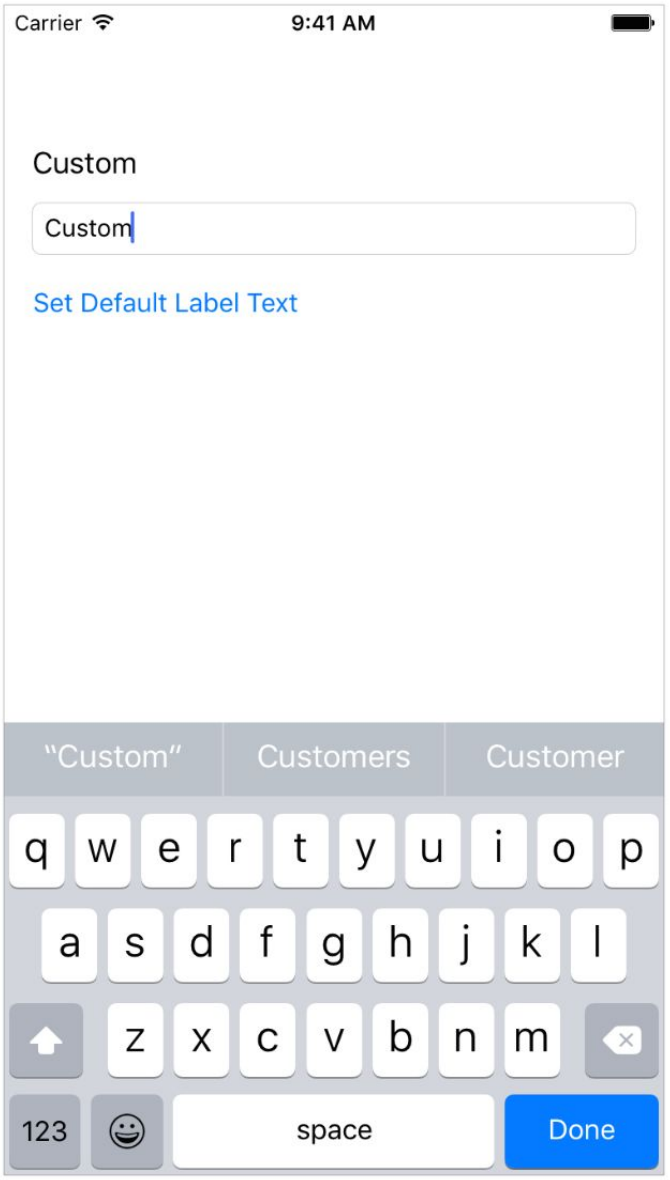
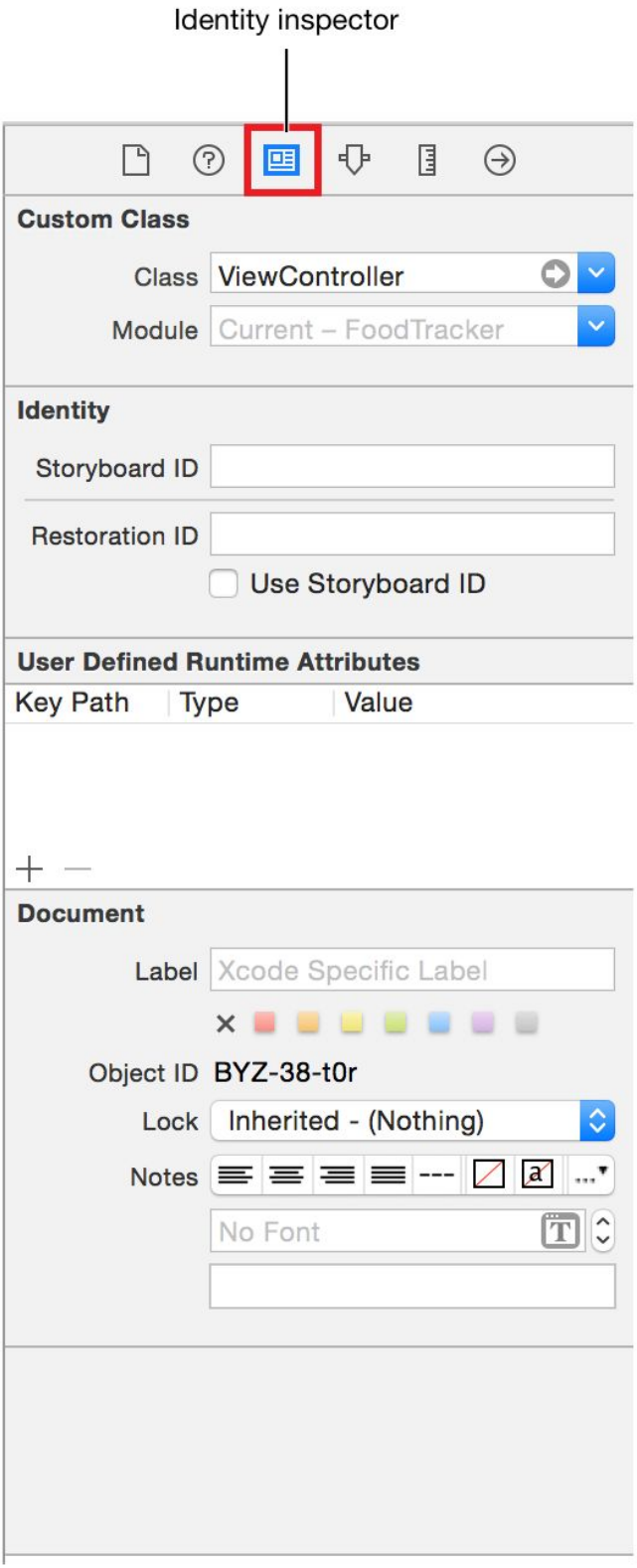
Lab - 04

