



Asignatura: Diseño de interfaces y principios básicos de UI/UX

Guía de aprendizaje N.º 2
20 de Julio de 2024

Managua, 20 de Julio de 2024

Diseño de interfaces y principios básicos de UI/UX

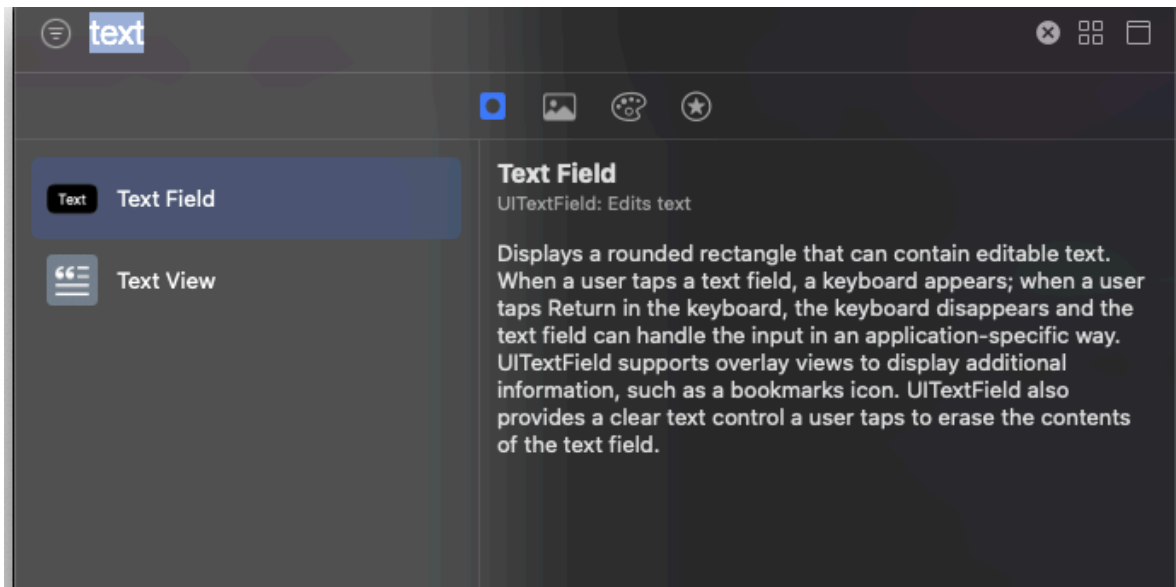
I. CONTENIDOS

Introducción:

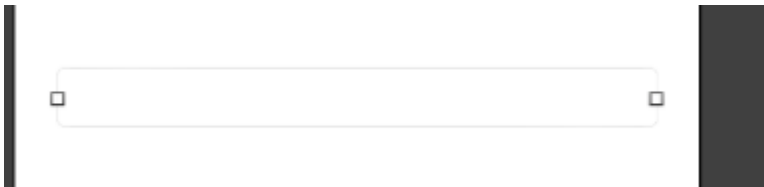
En este documento se creará una serie de procedimientos relacionados a los componentes visuales del toolbox de xcode (textfield, switch, segmentedControl, stepper), el objetivo es poder comprender las funciones básicas de los componentes más usados en las aplicaciones.

Este documento es una continuación de los componentes visuales de xcode, por lo tanto puedes ocupar el mismo proyecto que empezamos, o crear uno nuevo.

Abrimos la caja de herramienta de xcode, pero en este caso vamos a ingresar un textfield y un switch.



