



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS Alert Controller

IWEB 2015-2016
Santiago Pavón

ver: 2014.11.18

iOS 7 y Anteriores

- En iOS 7 y anteriores teníamos **UIAlertView** y **UIActionSheet**.
 - Se usaban para presentar un popover con información, botones, entradas de texto,...
- En iOS 8 se han sustituido por un nuevo view controller: **UIAlertController**.



iOS 7
Alert Views



iOS 7
Action Sheet



UIAlertController

- **UIAlertController** es un View Controller que se muestra usando el método:
presentViewController:animated:completion:
 - Es el mismo método usado para mostrar un VC de forma modal.
- Al crear un objeto Alert Controller, en el inicializador se proporcionan los siguientes datos:
 - Un título
 - Un mensaje
 - El estilo de presentación preferido:
 - **UIAlertControllerStyle.Alert** para que se muestre como un Alert View.
 - **UIAlertControllerStyle.ActionSheet** para que se muestre como un Action Sheet.
 - Atención: UIAlertController es adaptativo:
 - Se adapta al tipo de terminal para presentarse de la forma más adecuada:
 - como un popover, como una vista modal a pantalla completa.
- A los objetos Alert Controller creados se le pueden añadir:
 - Acciones: botones que al pulsarse ejecutan una closure.
 - Y en los **.Alert** también se puede añadir Text Fields para introducir texto.

Presentar un Alert

```
class ViewController: UIViewController {  
  
    // Este VC tiene un botón que al ser pulsado ejecuta esta acción:  
    @IBAction func showAlert() {  
  
        // Crear un UIAlertController:  
        let alert = UIAlertController(title: "El Título",  
                                     message: "Hola Mundo",  
                                     preferredStyle: UIAlertControllerStyle.Alert)  
  
        alert.addAction(...) // Añadir una Action .Default  
  
        alert.addAction(...) // Añadir una Action .Destructive  
  
        alert.addAction(...) // Añadir una Action .Cancel  
  
        alert.addTextFieldWithConfigurationHandler(...) // Añadir un campo de texto  
  
        // Presentar el Alert Controller:  
        presentViewController(alert, animated: true, completion: nil)  
    }  
}
```

```
alert.addAction( // Añadir una Action .Default
    UIAlertAction(title: "OK",
        style: UIAlertActionStyle.Default,
        handler: {(aa :UIAlertAction) in
            print("Se pulsó OK")
        })
    ))
```

```
alert.addAction( // Añadir una Action .Destructive
    UIAlertAction(title: "Dar de Baja",
        style: UIAlertActionStyle.Destructive,
        handler: {(aa :UIAlertAction) in
            if let tf = alert.textFields?.first,
                let name = tf.text {
                print("Explugar a \(name)")
            }
        })
    ))
```

