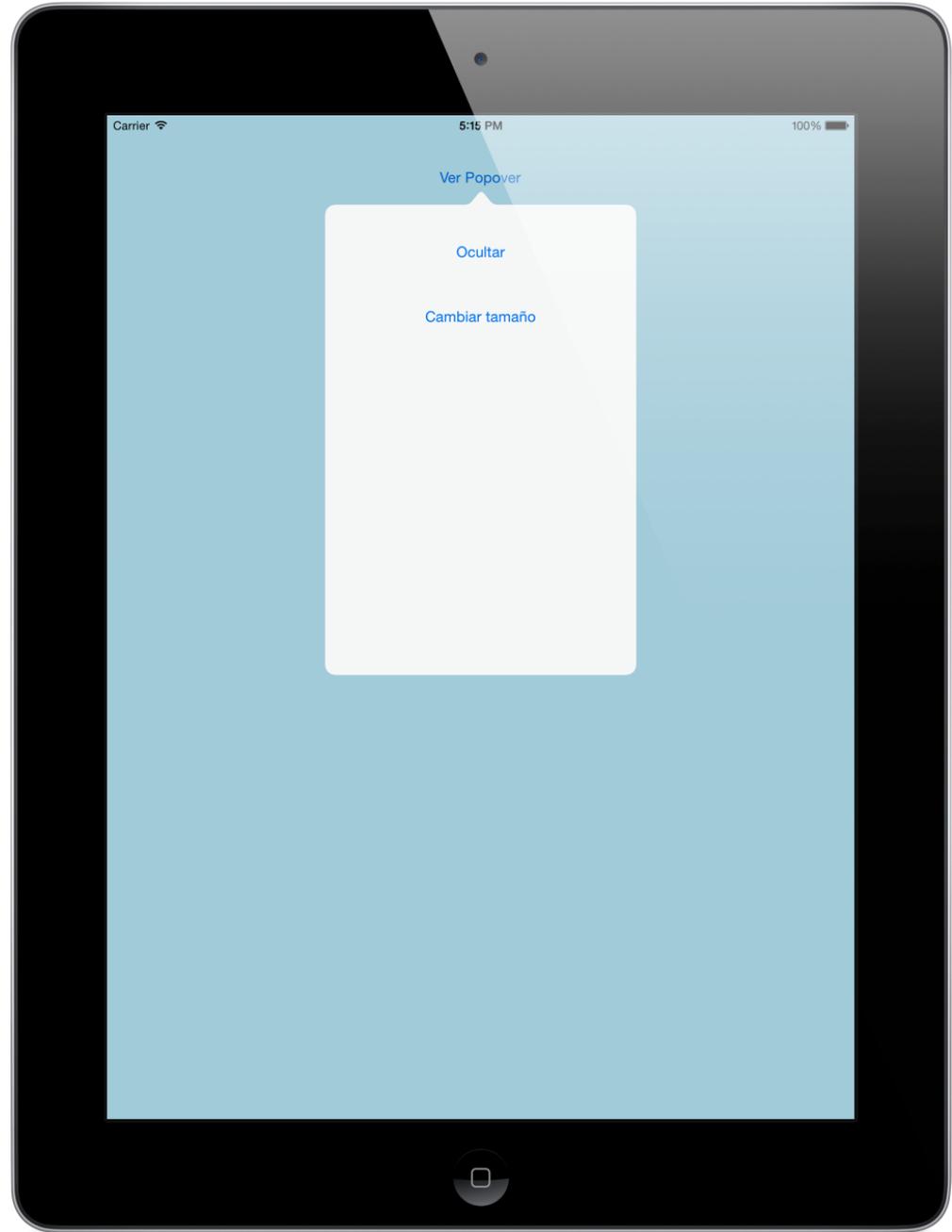
        // Mostrar en un popover un VC de la clase OtherViewController.
        // OtherViewController ha sido diseñado en el Storyboard, y el
        // valor de su propiedad Storyboard ID es "Other VC".
        if let ovc = storyboard?.instantiateViewController(
            withIdentifier: "Other VC")
            as? OtherViewController {

            ovc.modalPresentationStyle = .popover

            // Configurar el popoverPresentationController.
            // Es un Optional. Solo si Size Class Horizontal es Regular.
            ovc.popoverPresentationController?.sourceView = view
            ovc.popoverPresentationController?.sourceRect = sender.frame