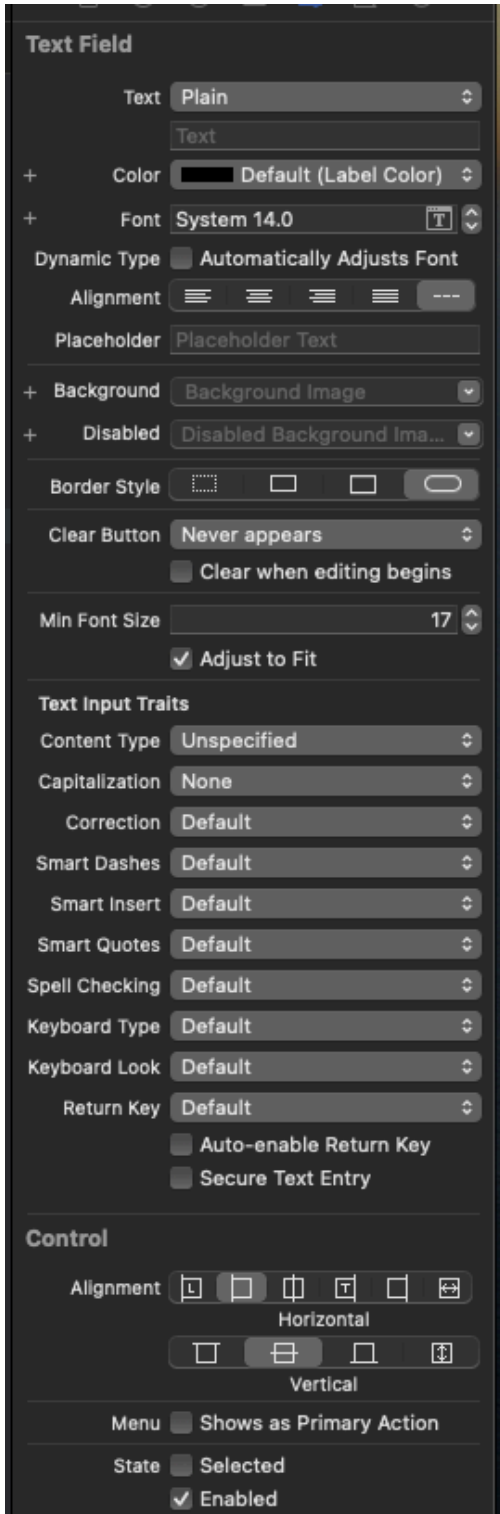
Para insertar Componentes recuerda que lo único que hay que hacer es buscar el componente arrastrar y soltarlo en nuestra interfaz.



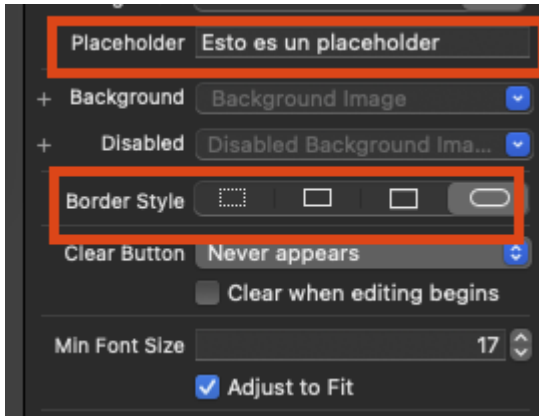
Listo, ya hemos agregado nuestro **textfield** a nuestra interfaz.

Al igual que el componente Label como el button, tenemos a nuestro panel derecho, información que nos ayuda a personalizar nuestro textfield, en este caso añadiremos un

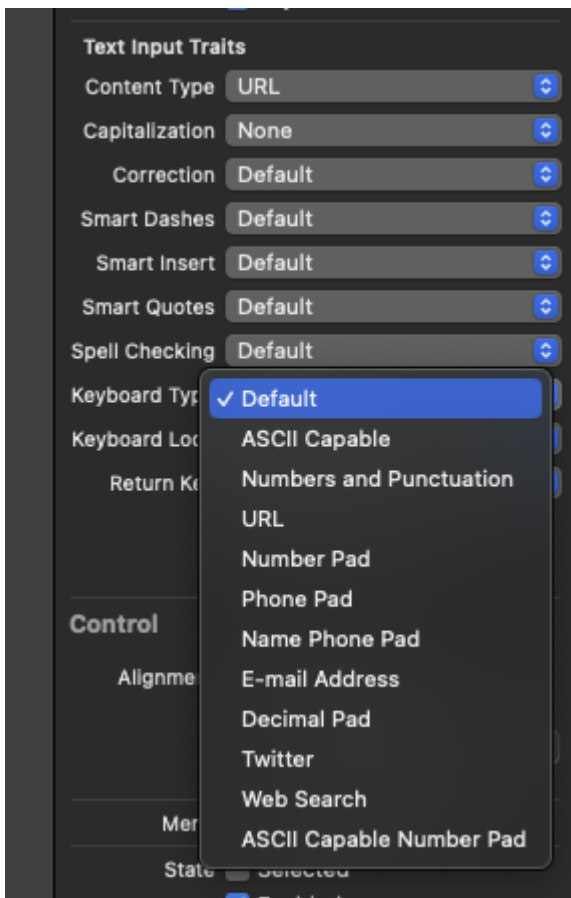
placeholder y algo de estilo como borde.