Objetivos

- Conectar la UI con el Código
- Conocer IBOUTLETs, IBActions
- Escuchar eventos de teclado en el dispositivo

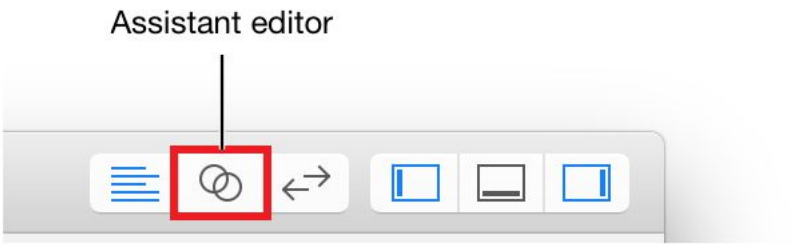


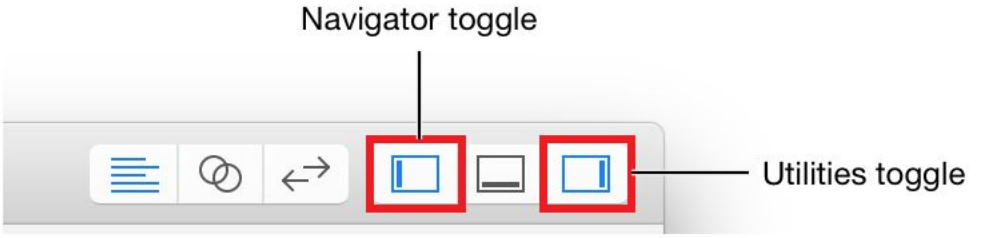
Identity Inspector:
Esta opción permite cambiar la class que controla la Vista como tal



