

EditText

Permite editar texto en pantalla

Definición en Layout

```
<EditText android:id="@+id/et_1"  
    android:text="@string/texto_1"  
    android:digits="abc 123"  
    android:maxLength="30"  
    android:hint="@string/sugerencia_1"  
    android:hintColor="@android:color/red"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

text valor inicial del texto. Puede ser un literal o la referencia a un recurso @string (preferible)

digits representa los únicos caracteres que serán aceptados por el EditText.

maxLength longitud máxima de texto a introducir

hint texto que aparecerá cuando la caja de texto esté vacía. Actúa a modo de ayuda para el usuario indicándole qué se espera que deba introducir.

hintColor representa el color con el se se mostrará el hint (color)

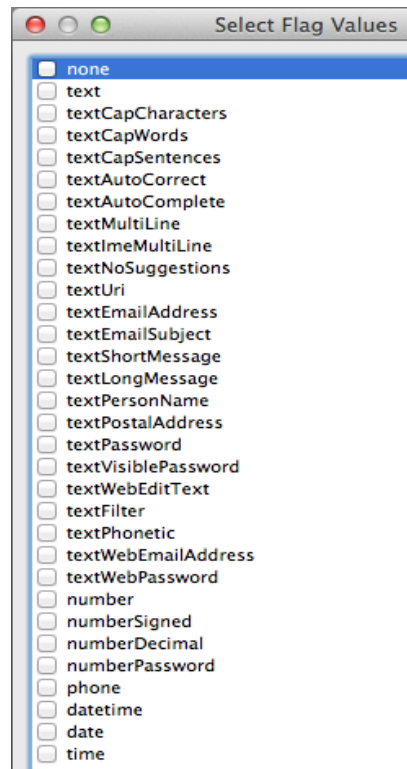
EditText

Permite editar texto en pantalla

Definición en Layout

```
<EditText android:id="@+id/et_1"  
    android:hint="@string/sugerencia_1"  
    android:text="@string/texto_1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="phone"/>
```

inputType representa el tipo de dato que debe introducir el usuario. Este atributo establece indirectamente qué tipo de teclado le aparecerá al usuario. Puede tomar uno o varios valores (separador por el carácter |) de entre la siguiente lista:



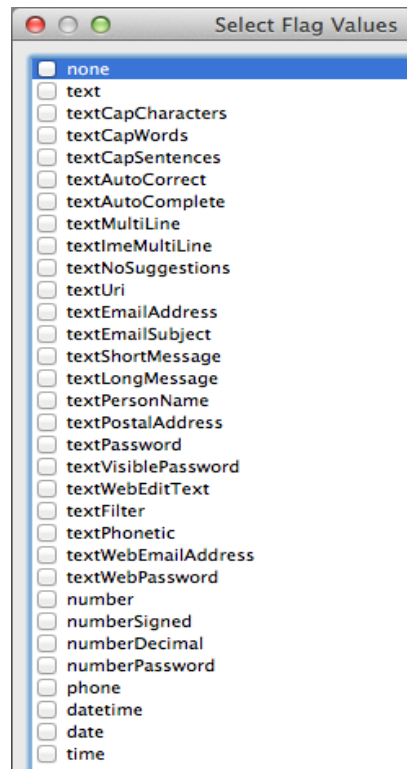
EditText

Permite editar texto en pantalla

Definición en Layout — atributos visuales