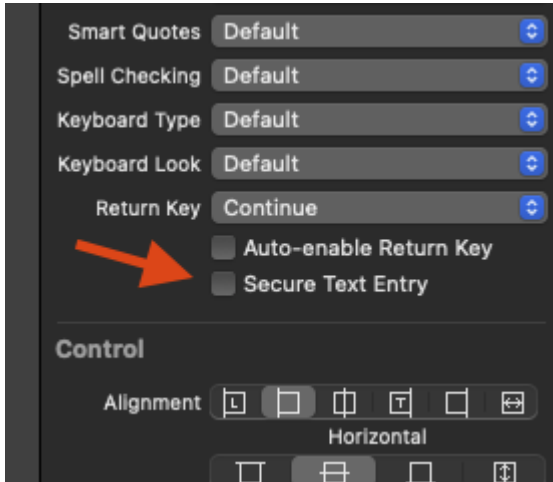
En cuanto a nuestro placeholder podemos agregar un texto cualquiera, y podemos igual también jugar con el borde.



El textfield, tiene unos aspectos muy importante, por ejemplo, podemos modificar el tipo de teclado que queremos que nos salga



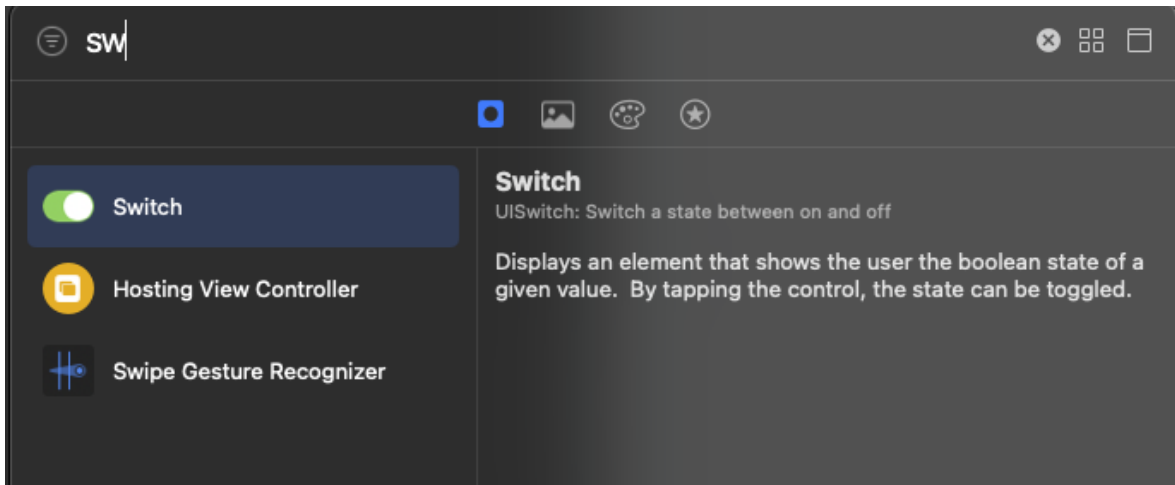
o si estamos implementando algún tipo de textfield para contraseña solo debemos activar esta opción.



y el resultado será el siguiente.



Ahora ingresemos un componente switch a nuestra interfaz.



Lo siguiente a hacer es agregar el outlet de nuestro switch a nuestra interfaz visual, también conecta el outlet de nuestro textfield.

```
14
15
16  @IBOutlet var myTextField: UITextField!
17  @IBOutlet var mySwitch: UISwitch!
18
```

vamos ahora a jugar con nuestro switch, cambiando el color, y desactivandolo por defecto, para eso, vamos debajo de nuestro viewDidLoad y agregamos el siguiente código.

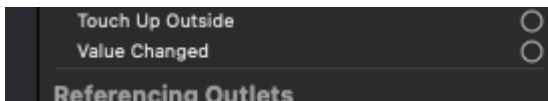
```
//switch  
  
mySwitch.onTintColor = .purple  
mySwitch.isOn = false|
```

Ejecuta la aplicación para observar los cambios.

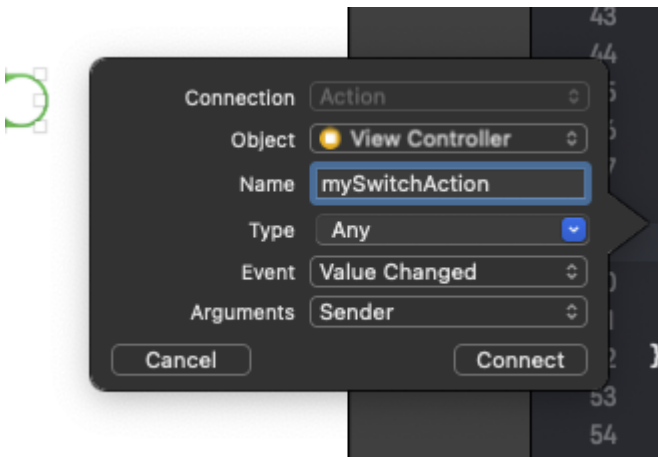
Como observamos nuestro switch por defecto está apagado, y cuando lo encendemos, el color ya no es verde, ahora es purple.