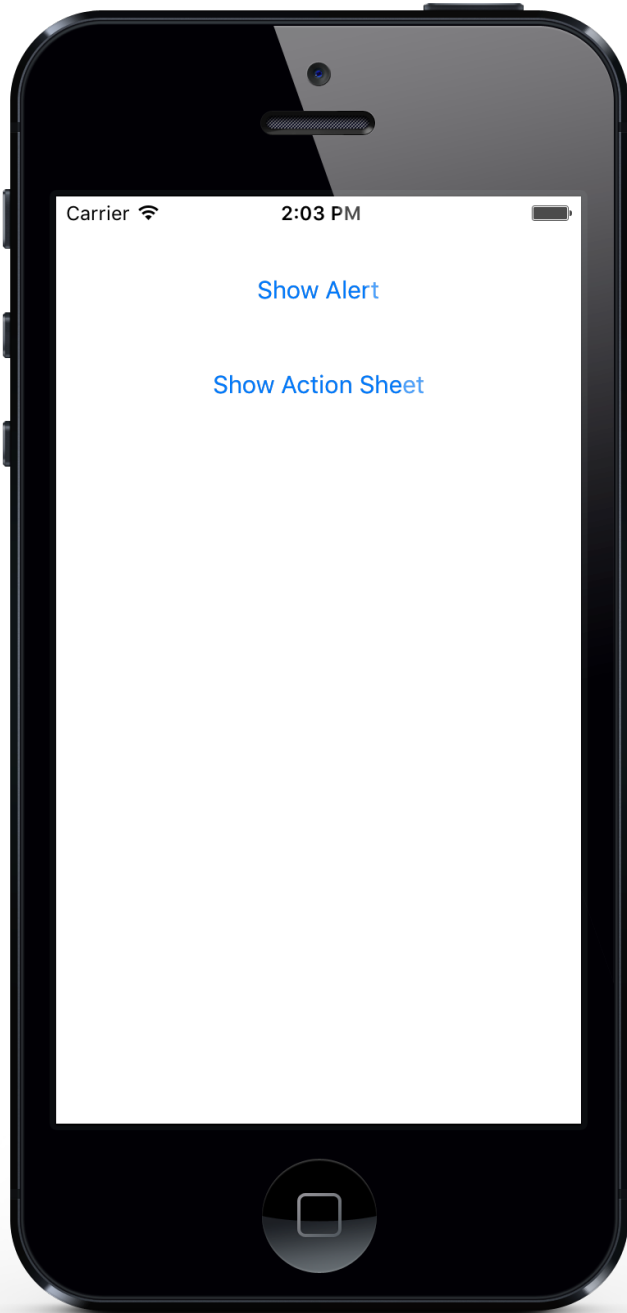
Acceder al texto del Text Field

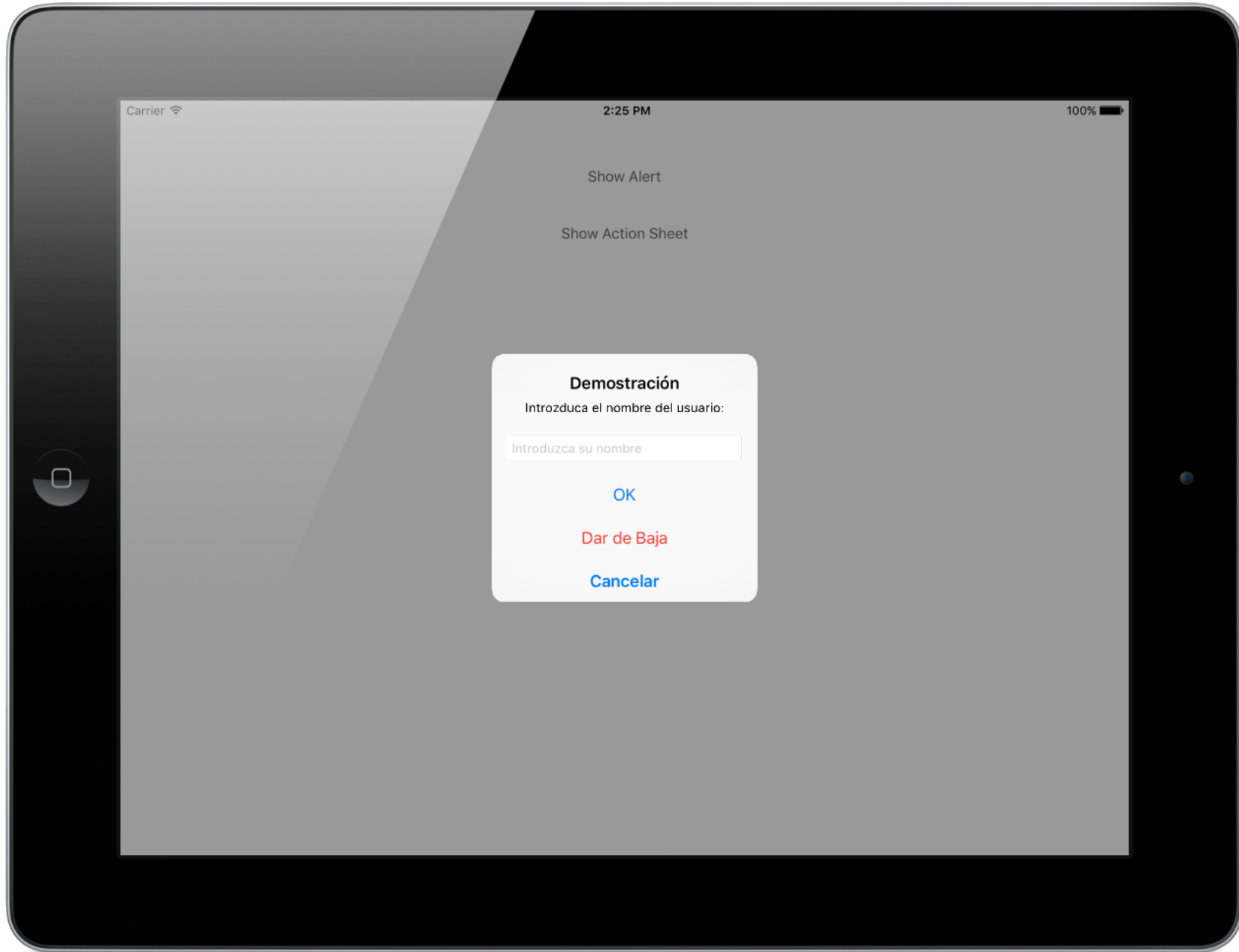
```
alert.addAction( // Añadir una Action .Cancel
    UIAlertAction(title: "Cancelar",
        style: UIAlertActionStyle.Cancel,
        handler: {(aa :UIAlertAction) in
            print("No hacemos nada")
        })
    )))
```

```
alert.addTextFieldWithConfigurationHandler( // Añadir un
                                            // campo texto
    {(tf: UITextField) in
        tf.placeholder = "Introduzca su nombre"
    })
```