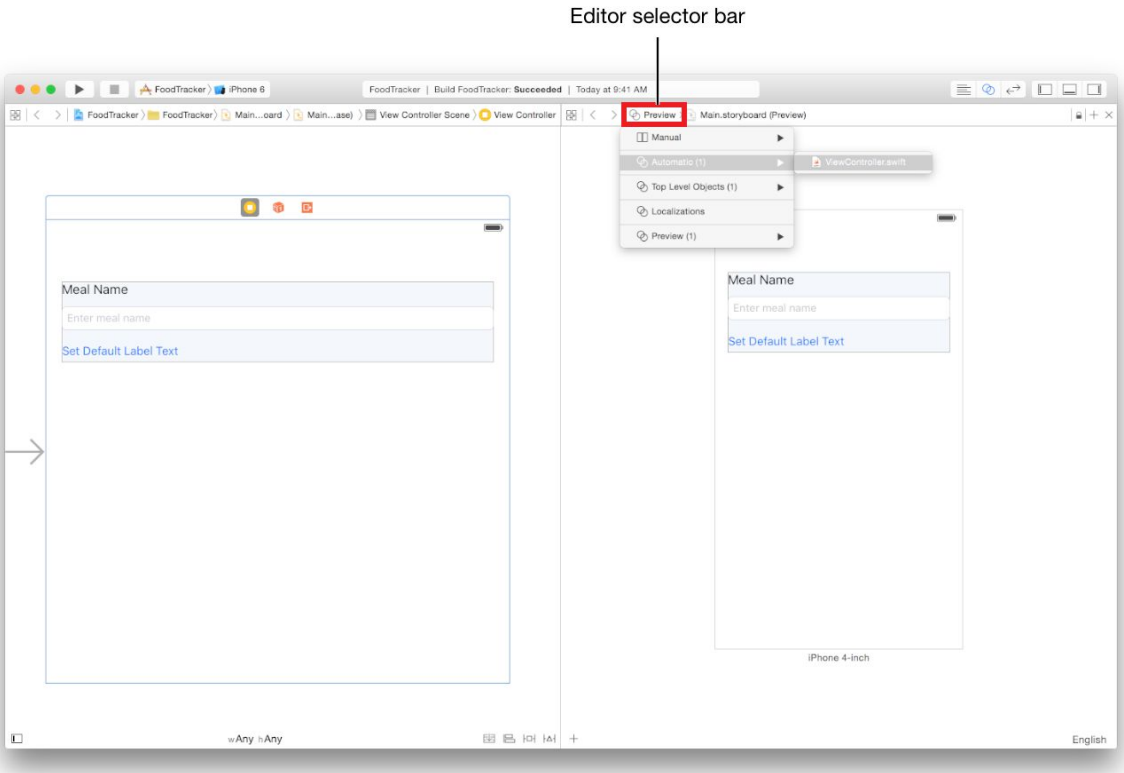
Conectar UI con el Codigo

1. Abrir story board, Main.storyboard.

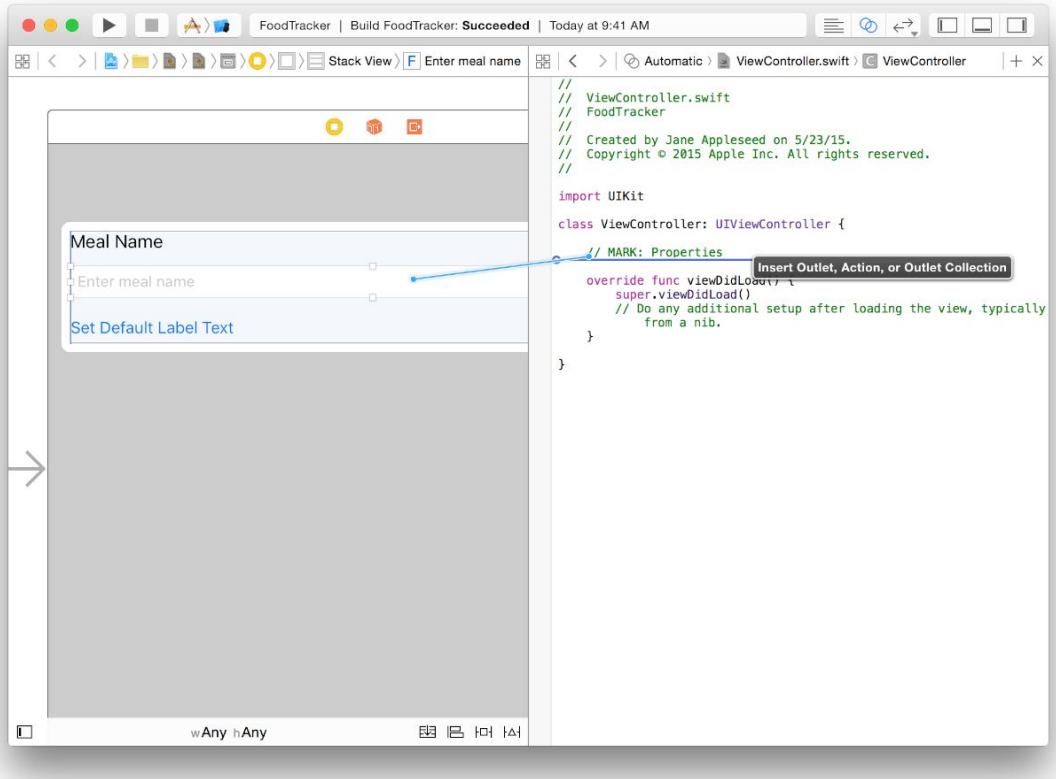
2. Click en assistant editor.
3. 