```
<EditText android:id="@+id/et_1"  
    android:hint="@string/sugerencia_1"  
    android:text="@string/texto_1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="textMultiLine"/>
```

inputType="textMultiLine" indica que el EditText permite al usuario introducir varias líneas de texto. Si se omite, el EditText sólo permitirá una línea.

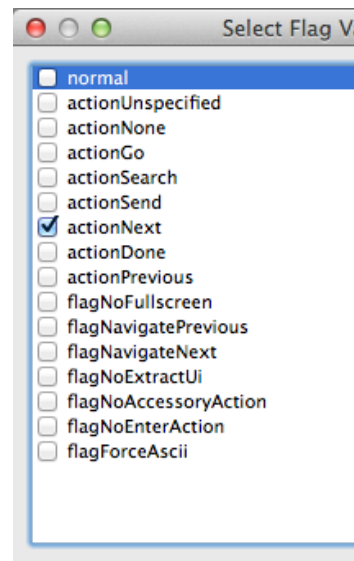
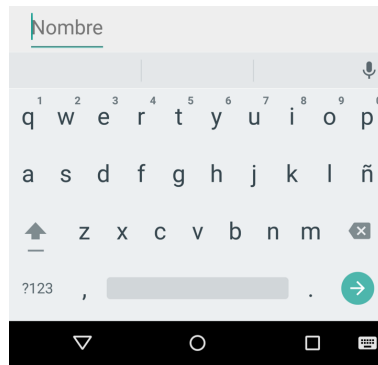


EditText

Permite editar texto en pantalla

Definición en Layout

```
<EditText android:id="@+id/et_1"  
    android:hint="Nombre"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="textPersonName"  
    android:imeOptions="actionGo"/>
```



EditText

Métodos

`setText(text: CharSequence)`

`setText(stringID: Int)`

`getText(): Editable`

Establece y recupera el texto que mostrará la vista. El texto puede indicarse a través de una cadena de caracteres o mediante un entero que represente la identidad de un recurso de tipo String (**R.String.xxx**)

`setError(text: CharSequence)`

`setError(text: CharSequence, icon: Drawable)`

Establece en el EditText, como consecuencia de una validación, un texto que describe el error cometido por el usuario al introducir el valor.

`addTextChangedListener (watcher: TextWatcher)`

Fija un listener que se ejecutará a medida que se modifique el texto durante la edición

```
TextWatcher()
{
    override fun beforeTextChanged(s: CharSequence, start: Int, count: Int, after: Int) : Unit { /* cadena anterior al cambio */ }
    override fun onTextChanged(s: CharSequence, start: Int, before: Int, count : Int): Unit { /* cadena cambiada */ }
    override fun afterTextChanged(editable: Editable): Unit { /* cadena posterior al cambio */ }
};
```

EditText

Métodos

```
addTextChangedListener (beforeTextChanged : (text : CharSequence?, start: Int, before: Int, count: Int) -> Unit = {_,_,_ -> },  
onTextChanged : (text : CharSequence?, start: Int, before: Int, count: Int) -> Unit = {_,_,_ -> },  
afterTextChanged: (text: Editable?) -> Unit)
```

```
editText.addTextChangedListener { text -> : : : }
```

```
doBeforeTextChanged (action: (text : CharSequence?, start: Int, before: Int, count: Int) -> Unit )
```

Fija un listener que se ejecutará a medida que se modifique el texto durante la edición en el momento **BeforeTextChanged**

```
doOnTextChanged (action: (text : CharSequence?, start: Int, before: Int, count: Int) -> Unit )
```

Fija un listener que se ejecutará a medida que se modifique el texto durante la edición en el momento **OnTextChanged**

```
doAfterTextChanged (action: (text: Editable?) -> Unit)
```

Fija un listener que se ejecutará a medida que se modifique el texto durante la edición en el momento **AfterTextChanged**

EditText

Métodos

`setOnEditorActionListener(listener : TextView.OnEditorActionListener) : Unit`

Fija un listener que se ejecutará cuando el usuario pulse el botón Action una vez haya editado el texto.

```
et.setOnEditorActionListener { tv, actionId, event -> // Se oculta el teclado
    val imm: InputMethodManager = tv.context.getSystemService( Context.INPUT_METHOD_SERVICE ) as InputMethodManager
    imm.hideSoftInputFromWindow( tv.windowToken, 0 )

    // Se comprueba qué action se ha pulsado
    return when ( actionId )
    {
        EditorInfo.IME_ACTION_XXXX -> {
            ::::
            true
        }
        else -> false
    }
    // El método devuelve true si se ha realizado alguna acción, de lo contrario false
}
```