Ahora hagamos un poco de lógica, vamos a realizar una acción a nuestro switch, pero observa bien, como el switch mantiene dos estados, la acción no es igual a un botón el cual le indicamos con un **Touch Up Inside**, en el caso de nuestro switch debemos utilizar **Value Changed**.



Igual que hicimos el botón, en el círculo arrastramos hasta nuestro viewController.



y lo conectamos con el botón Connect.

```
47
48
49  @IBAction func mySwitchAction(_ sender: Any) {
50
51
52  }
53
54
```

Lo que quiero hacer es que cuando mi switch esté activado, este deje inhabilitado mi textfield, y cuando lo apague mi textfield logre habilitarse.

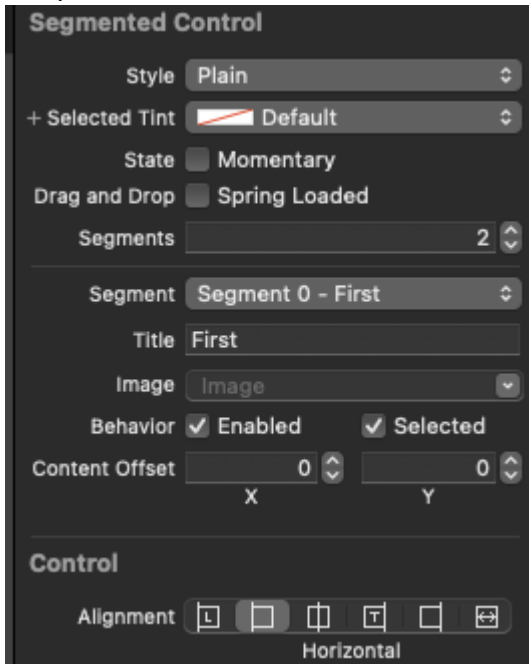
```
48
49  @IBAction func mySwitchAction(_ sender: Any) {
50
51      myTextField.isEnabled = mySwitch.isOn
52
53  }
54
```

Corre la aplicación y observa los cambios, pero hay un pequeño detalle, es momento de poner a prueba algo de lógica, intenta reparar tu mismo el pequeño problema presentado.

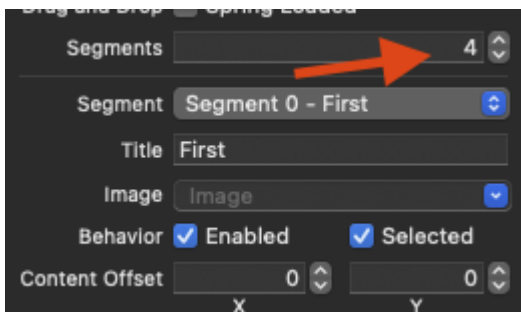
Es hora de aprender otro componente visual el **Segmented Control** lo añadimos desde nuestra caja de herramientas, buscamos segmented control y lo añadimos a nuestra interfaz, de modo que quede de la siguiente manera.



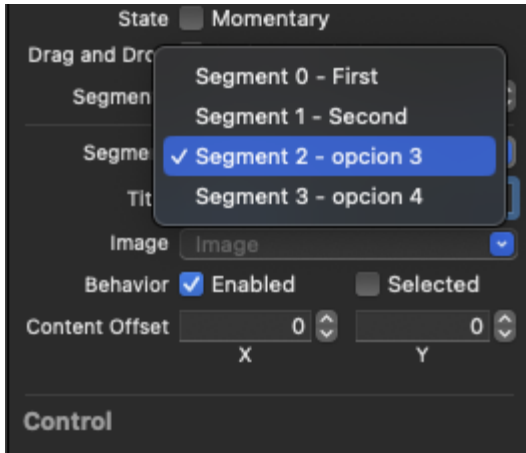
Igualmente vamos a explorar nuestra caja específica que contiene información de este componente.



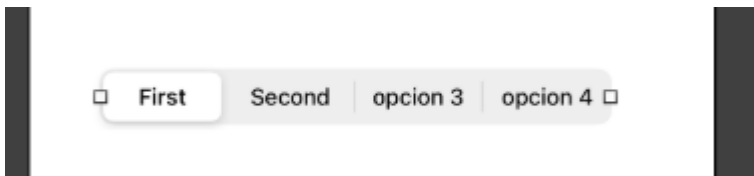
Observemos lo siguiente, nuestro control nos presenta dos opciones, pero nosotros podemos agregar más, ingresé 4 opciones en la parte donde dice **Segments**.