            present(ovc, animated: true)
        }
    }
}

```



UIPopoverPresentationController

- Las propiedades que podemos configurar en estos objetos son:

sourceView

- Es la vista que contiene el rectángulo al que debe apuntar el popover.
- Usar junto con sourceRect.

sourceRect

- Es el rectángulo dentro de sourceView al que debe apuntar el popover

barButtonItem

- Es el UIBarButtonItem al que debe apuntar el popover.
- Usar cuando no se usen sourceView y sourceRect.

permittedArrowDirections

- Son las direcciones hacia las que puede apuntar el popover.
- El valor es una máscara con los valores **UIPopoverArrowDirection.Down**, **.Up**, **.Left** y **.Right**.
 - Por defecto es **UIPopoverArrowDirection.Any**.

passthroughViews

- Vistas que al tocarlas no cierran el popover.

backgroundColor, . . .

- Es obligatorio configurar por lo menos **sourceView** y **sourceRect**, o configurar **barButtonItem**.

Tamaño del Popover

- El tamaño del popover puede dejarse sin especificar.
- También puede indicarse el tamaño usando la propiedad **preferredContentSize** del VC a mostrar.
 - Si el VC a mostrar es un contenedor, deberá ajustarse este valor en los VC hijos.
 - Por ejemplo, si el popover presenta un Navigation Controller, lo que hay que configurar es la propiedad preferredContentSize del VC que esté mostrando el Navigation Controller.
- Si se cambia el valor de esta propiedad mientras se está mostrando el popover, éste se redimensionará.

Cerrar el Popover

- Podemos cerrarlo de la misma manera que se hacía con las presentaciones modales, usando:

