

MOV02 - Lab 11

Juan Pablo Arias Mora

Marzo 2021

1 Enunciado

”Todo el mundo aquí tiene la sensación de que ahora es uno de esos momentos que están influenciando el futuro.” Steve Jobs

En clases se a revisado el concepto de **Cocoapods**, por tanto la finalidad del laboratorio es presentarle un ejemplo sencillo de como implementar una biblioteca de terceros dentro de un proyecto en IOS.

1.1 Versiones Permitidas

- MacOS 10.15.7 - Inglés/Español
- Xcode 12.4
- Simulador iPhone 12 Pro Max

1.2 Creación de aplicación base

Paso 1: Cree un nuevo proyecto en Xcode, utilizando el template App dentro de iOS, en la ventana de opciones para la configuración especifica de la nueva aplicación.

Product Name : cocoapods
Organization Identifier : com.cenfotec.mov02.11
Interface : SwiftUI
Life Cycle : SwiftUI App
Languge : Swift
Use Core Data: Sin Seleccionar
Include Test: Sin Seleccionar

Paso 2: Abrimos la terminal de MacOS (ver figura 1)

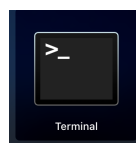


Figure 1: Icono Terminal

Paso 3: Ubicamos la carpeta donde reside nuestro proyecto utilizando **Finder**.

Paso 4: Dentro de la Terminal escribimos el comando **cd**, para cambiarnos al directorio donde se encuentra nuestra aplicacion, esto para facilidad se puede lograr después de escrito el comando haciendo **DragDrop** de la carpeta dentro de **Finder**. (ver figura 2)

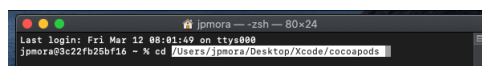


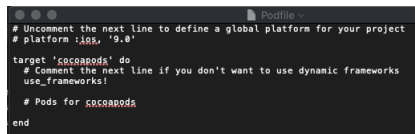
Figure 2: Cambiando de Directorio

Paso 5: Una vez dentro del directorio inicializamos **Cocoapods** dentro de nuestra aplicación utilizando el siguiente comando.

```
pod init
```

Paso 6: Manteniéndonos dentro de la consola editamos el **Podfile** con el comando **Open**, como resultado deberíamos ver algo similar a la figura 3.

Open Podfile



```
Podfile
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'cocoapods' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  # Pods for cocoapods

end
```

Figure 3: Default Podfile

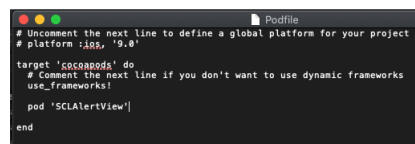
Paso 7: Debemos reemplazar la linea que contiene el comentario

```
# Pods for cocoapods
```

Por la siguiente linea:

```
pod 'SCLAlertView'
```

Como resultado el archivo debe verse similar a 4



```
Podfile
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'cocoapods' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  pod 'SCLAlertView'

end
```

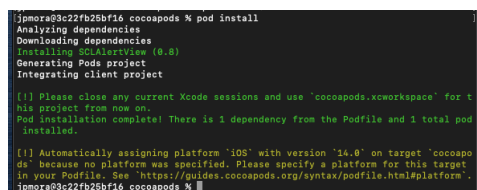
Figure 4: Podfile Final

Paso 8: Guardamos los cambios sobre el archivo Podfile y lo cerramos.