4. Mostrar los siguientes tool bars:
5. 

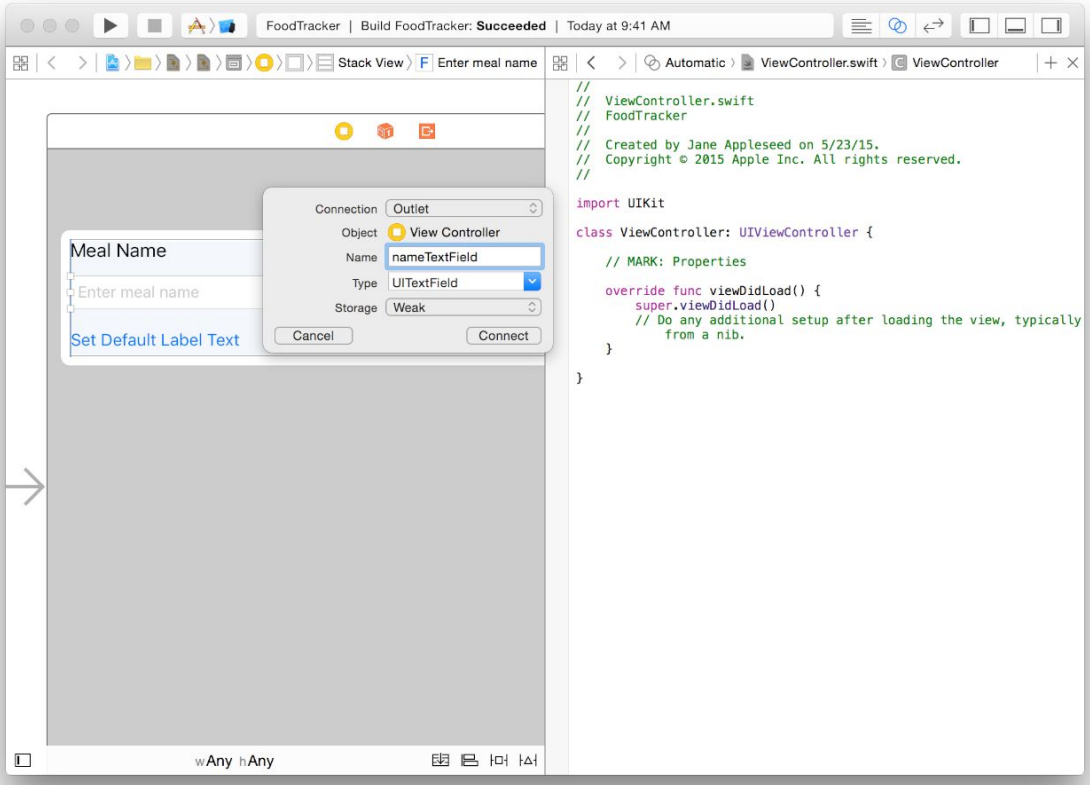
.
6. En la barra de seleccion editor, que aparece en la parte superior del editor asistente, cambiar el editor asistente de vista previa para automático> ViewController.swift.



7.
8. ViewController.swift se muestra a la izquierda del editor
9. En su story board seleccionar el TextField
10. Presione la tecla control y click izquierdo y arrastre hacia la clase ViewController.swift.



11.
12. En el diálogo que aparece poner el nombre de: `nameTextFieldb` .
13. El diálogo debería mostrarse de la siguiente manera:



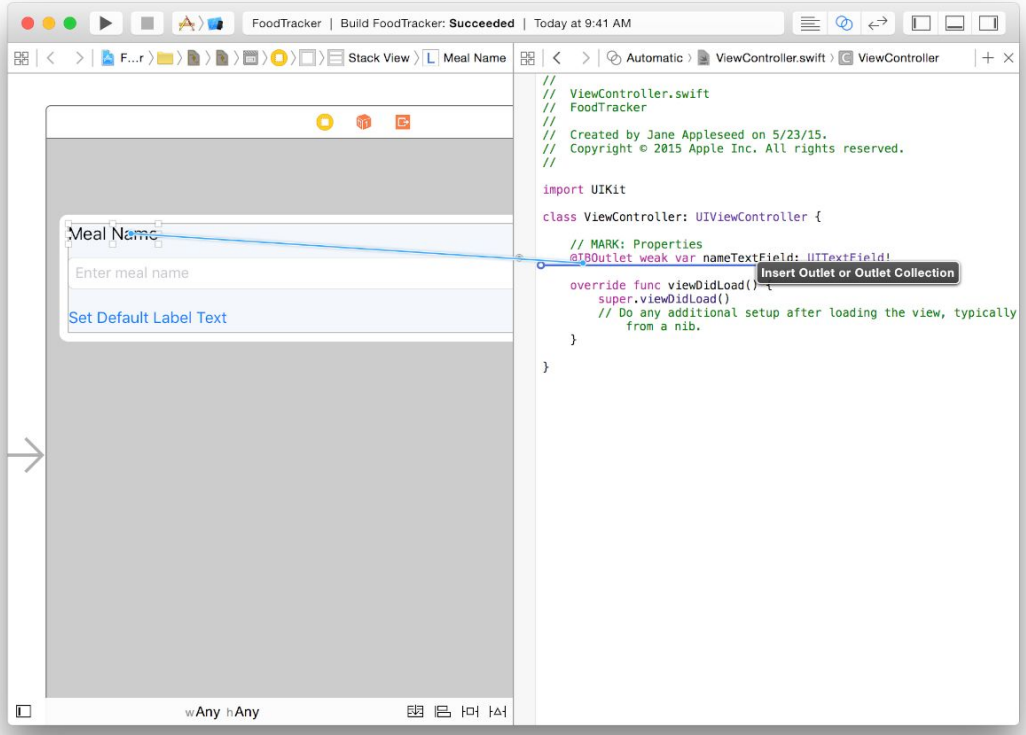
14.
15. Click Connect.
16. Xcode añade el código necesario para ViewController.swift para almacenar un puntero al campo de texto y configura el guión gráfico para establecer esa conexión.

```
@IBOutlet weak var nameTextField: UITextField!
```