Configuro el Text Field creado





Presentar un Action Sheet

- En el ejemplo anterior, para presentar el Alert Controller como un Action Sheet, hay que cambiar el estilo del Alert Controller a **UIAlertControllerStyle.ActionSheet**.
- Al ejecutar en un **iPhone**, este ejemplo funciona perfectamente.
- Pero al ejecutar en un **iPad**, la aplicación se muere.
 - En un iPad el Alert Controller se intenta presentar en un popover.
 - Por tanto, es necesario asignar valores a las propiedades **sourceView** y **sourceRect**, o a la propiedad **barButtonItem**, del **popoverPresentationController** del objeto Alert Controller.
 - La propiedad popoverPresentationController es Optional.
 - Se crea automáticamente en los View Controllers solo cuando estos se intentan presentar en un popover; pero no existe en otros casos.
- Asignamos valores a las propiedades **sourceView** y **sourceRect** de **popoverPresentationController** del Alert Controller.
 - Notar que usamos sentencias Optional Chaining.

```
// Este VC tiene un botón que al ser pulsado ejecuta esta acción:
@IBAction func showActionSheet(sender: UIButton) {

    // Crear un UIAlertController:
    let alert = UIAlertController(title: "Enviar Mensaje",
        message: "¿Cómo enviamos el mensaje?",
        preferredStyle: UIAlertControllerStyle.ActionSheet)

    // Configurar el popoverPresentationController.
    // Es un Optional. Solo existe en iPad.
    alert.popoverPresentationController?.sourceView = view
    alert.popoverPresentationController?.sourceRect = sender.frame

    // Añadir Actions:
    alert.addAction(...)
    alert.addAction(...)
    alert.addAction(...)
    alert.addAction(...)

    // Presentar el Alert Controller:
    presentViewController(alert, animated: true, completion: nil)
}
```

```
alert.addAction( // Añadir una Action .Default
    UIAlertAction(title: "email",
        style: .Default)
    {(_: UIAlertAction) in
        print("Se envia por correo")
    })
```

```
alert.addAction( // Añadir una Action .Default
    UIAlertAction(title: "Fax",
        style: .Default)
    {(_: UIAlertAction) in
        print("Se envia por Fax")
    })
```

```
alert.addAction( // Añadir una Action .Destructive
    UIAlertAction(title: "No enviar",
        style: .Destructive)
    {(_: UIAlertAction) in
        print("Descartar")
    })
```

```
alert.addAction( // Añadir una Action .Cancel
    UIAlertAction(title: "Cancelar",
        style: .Cancel)
    {(_: UIAlertAction) in
        print("No hacemos nada")
    })
```