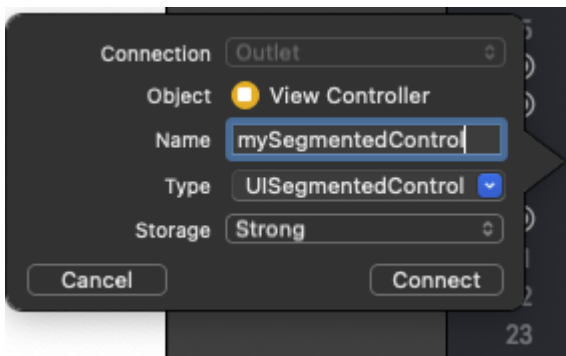
Ahora en la opciones donde dice segment observa que empieza desde el segmento cero y tiene un título, podemos agregar a nuestro segmento 2 y 3 los títulos que quieras, en este caso pondré opción 3 y opciones 4.



Recuerda estirar un poco más el componente para ver las opciones añadidas.



Vamos hacer ahora algo de código, ingresa el outlet de nuestro segmented control a nuestro viewController.



Lo conectamos y ahora vamos a añadir algo de logica, crear un arreglo de string llamado myStringValues e ingresa opciones como Uno, dos, tres , cuatro..

```
private let myStringValues = ["Uno", "Dos", "Tres", "Cuatro", "Cinco"]
```

Dentro de nuestro viewDidLoad vamos a agregar el siguiente código.

```
// segmented control
mySegmentedControl.removeAllSegments()
for (index, value) in myStringValues.enumerated() {
    mySegmentedControl.insertSegment(withTitle: value, at: index, animated:
        true)
}
```

Cómo logras apreciar, lo primero que he hecho es eliminar todas las opciones de nuestro **segmentedControl**, ahora con un ciclo for y nuestro arreglo anteriormente creado, he hecho un recorrido por cada item y lo he asignado a nuestro **segmentedControl**, corre la aplicación y veras como ahora tiene nuevos elementos.



Ahora ingresemos un stepper, igual abre la caja de herramientas y busca el componente stepper, agregalo a nuestra interfaz visual, y este lo vamos a acompañar con un Label, quedando de la siguiente manera



conecta sus outlet a nuestro viewController

```
22
23
24 @IBOutlet var myStepperControl: UIStepper!
25 @IBOutlet var myLabelStepper: UILabel!
26
27
```

ahora vamos a agregar una funcionalidad a nuestro stepper, ya que son dos posibles opciones en la parte de eventos siempre trabajaremos con Value Changed al igual que lo hicimos con nuestro switch.

```
74
75  @IBAction func myStepperAction(_ sender: UIStepper) {
76
77      |
78  }
```

y agregamos el siguiente código

```
74
75  @IBAction func myStepperAction(_ sender: UIStepper) {
76
77      let value = myStepperControl.value
78
79      myLabelStepper.text = "\(value)"
80  }
81
```

Ejecuta nuevamente la aplicación y verás el funcionamiento de este componente visual.



Ejercicios

Descripción: Los estudiantes realizarán una serie de ejercicios prácticos que cubrirán todos los temas tratados en la clase. Los ejercicios permitirán a los estudiantes poner en práctica lo aprendido y recibir retroalimentación inmediata sobre su comprensión.

II. LOGROS DE APRENDIZAJES

Al finalizar el encuentro, los estudiantes serán capaces de:

- Conocer los componentes visuales del toolbox de xcode switch, Textfield
- Poder implementar componentes visuales, detallando colores y funcionamiento.
- Identificar la caja de herramientas, storyboard y menú de implementación

III. ORIENTACIONES METODOLÓGICAS

Para alcanzar los logros de aprendizaje, luego de la evaluación trabajaremos con las siguientes estrategias y recursos:



Actividad de Aprendizaje No.2

Modalidad: Individual

Tiempo aproximado: 20 minutos

Periodo de realización: 20 de Julio, en el aula de clases

Puntaje: 10 pts

Descripción de la actividad: Es momento de poner en práctica tu lógica y razonamiento, busca en la caja de herramientas el componente llamado `UIActivityIndicatorView`, observa su caja de herramientas, intenta cambiar color.. conecta con el `viewController`, lo que debes hacer es que cuando el switch este encendido, que el `activityIndicator` haga una función (`startAnimating()`) y cuando el switch esté apagado la otra función(`stopAnimating()`).