



PopOver

Yolanda Martínez Treviño

popOver

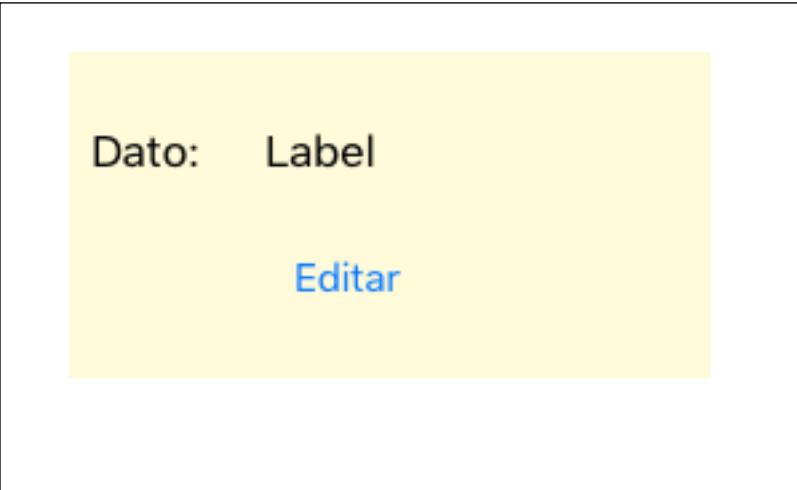
- + Se usan para presentar información temporal al usuario sin ocupar toda la pantalla. Tienen un pequeño indicador que apunta a algún elemento de la interfaz.
- + El popOver se muestra sobre el contenido que se encuentre en la pantalla en ese momento.
- + Permanece visible mientras el usuario no haga un tap afuera del popOver.
- + Originalmente era exclusivo para uso en iPad, pero en la última versión ya se puede usar también para iPhone. El valor default es que si corre en iPhone aparece como vista modal y en iPad aparece como popOver, pero se puede configurar para tener la vista como popOver también en el iPhone.

Ejemplo de como se ve un popOver



Crea una aplicación de tipo SingleView

- + Agrega al storyboard un segundo viewController, crea una clase y asóciala.
- + Coloca los elementos de interfaz que tendrán las vistas.



Dato: Label

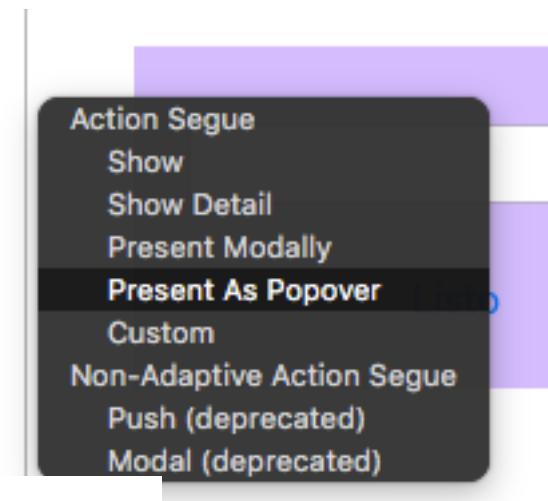
Editar



Listo

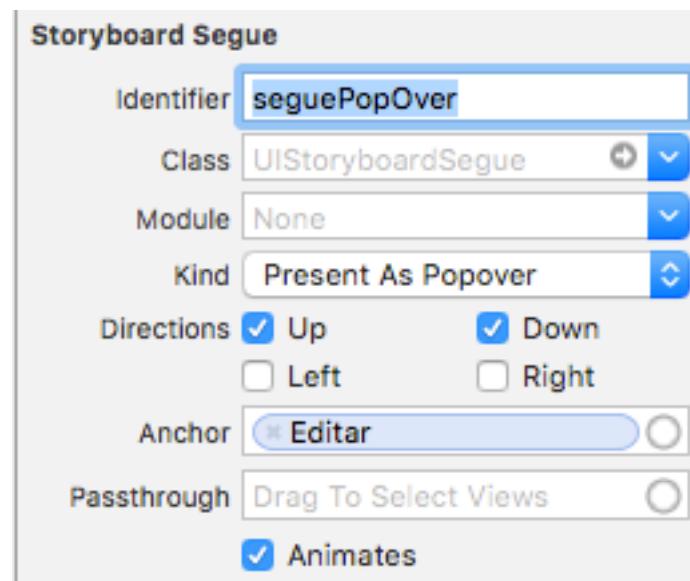
Conecta la vista como Popover

- + Agrega el segue, y al elegir tipo de segue, selecciona Present as Popover como se muestra