Paso 9: Manteniéndonos dentro de la consola ejecutamos la instalación del **SCLAlertView Pod**, con el siguiente comando:

```
pod install
```

Paso 10: Revisemos los mensajes en consola como buena practica, luego de una ejecución de **pod install**. En nuestro caso deberían existir similares a la figura 5



```
jpmora03c22fb25bf16 cocoapods % pod install
Analyzing dependencies
Downloading dependencies
Installing SCLAlertView (0.8)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use 'cocoapods.xcworkspace' for this project from now on.
Pod installation complete! There is 1 dependency from the Podfile and 1 total pod installed.

[!] Automatically assigning platform 'ios' with version '14.0' on target 'cocoapods' because no platform was specified. Please specify a platform for this target in your Podfile. See 'https://guides.cocoapods.org/syntax/podfile.html#platform'.
jpmora03c22fb25bf16 cocoapods %
```

Figure 5: Mensajes de salida luego de instalación

Paso 11: Revisemos la versión de la biblioteca que se instalo, debería ser '0.8' o superior ya que no establecimos explícitamente el valor dentro de nuestro **Podfile**. En la figura 5 es el primer mensaje color verde debería ser similar a:

```
Installing SCLAlertView (0.8)
```

Paso 12: Revisemos la versión de la plataforma en donde han sido compiladas las bibliotecas, debería ser '14.0' o superior ya que no establecimos explícitamente el valor dentro de nuestro **Podfile**. En la figura 5 es el ultimo mensaje color amarillo debería ser similar a:

[!] Automatically assigning platform 'iOS' with version '14.0' on target 'cocoapods' because no platfor

Paso 13: Debemos en este momento cambiar la forma en que usualmente abrimos nuestro proyecto para poder tener un mejor manejo de las bibliotecas. Y esto esta explícitamente advertido en el segundo mensaje color verde presenten en la figura 5.

[!] Please close any current Xcode sessions and use 'cocoapods.xcworkspace' for this project from now o

Paso 14: Debemos cerrar **Xcode**. Y abrir desde **Finder**, dentro de la carpeta de nuestro proyecto el archivo **cocoapods.xcworkspace** (ver figura 6). También podemos observar un par de carpetas relacionadas con **Cocoapods**.

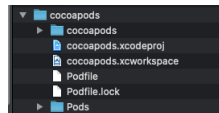


Figure 6: Carpeta del proyecto

Paso 15: Una vez damos doble click sobre el archivo **cocoapods.xcworkspace**, deberíamos tener en pantalla una versión de **Xcode** similar a la figura 7.

