

## WIDGETS

~~import TextWidget~~

import android.widget.\*

Button miboton;

miboton = (Button) findViewById(R.id.number\_id);

TextView mitekto;

mitekto = (TextView) findViewById(R.id.number\_edit);

miboton.setOnClickListener (new

View.OnClickListener() {

public void onClick (View v) {

} } );

editText e = (EditText) findViewById(R.id.number\_edit);

## ~~CHECKBOX~~ CHECKBOX

- Descarga de TextWidget

- Listener

checkbox b = (checkbox) findViewById(R.id.c)

b.setOnClickListener (new CheckOnChecked (change listener (new check\_onchecked (change listener) )

public void onCheckedChange (CompoundButton button)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

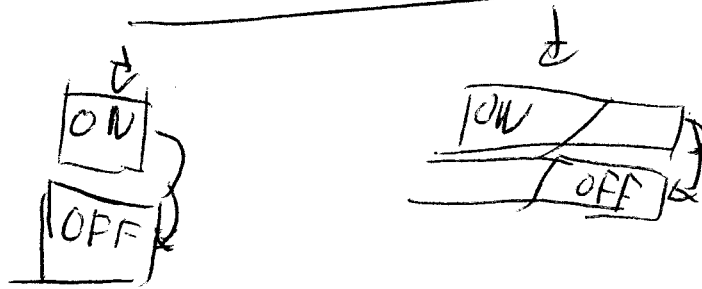
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

→ U. set G76 V75 (U. en. en. V75 V75)

Cartagena99

## TOGGLE BUTTON Y SWITCH



Android: checked = "false" or "true"

Android: Text On = "ON"

Android: Text Off = "OFF"

Listener: Igual que Radio Button y Checkbox.

## ~~EMBEDED~~ X ~~EMBEDDED~~ BUTTON

### EMBEDDED VIEW X EMBEDDED BUTTON

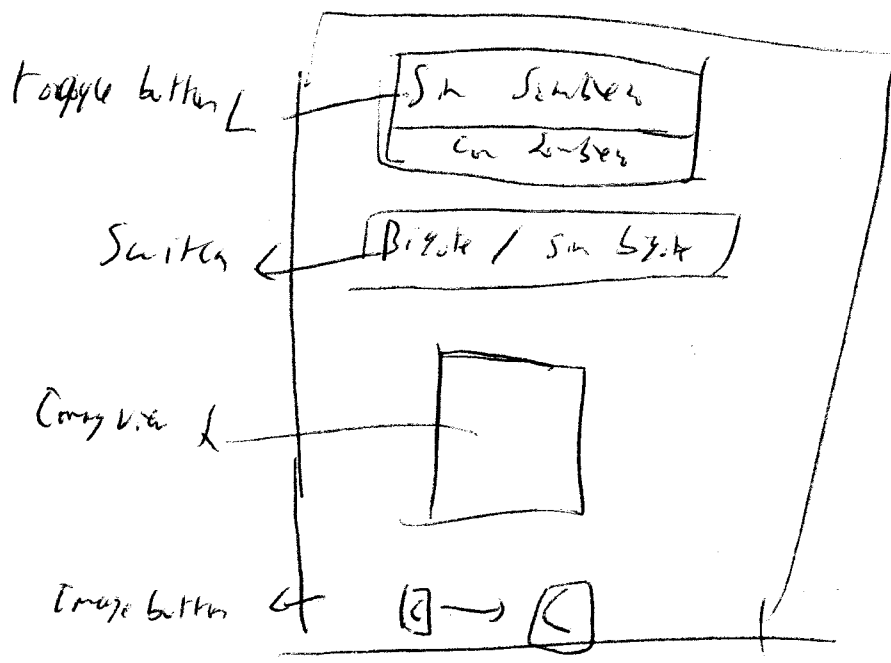
- Igual que TextView y button con imágenes.
- Las imágenes se guardan en Drawable.
- Formato: PNG.
- (Identificador) -> R.drawable.identificador
- Propiedad -> Android:src.
- Galería Android -> Botón de texto, carpeta y new image asset.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70



## EDIT TEXT

- Subclase de Text View.

- Propiedades:
  - ↳ Android: AutoText
  - ↳ Android: Digits
  - ↳ Android: InputType
    - ↳ Text
    - ↳ Number
    - ↳ Phone
    - ↳ TextEmailAddress

## STORAGE.XML

Hay que mantener el texto fuera del código fuente.

• Ruta Res/Values/Strings.xml.

Uso:

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
 CALL OR WHATSAPP: 689 45 44 70



Etiqueta String: Almacena el texto que describe el recurso.

Referencia: Desde el diseño de String/Recurso.

Desde el código: String s = getString(R.string.Recurso).

## EJERCICIO FDOT TEXT.

Re/Values → lenguaje original

introduce tus datos

Nombre:

DNI:

Telefono:

Email

Re/values - FR

Res/values - EN

Clonados.

Boton delete en Re/  
y alisar a new Resource  
delecion en Res.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

# CONTENEDORES

Gestionan la distribución de los widgets

Tipos:

- RelativeLayout: Modelos basados en reglas
- LinearLayout: Modelos basados en cajas
- TableLayout: Modelos basados en celdas

~~Se p~~

Se pueden anidar unos dentro de otros.

## RELATIVE LAYOUT

Permite a los hijos (contenidos) especificar su posición con relación al padre (contenedor) o hermanos.

Propiedades:

	Android: Layout_AlignParentTop = "top"	"	} se alinea a el borde superior del padre. true o false.
center " hijo vertical)	Android: Layout_centerVertical = "centerInParent"	"	
horizontalmente con el padre.	Android: Layout_centerHorizontal = "centerInParent"	"	
	Android: Layout_Bottom = "@+android:bottom"	"	} esto es elemento debajo / derecha de " "
	Android: Layout_Right = "@+android:right"	"	

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

## LINEAR LAYOUTS

### Alinear

permite alinear los contenidos horizontalmente en una fila o verticalmente en una columna.

Propiedades:

Android: Orientation =  $\left\{ \begin{array}{l} \text{"vertical"} \\ \text{"horizontal"} \end{array} \right.$

Android: Layout\_width =  $\left\{ \begin{array}{l} \text{"wrap\_content"} \\ \text{"wrap\_content"} \end{array} \right.$

Android: Layout\_height =  $\left\{ \begin{array}{l} \text{"wrap\_content"} \end{array} \right.$

Android: Layout\_gravity = " " ~~for se alinea de arriba~~

Todo se alinea por de arriba a abajo > de izquierda a derecha ~~por de~~  
por defecto.

## TABLE LAYOUTS Y GRID LAYOUT

permite crear una distribución basada en filas y columnas, es decir, mediante tablas.

Propiedades:

Android: RowCount = num

Android: ColumnCount = num

} especifica

numero de filas y columnas, empieza a 0.

Android: Layout\_Row = "Num".

Android: Layout\_Col = "Num".

} hay que especificar en cada widget

el # de fila y columnas pertenece.

Para que un elemento ocupe más de un celda:

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

Cartagena99