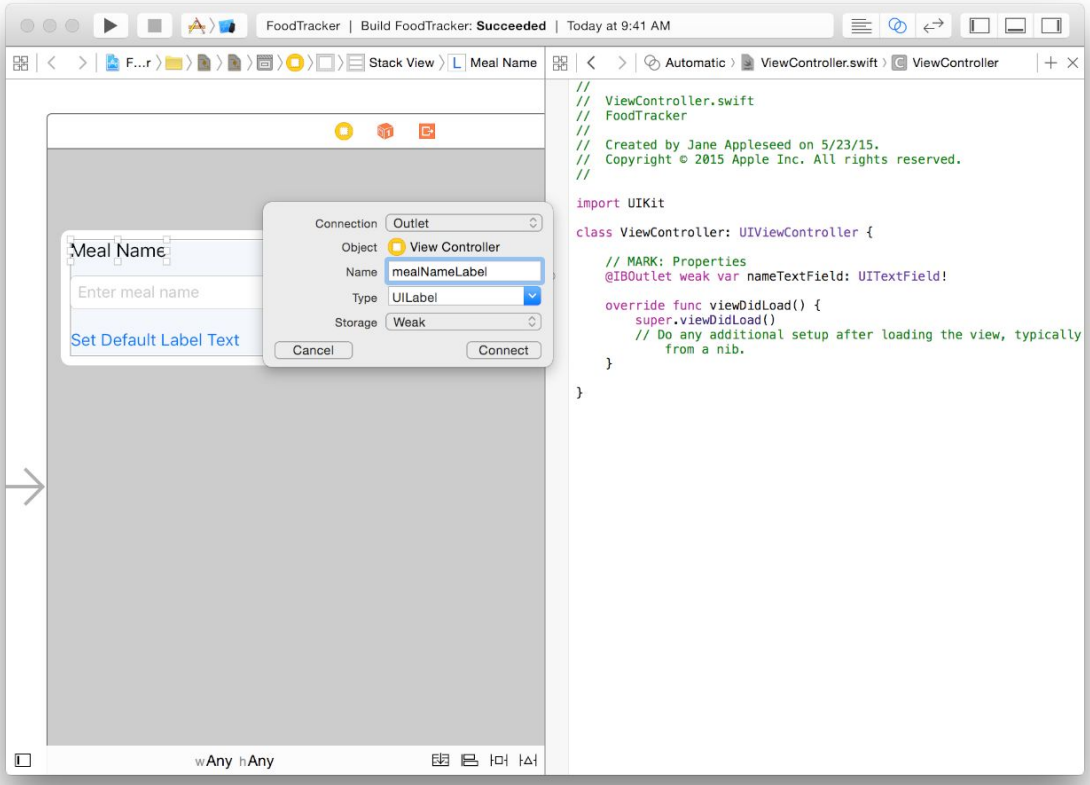
17.

Para conectar la etiqueta con el código de ViewController.swift

1. In su story board seleccione el Label
2. Presione la tecla control y click izquierdo y arrastre hacia la clase ViewController.swift.
3.



4.
5. En el dialogo que se muestra ingrese el text `mealNameLabel`.
6. Al final el dialogo debería mostrarse de la siguiente manera:



7.
8. Click Connect.