```
dismiss(animated flag: Bool,  
         completion: (@escaping () -> Swift.Void)? = nil)
```

Con IB - Storyboards

- Objetivo: Editar en Storyboard la presentación de un View Controller en un Popover.
- Editar el storyboard para crear un segue de tipo: **Present as Popover**.
 - Ctrl+Arrastrar desde el elemento origen del segue (botón, VC, etc...) hasta el VC que queremos mostrar en un Popover.
 - Editar las propiedades de segue:
 - Identificador único del segue.
 - Dirección de la flecha del segue apuntando al elemento que abrió el popover.
 - Anchor al que apunta la flecha del popover.
 - Añadir views a Passthrough (views que no ocultan el popover al interactuar con ellas).
 - ...
- Editar el VC origen para configurar la transición sobrescribiendo el método **func prepare(for segue: UIStoryboardSegue, sender: Any?)**
 - Si se desea acceder al objeto **UIPopoverPresentationController**, usar:
`segue.destination.popoverPresentationController`
 - Recordad que hay que asignar un identificador único al segue creado
 - prepare(for:sender:) lo usa para identificar cuál ha sido el segue disparado.

Demo

- Crear un nuevo proyecto con la plantilla Single View Application.
- Editar el storyboard:
 - Añadir un botón al VC creado con el proyecto con el título "Ver Fecha".
 - Añadir un nuevo VC al Storyboard,
 - crear una clase nueva llamada **DateViewController**.
 - en el fichero **DateViewController.swift**
 - que deriva de **UIViewController**.
 - Con el inspector de atributos del storyboard, asignar esta clase al VC creado.
 - Crear la siguiente propiedad en **DateViewController**:
 - **var date: Date?**
 - Editar el storyboard para crear una UILabel en **DateViewController**,
 - crear un outlet a la label
 - en **viewDidLoad** poner como texto de la label el valor de la propiedad **date**.
 - Crear un segue de tipo **Present as Popover** desde el botón "Ver Fecha" del primer VC hasta el DateVC.
 - Ctrl+Arrastrar desde el botón hasta **DateViewController**.
 - Asignar un identificador al segue: "Show Date".
 - Sobrescribir el método **prepare(for:sender:)** del primer VC para asignar a la propiedad **date** del **DateViewController** el valor de la fecha actual.
 - Editar con el inspector las propiedades del segue.
 - Asignar un valor a la propiedad de **preferredContentSize** de **DateViewController** en el método **viewDidLoad**.
 - Crear un botón en **DateViewController** que llame a **dismiss(animated:completion:)** para cerrar el popover.
 - Alternativa para cerrar el popover: Crear un segue unwind.
 - ...

UIPopoverPresentationControllerDelegate

- Los objetos **UIPopoverPresentationController** tienen una propiedad llamada **delegate**.
- Este delegado es conforme al protocolo **UIPopoverPresentationControllerDelegate**, que tiene métodos para:
 - Realizar configuraciones adicionales cuando el popover se va a presentar.
prepareForPopoverPresentation:
 - Controlar si se permite cerrar el popover o no.
popoverPresentationControllerShouldDismissPopover:
 - Avisar cuando ya se ha cerrado el popover.
popoverPresentationControllerDidDismissPopover:
 - Avisar cuando se va a repositionar el popover.
popoverPresentationController:willRepositionPopoverTo:in:
- Además, el protocolo **UIPopoverPresentationControllerDelegate** es conforme con el protocolo **UIAdaptivePresentationControllerDelegate**, por lo que podemos implementar los métodos que configuran como se adapta la presentación de un Popover según el tamaño del terminal.

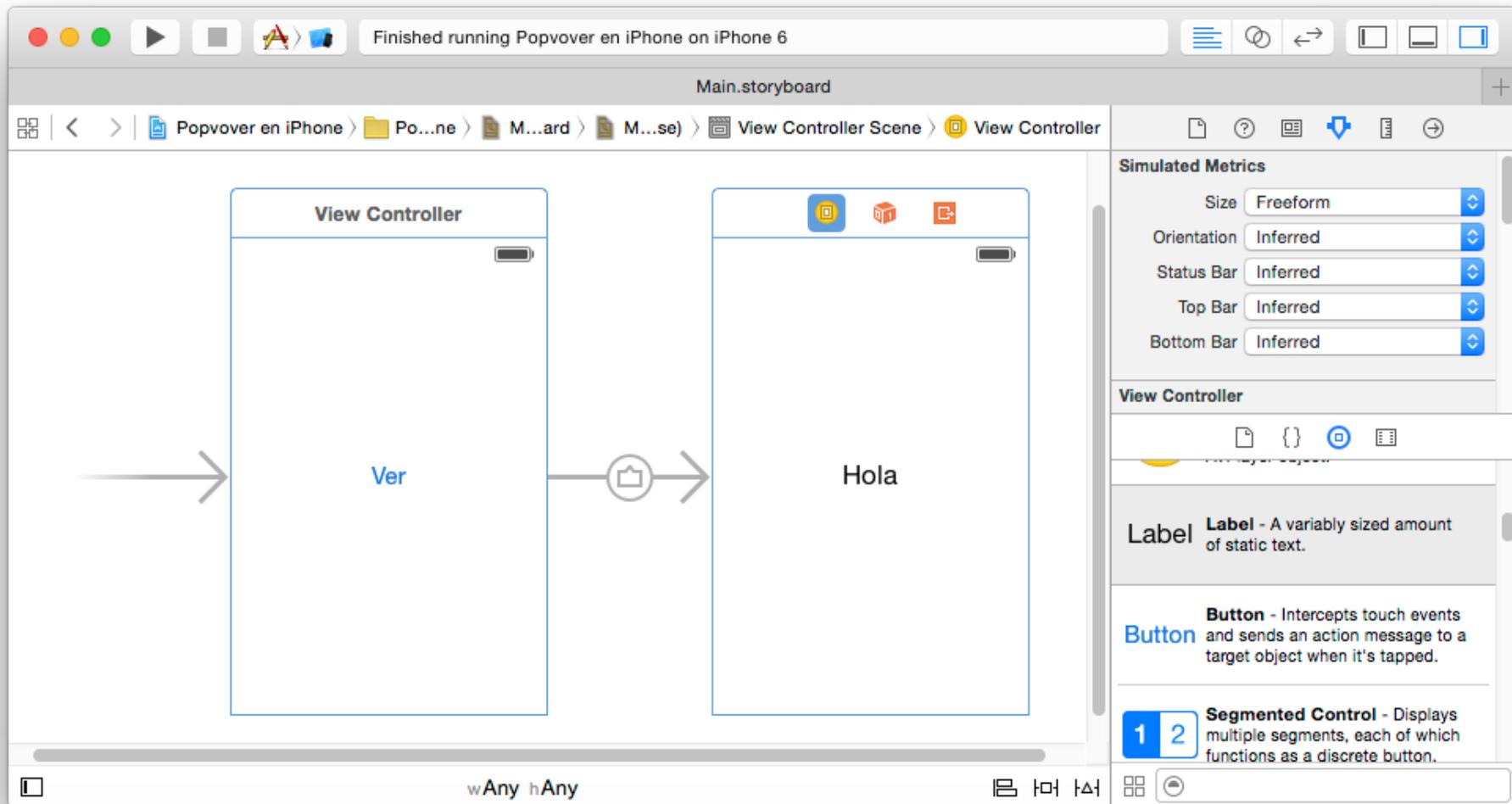
```
optional func adaptivePresentationStyle(for controller: UIPresentationController)
    -> UIModalPresentationStyle

optional func presentationController(_ controller: UIPresentationController,
    viewControllerForAdaptivePresentationStyle style: UIModalPresentationStyle)
    -> UIViewController?

. . .
```

Presentación Adaptativa

- Ejemplo: No adaptar la presentación de un Popover en un iPhone.



```

class ViewController: UIViewController, UIPopoverPresentationControllerDelegate {

    func adaptivePresentationStyle(for controller: UIPresentationController)
                                    -> UIModalPresentationStyle {
        return .none
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "Mi Segue" {
            if let pvc = segue.destination as? PopoverViewController {
                pvc.presentationController?.delegate = self
            }
        }
    }
}

```

