

# Programación multimedia y dispositivos móviles

## 3

## Elementos básicos

IES Nervión  
Miguel A. Casado Alías

# TextView

---

- Es una etiqueta de texto, no editable por el usuario
- Tiene numerosas propiedades referentes al aspecto, p.ej:
  - android:textStyle para asignar el estilo (negrita, itálica...)
  - android:textColor para asignar el color del texto

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:text="@string/texto"  
>
```

# Button

---

- Es una subclase de `TextView`
- Desde Android 1.6 se puede determinar qué hace el botón al ser pulsado de la siguiente forma:
  - En la actividad que contiene el botón, definir un método que tome un parámetro *View* y devuelva *void*.
  - En el layout XML, dentro del elemento *Button* , incluir un atributo *android:onClick* con el nombre del método definido anteriormente

```
<Button
```

```
    android:onClick="miMetodo"
```

```
    ...
```

```
/>
```

# ImageView & ImageButton

---

- Sirven para insertar imágenes o imágenes que actúan como botón
- ImageButton es una subclase de ImageView
- `android:src` hará referencia normalmente al recurso *drawable* donde está la imagen

# EditText

---

- Sirve para que el usuario escriba un texto
- Es una subclase de TextView
- Algunas propiedades interesantes:
  - `android:autoText`
  - `android:capitalize`
  - `android:digits` para determinar los caracteres que queremos aceptar, por ejemplo:
    - `android:digits = "0123"`
  - `android:singleLine` para controlar si se permite más de una línea o no

# CheckBox

---

- Tiene dos estados: marcado (checked) y no marcado.
- Hereda de CompoundButton, que a su vez hereda de TextView
- Algunos métodos aplicables:
  - isChecked(): Devuelve un booleano indicando si está marcado o no
  - toggle (): Cambia el estado al inverso del estado actual
  - setChecked (boolean ch): Pone el estado igual al valor del parámetro que le mandemos. Si le mandamos “true” lo pondrá marcado, si le mandamos “false” se desmarcará
- Ver proyecto ejemplo Basic / CheckBox

# RadioButton

---

- Es un botón con dos estados, pero a diferencia de los CheckBox, los RadioButton se pueden agrupar para que sólo uno de los RadioButton del grupo pueda estar marcado
- Hereda de CompoundButton y de TextView
- También se pueden usar métodos como isChecked (), toggle (), setChecked (boolean ch)...
- La mayoría de las veces agruparemos elementos RadioButton dentro de RadioGroup (ver siguiente diapositiva)

# RadioGroup

---

- Sirve para agrupar RadioButton
- Algunos métodos interesantes:
  - `check(int id)` para marcar un RadioButton
    - P.ej: `group.check(R.id.radio1)`
  - `clearCheck()` Para desmarcar todos los RadioButton
  - `getCheckedRadioButtonId()` para obtener el identificador del RadioButton que está marcado (-1 si ninguno lo está)
- Ver proyecto de ejemplo Rad



# ToggleButton y Switch

---

- Tienen la misma funcionalidad que CheckBox, únicamente cambia el aspecto



- Al igual que CheckBox, ToggleButton y Switch son subclases de CompoundButton, y por tanto podremos usar la mayoría de métodos que se usan con CheckBox

# Algunas propiedades interesantes

---

- padding, paddingLeft, paddingTop, paddingBottom, paddingRight
- visibility , tiene tres posibles valores:
  - VISIBLE
  - INVISIBLE: La vista se hará invisible pero seguirá ocupando su espacio en el layout
  - GONE: La vista no se verá y además su espacio quedará disponible para otro elemento
- nextFocusDown, nextFocusUp, nextFocusRight, nextFocusLeft para controlar el foco cuando el usuario usa un D-pad o similar

# Algunos métodos interesantes

---

- `setEnabled()` sirve para deshabilitar o habilitar un widget
- `isEnabled()` sirve para verificar si el widget está activo
- `requestFocus()` para solicitar el foco
- `isFocused()` para saber si el widget tiene el foco
- `getParent()` nos dice quién es el padre del widget
- `getRootView()` para obtener la raíz del árbol que contiene la vista sobre la que lo apliquemos (devuelve un `View`)