



POLITÉCNICA

ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS Alert Controller

IWEB 2016-2017  
Santiago Pavón

ver: 2016.09.02

# UIAlertController



← Alert

- **UIAlertController** es un View Controller que se usa para presentar modalmente un popover con información, botones, entradas de texto,...

Action Sheet →



# UIAlertController

- **UIAlertController** es un View Controller que se muestra usando el método:

`present(_ viewController: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 según el 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 de estilo **.alert** también se pueden 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: .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:  
        present(alert, animated: true)  
    }  
}
```

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

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

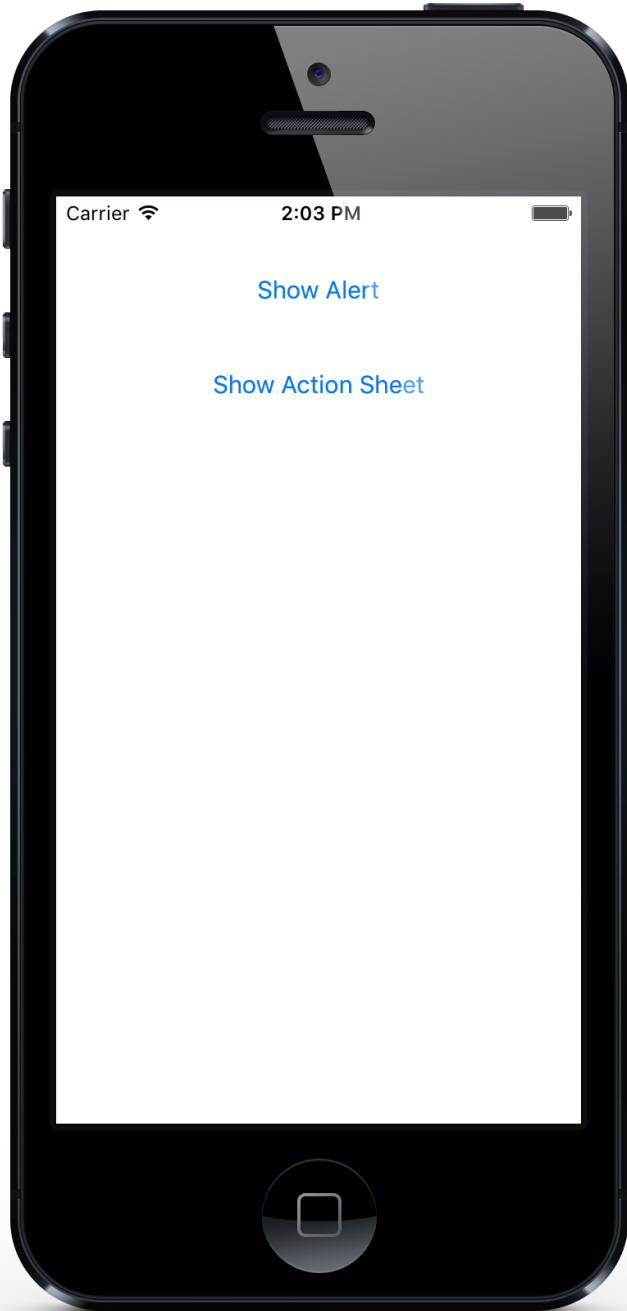
Acceder al texto del Text Field

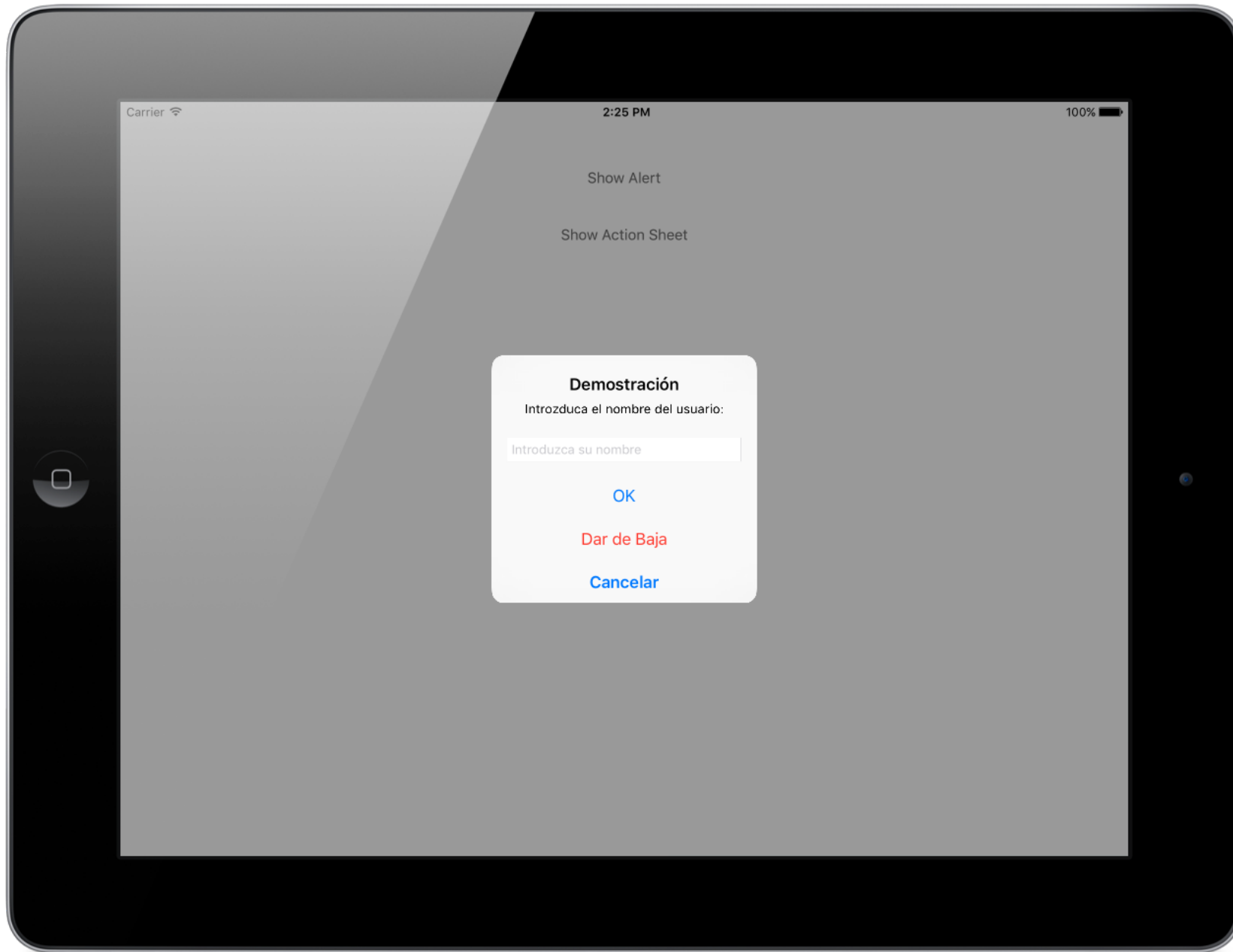
```
alert.addAction( // Añadir una Action .cancel
    UIAlertAction(title: "Cancelar",
                  style: .cancel,
                  handler: {(aa :UIAlertAction) in
                        print("No hacemos nada")
                    })
    )))
```

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



Configuro el Text Field creado







# Presentar un Action Sheet

- Para presentar el Alert Controller del ejemplo anterior como un Action Sheet, hay que cambiar el estilo del Alert Controller a **.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: .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:
    present(alert, animated: true)
}
```

```
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")
    })
```