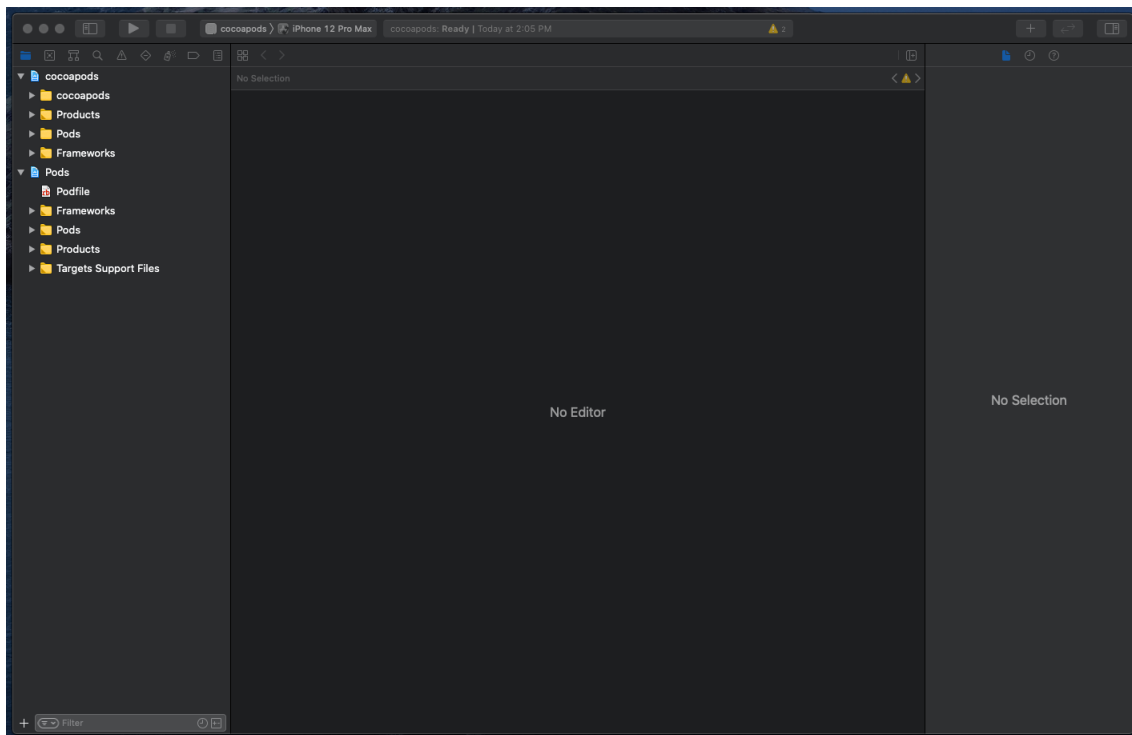


Figure 7: Xcode

Paso 16: Abrimos el archivo **ContentView.swift**.

Paso 17: Compilamos y ejecutamos la aplicación. Esto de manera usual, ya que debemos comprobar que todo siga compilando bien antes de agregar las referencias de biblioteca dentro de nuestra aplicación.

Paso 18: Dentro del archivo **ContentView.swift**, agregamos un nuevo **import**.

```
import SCLAlertView
```

Paso 19: Reemplazar el código de **Text("Hello World")**, con el siguiente código:

```
VStack{
    Text("Hello, world!")
    Button(
```

```

        action:{
            self.showAlert()
        },
        label: {
            Text("Click Me!")
        }
    )
}

```

Paso 20: A este punto el **View** carece de un método **showAlert**, así que debemos agregar el siguiente código para corregirlo:

```

func showAlert(){
    let appearance = SCLAlertView.SCLAppearance(
        showCloseButton: false
    )
    let alertView = SCLAlertView(appearance: appearance)
    alertView.showSuccess(
        "No button", subTitle: "You will have hard times trying to close me"
    )
}

```

Paso 21: Como resultado deberíamos tener un código similar al presente en la figura 8.

Paso 22: Compilamos y ejecutamos. Una vez presionamos el botón de **"Click Me!"**, deberíamos ver un **Pop-up** similar a la figura 9.

Paso 23: Nuevamente en el código donde aparece el parámetro **showCloseButton** como **false** lo podemos reemplazar por **true**

Paso 24: Compilamos y ejecutamos. Una vez presionamos el botón de **"Click Me!"**, deberíamos ver un **Pop-up** similar a la figura 10, el cual ahora tiene un botón **Done**.

Para concluir se presentó un ejemplos de la documentación de la biblioteca, en caso de querer realizar una investigación de mayor profundidad puede consultar el siguiente enlace <https://github.com/vikmeup/SCLAlertView-Swift>

```

8 import SwiftUI
9 import SCLAlertView
10
11 struct ContentView: View {
12     var body: some View {
13         VStack{
14             Text("Hello, world!")
15             Button(
16                 action:{
17                     self.showAlert()
18                 },
19                 label: {
20                     Text("Click Me!")
21                 }
22             )
23         }
24     }
25
26     func showAlert(){
27         let appearance = SCLAlertView.SCLAppearance(
28             showCloseButton: false
29         )
30         let alertView = SCLAlertView(appearance: appearance)
31         alertView.showSuccess(
32             "No button", subTitle: "You will have hard times trying to close me"
33         )
34     }
35 }
36
37 struct ContentView_Previews: PreviewProvider {
38     static var previews: some View {
39         ContentView()
40     }
41 }
42

```

Figure 8: Código integrando SCLAlertView

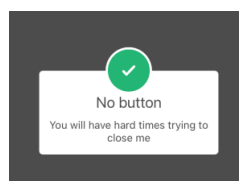


Figure 9: Aplicación mostrando un SCLAlertView Success

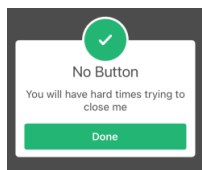


Figure 10: Aplicación mostrando un SCLAlertView Success con botón de Done