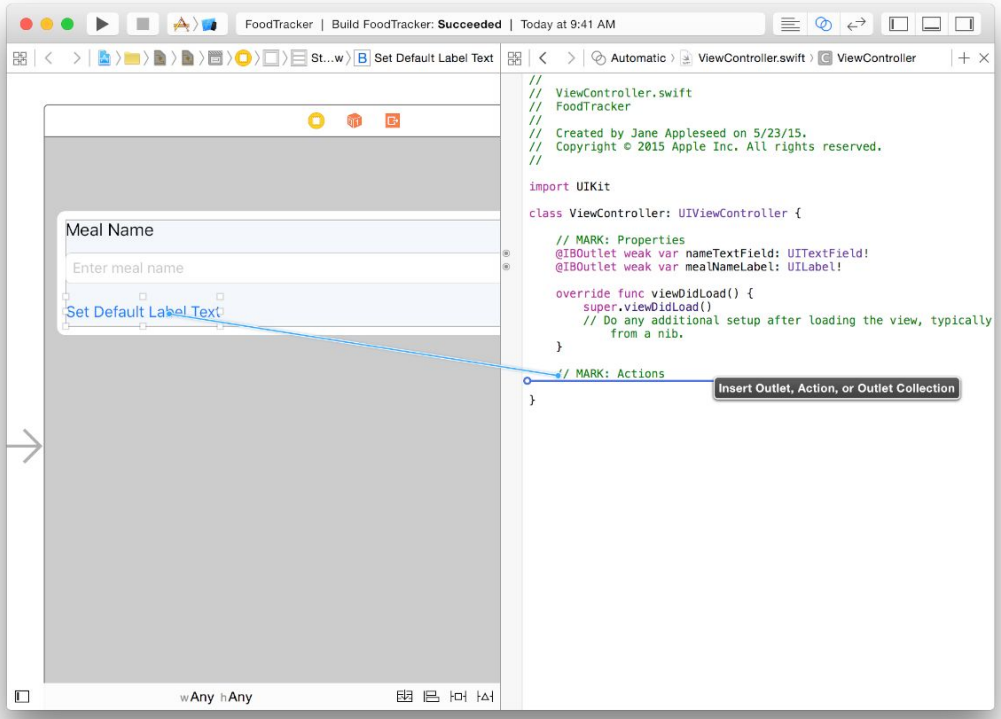
Definir an Action to Perform

Las aplicaciones de iOS se basan en la programación orientada a eventos. Es decir, el flujo de la aplicación se determina por eventos: eventos del sistema y las acciones del usuario. El usuario lleva a cabo acciones en la interfaz que desencadenan eventos en la aplicación. Estos acontecimientos dan lugar a la ejecución de la lógica y la manipulación de los datos de la aplicación.

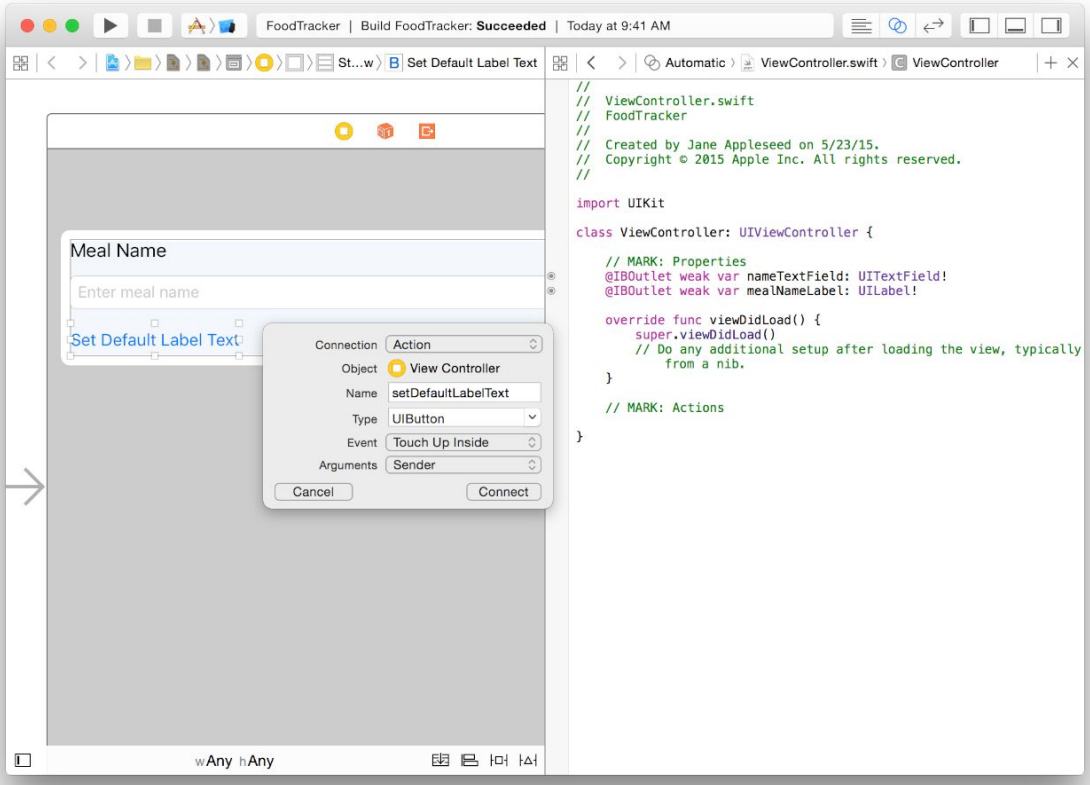
Crear una action del Botón dentro del ViewController.swift code

1. En ViewController.swift, antes de la última llave de cierre (}), añada lo siguiente:
2. // Nota: acciones
3. Este comentario indica que esta es la sección del código que lista las acciones.

4. En el guión gráfico, seleccione el botón de texto de etiqueta predeterminada Set.
5. Control y arrastre desde el botón de etiquetas de texto Establecer predeterminado en el lienzo de la pantalla de código en el editor de la derecha, detener el arrastre en la línea debajo del comentario que acaba de añadir en ViewController.swift.
6.



7.
8. En el cuadro de diálogo que aparece, por conexión, seleccione Acción.
9. En Nombre, Tipo **setDefaultLabelText**.
10. En Tipo, seleccione UIButton.
11. Usted puede haber notado que el valor de los valores predeterminados de campo Tipo a AnyObject. En Swift, AnyObject es un tipo utilizado para describir un objeto que puede pertenecer a cualquier clase. Especificando el tipo de este método de acción para ser UIButton significa que sólo los objetos de botón pueden conectarse a esta acción. Aunque esto no es significativo para la acción que se está creando en este momento, es importante recordar para más adelante.
12. Deje el resto de las opciones como están. Su diálogo debería tener este aspecto:



13.
14. Click Connect.

