



POLITÉCNICA

ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS

## Entrada de Texto

IWEB 2019-2020  
Santiago Pavón

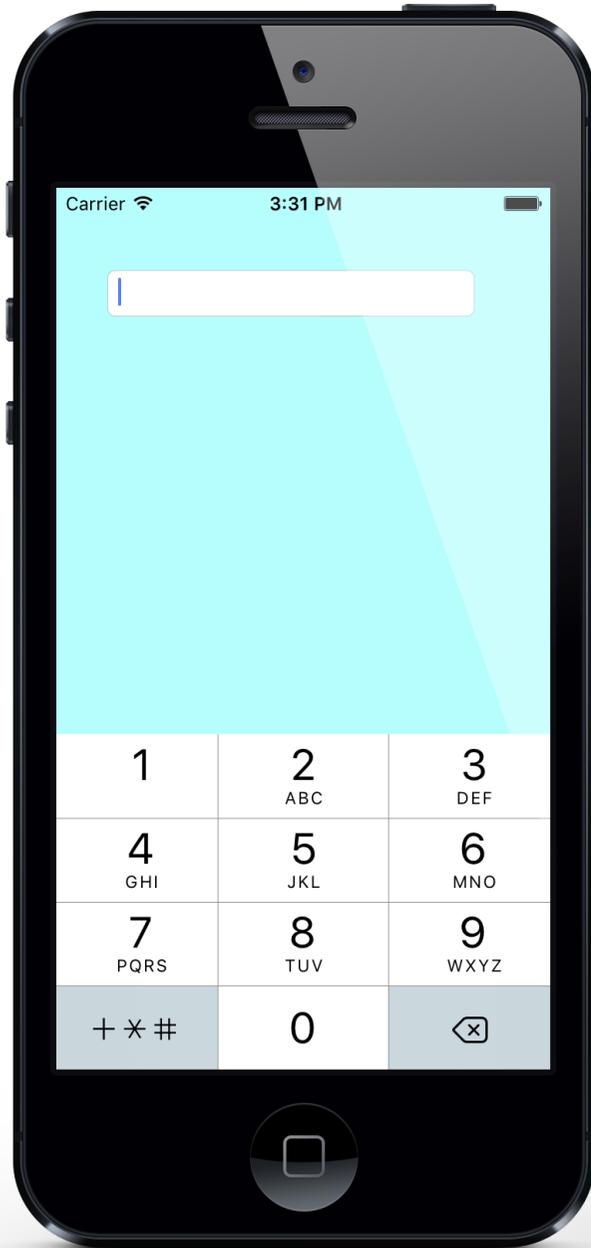
ver: 2019.11.12

# UITextField y UITextView

- Las views de entrada de texto son:
  - **UITextField** - editar una línea de texto.
  - **UITextView** - editar múltiples líneas, permite scroll.
- Cuando una view de entrada texto se convierte en el **primer responder**, el teclado aparece automáticamente.
  - La view se convierte en el primer responder cuando:
    - El usuario toca con el dedo sobre ella
    - o se ejecuta su método **becomeFirstResponder**.
- El teclado desaparece cuando la view deja de ser el primer responder.
  - Tocando sobre otra view para que se convierta en el primer responder
  - o ejecutando su método **resignFirstResponder**.

# El Teclado

- El teclado ya está hecho, pero podemos configurar muchas características del teclado asignando valores a las propiedades definidas por el protocolo **UITextInputTraits**.
  - ascii, numérico, mail, url, tipo de tecla return, capitalización automática, corrección del texto, seguridad para passwords, etc.
  - Estas propiedades también pueden configurarse desde el interface builder.
  - Este protocolo lo implementan **UITextField** y **UITextView**.
- El teclado también puede ampliarse con controles creados por el usuario.
  - Views accesorias añadidas usando la propiedad **inputAccessoryView** de UITextField o UITextView.
    - Se muestran encima del teclado.
- También podemos crearnos nuestro propio teclado asignándoselo a la propiedad **inputView** de UITextField o UITextView.



# Delegados

- Los controles de entrada de texto pueden tener delegados a los que informar sobre las actividades relacionadas con la edición.
- Los protocolos delegados son: **UITextFieldDelegate** y **UITextViewDelegate**.
- Estos protocolos definen métodos para:
  - Consultar si se permite que comience la edición de un campo
    - textFieldShouldBeginEditing(:)**
    - textViewShouldBeginEditing(:)**
  - Informar de que ha empezado la edición de un campo de texto
    - textFieldDidBeginEditing(:)**
    - textViewDidBeginEditing(:)**
  - Consultar si se permite que termine la edición de un campo
    - textFieldShouldEndEditing(:)**
    - textViewShouldEndEditing(:)**
  - Informar de que ha terminado la edición de un campo de texto
    - Este suele ser el método utilizado para acceder a la propiedad **text** y extraer el texto introducido.
    - textFieldDidEndEditing(:)**
    - textViewDidEndEditing(:)**
  - Consultar al delegado si debe atenderse a la pulsación de la tecla **Return**.
    - textFieldShouldReturn(:)**
  - ...

# Más Propiedades / Métodos

- **UITextField** tiene propiedades para:
  - indicar si se debe borrar el texto existente al comenzar una nueva edición  
**clearsOnBeginEditing**
  - Indicar el tamaño del texto  
**adjustsFontSizeToFitWidth, minimumFontSize**
  - Texto a mostrar cuando el campo está vacío  
**placeholder**
  - Imágenes usadas para la apariencia  
**background, disabledBackground**
  - Overlay views  
**clearButtonMode, leftView, leftViewMode, rightView, rightViewMode**
  - ...
- **UITextView** tiene un método para hacer scroll a un rango del texto  
**scrollRangeToVisible(:)**
- Y ambos tienen también propiedades para indicar colores, fonts, alineamiento de texto, etc.

# Eventos

- Podemos crear acciones para los eventos que genera el Text Field.
  - Did End On Exit
    - Se produce al pulsar Return en el teclado.
  - Editing Changed
  - Editing Did End

# Demo

- Crear un objeto **UITextField** para editar una línea de texto
  - crear un outlet que lo apunte y llamarlo **textField**
- Jugar con las características (traits) del teclado.
- Configurar **ViewController** como el delegado del **Text Field** creado.
  - Con IB, mostrar popup con propiedades del **Text Field** y conectarlo con el **ViewController**.
  - Editar **ViewController** para que sea conforme con **UITextFieldDelegate**.
  - Sobrescribir los métodos del delegado a los que se quiera atender.
    - Por ejemplo, **textFieldDidEndEditing(:)**.
- Crear un objeto **UIButton** que tenga como texto "Done".
  - Para que al pulsar el botón se termine la edición y se oculte el teclado, hay que crear una IBAction para el botón que ejecute:  
**textField.resignFirstResponder()**
- También podemos crear una acción para el evento **Did End On Exit** del **UITextView**.
  - Al pulsar **Return** (o similar) en el teclado, se llama a la acción, termina la edición y se oculta el teclado.
- También podemos crear una acción para el evento **Editing Did End** del **UITextView**.
  - Se llamará a la acción cuando el TextInput pierda el foco.
- Además, en la acción o método que se quiera usar hay que coger el valor del TextField y hacer lo que se quiera con él.
- ...