Configura el popOver

- + Selecciona el segue y observa el attributes inspector para realizar la configuración necesaria



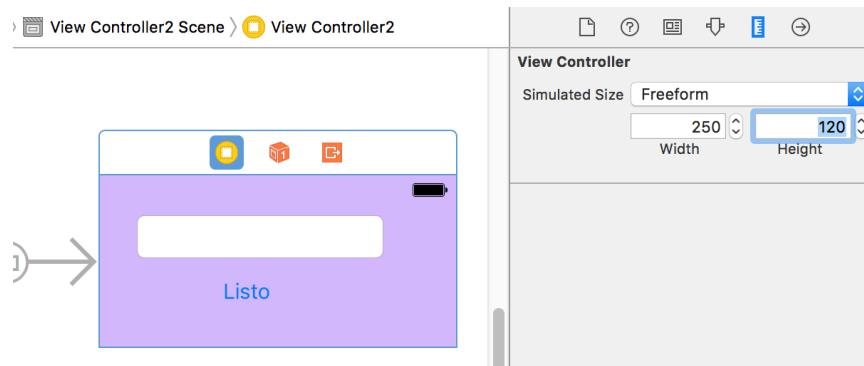
Configura el tamaño del popOver

- + El tamaño de la vista popOver generalmente es mas pequeño que el tamaño de la vista completa.
- + Es posible modificar el tamaño de la vista que se presenta como popOver estableciendo el valor para la variable `preferredContentSize` en el método `viewDidLoad` de la vista que se presenta como popOver

```
preferredContentSize = CGSize(width: 250, height: 120)
```

Configura el tamaño del popOver

- + También es posible cambiar el tamaño del view controller en el interface builder. Selecciona la vista y cambia su tamaño.



Pero de esta forma al correrlo no aparece del tamaño especificado; **es necesario definirlo también por programa**.

- + Pero es útil para fines de diseño de la interfaz.

Para que se vea como popOver en iPhone

En la clase ViewController

- + Es necesario implementar el protocolo

UIPopoverPresentationControllerDelegate

- + incluir el siguiente método

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

- + Y definir el viewController actual como el delegado para el popoverPresentationController de la vista que se presenta como popover:

```
let vistaPopOver = segue.destination as! ViewController2
```

```
vistaPopOver.popoverPresentationController!.delegate = self
```

Pasar valores entre vistas

- + Para pasar información al view controller que se presentará como popOver, se puede hacer dentro del método `prepare(for segue:)` usando el `segue.destination`
- + Para pasar información del view controller popOver a la vista que lo presentó se puede hacer usando el objeto predefinido `presentingViewController` de esta forma:

```
let vista1 = presentingViewController as! ViewController  
vista1.dato = tfDato.text!
```

popOver Presentation Controller Delegate Protocol

- + Si se requiere realizar alguna acción cuando se quita el popOver se puede implementar el protocolo

UIPopoverPresentationControllerDelegate

- + Es necesario implementar el método

```
func popoverPresentationControllerDidDismissPopover  
(popoverPresentationController: UIPopoverPresentationController)
```

IMPORTANTE: Este método no se manda llamar si se quita la vista popOver por programa

- + Y definir el delegado como el viewController

```
vistaPopOver.popoverPresentationController!.delegate = self
```

Y si quito el popOver por programa

- + Desde la vista presentada como popOver puedo llamar a un método de la vista anterior y luego quitar el popOver usando:

```
let vista1 = presentingViewController as! ViewController  
vista1.dato = tfDato.text!  
vista1.actualizaDato()  
dismiss(animated: true, completion: nil)
```

Completa el ejercicio

- + Al oprimir el botón Editar de la primera vista, aparece la segunda vista como popOver en la que se puede teclear un nuevo valor para el dato.
- + Al oprimir Listo en la segunda vista el dato debe aparecer en el label de la primera vista.