Xcode añade el código necesario para ViewController.swift para configurar el método de acción.

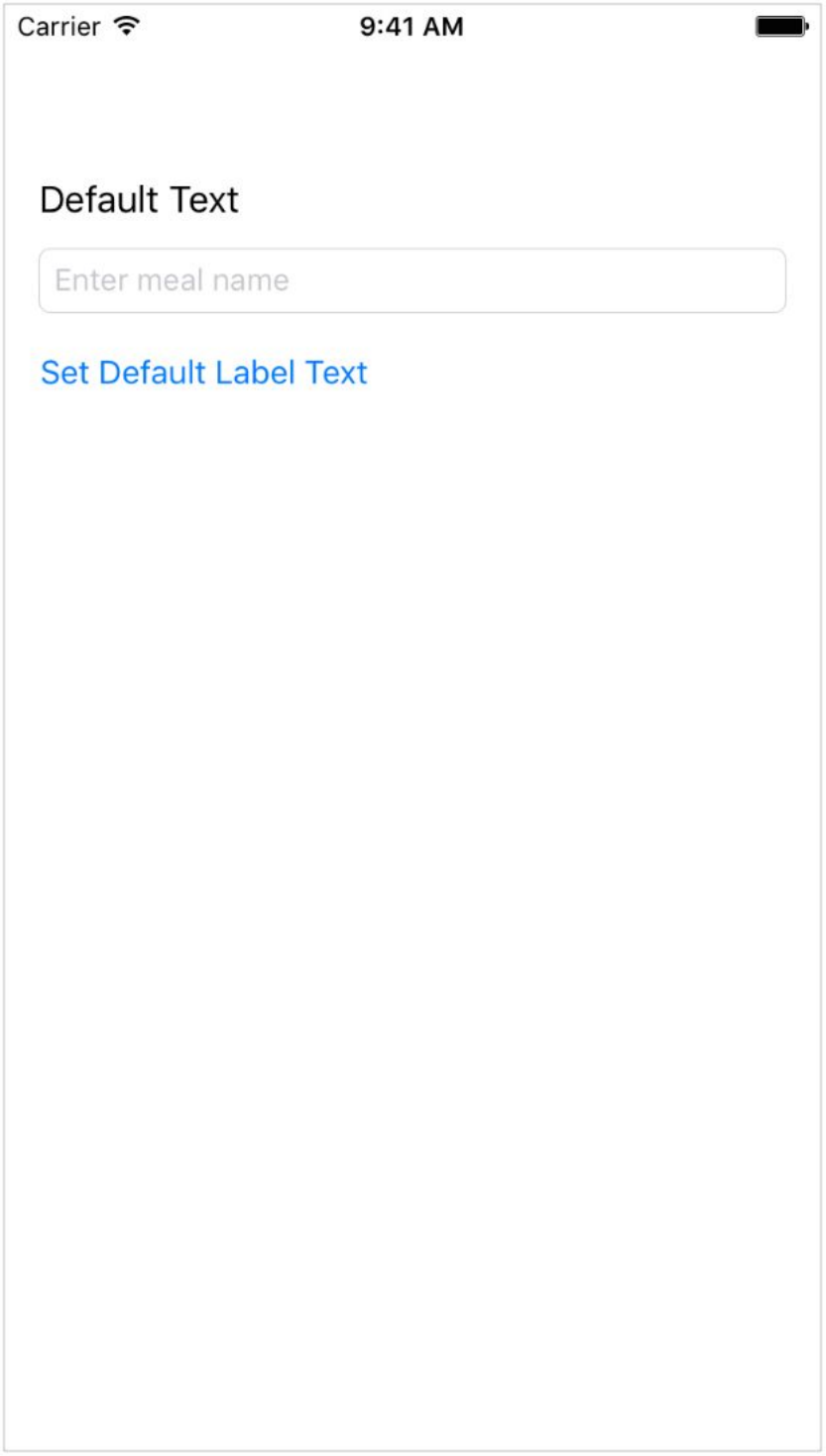
```
@IBAction func setDefaultLabelText(sender: UIButton) {

}
```

Para cambiar el Texto del label debemos cambiar el atributo text.

```
@IBAction func setDefaultLabelText(sender: UIButton) {
  mealNameLabel.text = "Default Text"
}
```

Ejecutar la Aplicación y probar que el label cambia de texto después de presionar el Boton



Process User Input

Para procesar entradas de texto, el componente TextFiled tiene la posibilidad de delegar el evento entrada “Input text” a cualquier controlador que quiere escuchar esos eventos eso gracias a los Delegados “Patron Delegate”

Para adoptar el delegado del text field siga los siguientes pasos

1. En ViewController.swift:

```
class ViewController: UIViewController {
```

2. Agregue la clase UITextFieldDelegate para adoptar el protocolo.

```
class ViewController: UIViewController, UITextFieldDelegate {
```

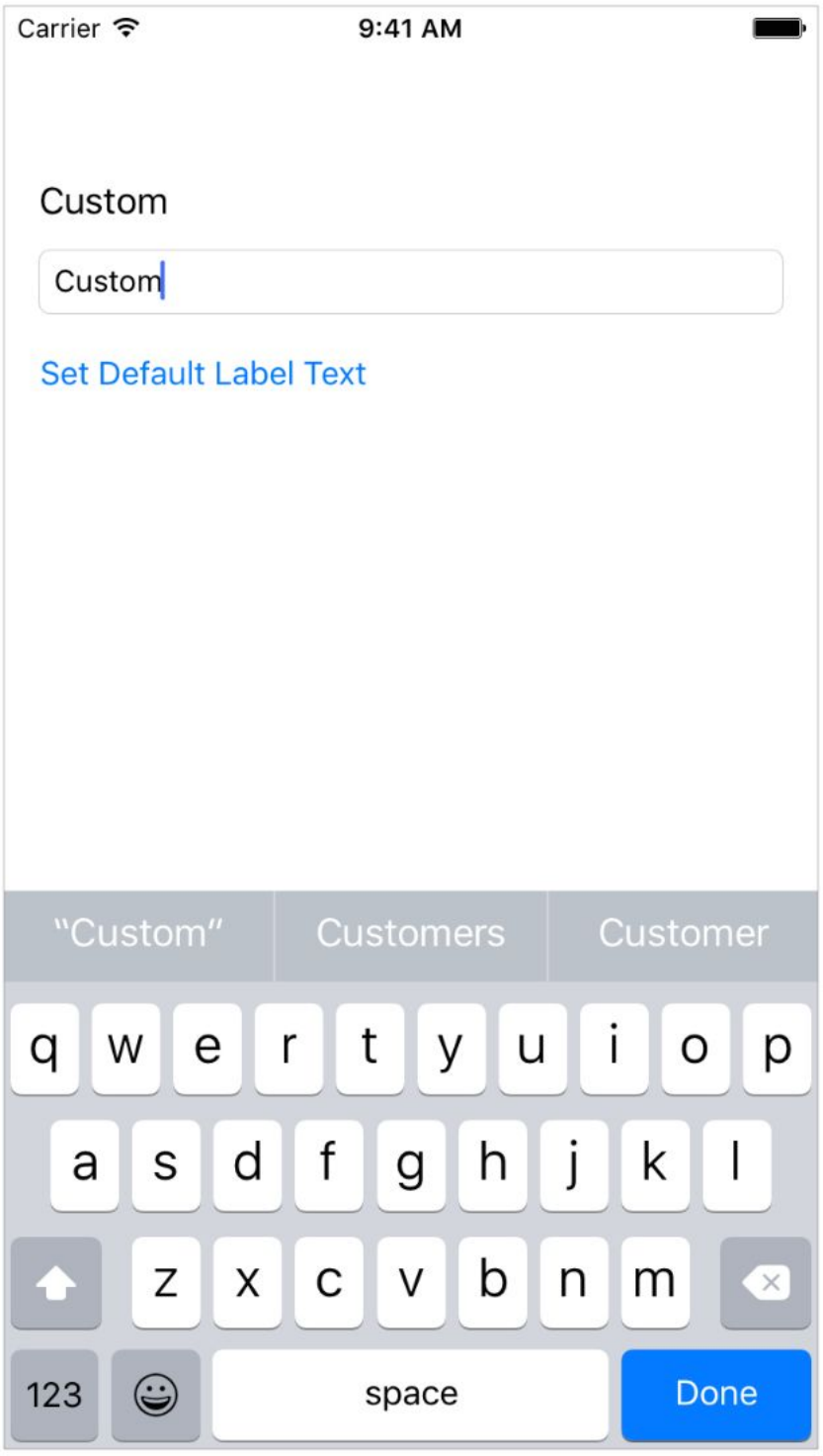
3. Direccionar como delegado al controlador de la UI en el metodo viewDidLoad()

```
override func viewDidLoad() {
    super.viewDidLoad()
    nameTextField.delegate = self

    // Do any additional setup after loading the view, typically from a nib.
}
```

4.- Implementar el method “textFieldShouldReturn” como parte del protocolo que es un optional protocol

```
func textFieldShouldReturn(textField: UITextField) -> Bool {
    // Hide the keyboard.
    textField.resignFirstResponder()
    return true
}
func textFieldDidEndEditing (textField: UITextField) {
// change the content
    mealNameLabel.text = textField.text
}
```


Carrier
9:41 AM
Custom
Custom
Set Default Label Text
"Custom"
Customers
Customer
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
space
Done