



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

PROGRAMACIÓN APLICADA III

VIEWS Y WIDGETS BÁSICOS



Universidad Nacional de Cajamarca



www.unc.edu.pe/



Universidad Nacional de Cajamarca

Índice



- ☐ Views
- ☐ Widgets básicos
- ☐ Práctica

I. Views

Un view es un objeto que sirve para dibujar algo en la pantalla con el cual el usuario pueda interactuar



I. Views

View

public class View

extends **Object** implements **Drawable.Callback**, **KeyEvent.Callback**,

java.lang.Object

↳ android.view.View

<https://developer.android.com/reference/android/view/View.html>

II. Widgets básicos

Un widget es un componente de la interfaz gráfica que puede o no permitir la interacción con el usuario.

Android proporciona un conjunto de componentes básicos que permiten crear interfaces gráficas más o menos complejas.

1. Declaración

Sintaxis

```
<Tipo_de_widget  
    android:id="@+[paquete:]id/nombre_recurso"  
    android:layout_width="dimensión"  
    android:layout_height="dimensión"
```

...

/>

2. Uso

Para poder utilizar el componente desde el código Java de la aplicación, basta con utilizar el método `findViewById` y pasarle como parámetro el identificador único del componente afectado. Este método devuelve un objeto de tipo `View` que hay que convertir a continuación en el tipo de clase adecuado.

Sintaxis

```
public View findViewById (int id)
```

II. Widgets básicos

a. TextView (campo de texto)

El componente TextView permite visualizar texto.

TextView

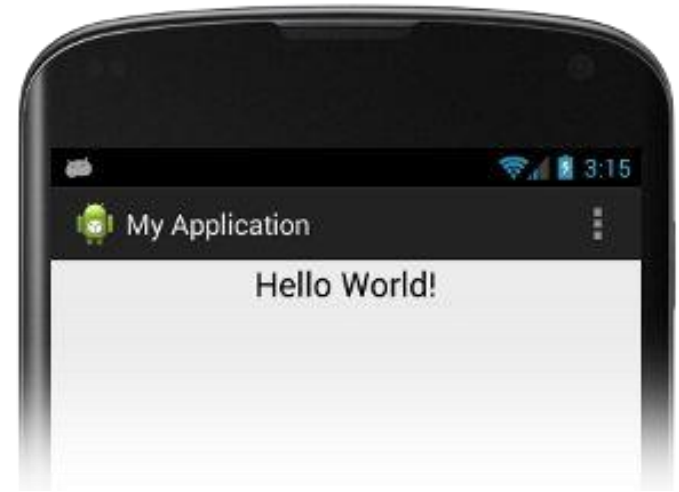
```
public class TextView
```

```
extends View implements ViewTreeObserver.OnPreDrawListener
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```



```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!" />
```

<https://developer.android.com/reference/android/widget/TextView.html>

II. Widgets básicos



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

a. TextView (campo de texto)

Las principales propiedades del componente TextView son:

Propiedad	Descripción
<code>android:autoLink</code>	Permite convertir automáticamente los enlaces de hipertexto y las direcciones de correo electrónico en enlaces sobre los que es posible hacer clic.
<code>android:ellipsize</code>	Especifica cómo debe mostrarse el texto cuando es más largo que la vista.
<code>android:gravity</code>	Indica dónde debe estar posicionado el texto en la vista cuando es más pequeño que ella.
<code>android:height</code>	Altura del componente.
<code>android:lines</code>	Número exacto de filas que se desea mostrar.
<code>android:maxHeight</code>	Altura máxima del componente.
<code>android:maxLines</code>	Número máximo de filas que se desea mostrar.
<code>android:maxLength</code>	Longitud máxima del componente.
<code>android:minLines</code>	Número mínimo de filas que se desea mostrar.
<code>android:text</code>	Texto que se quiere mostrar.
<code>android:textColor</code>	Color del texto.
<code>android:textSize</code>	Tamaño del texto.
<code>android:textStyle</code>	Estilo del texto.
<code>android:width</code>	Longitud del componente.

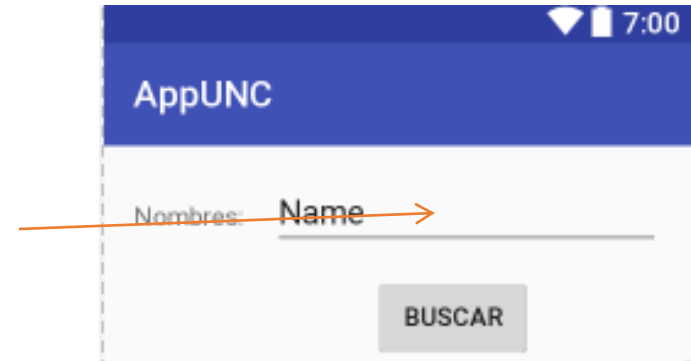
II. Widgets básicos

b. EditText (campo para introducir texto)

El componente EditText permite al usuario introducir texto. Cuando el usuario hace clic sobre el componente, aparece el teclado virtual. El usuario puede utilizar también el teclado físico si el dispositivo posee uno.

EditText

```
public class EditText  
extends TextView  
  
java.lang.Object  
↳ android.view.View  
    ↳ android.widget.TextView  
        ↳ android.widget.EditText
```



```
<EditText  
    android:id="@+id/email_address"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:hint="@string/email_hint"  
    android:inputType="textEmailAddress" />
```

<https://developer.android.com/reference/android/widget/EditText.html>

II. Widgets básicos

b. EditText (campo para introducir texto)

Este componente hereda del componente TextView. Sus propiedades son las mismas que las del componente TextView. Las principales propiedades utilizadas por este componente son:

Propiedad	Descripción
<code>android:inputType</code>	Permite filtrar el texto introducido por el usuario especificando el tipo de dato que acepta el componente.
<code>android:scrollHorizontally</code>	Permite deslizar el texto horizontal cuando es mayor que la longitud del componente.

II. Widgets básicos

c. **Button** Es una porción de texto, imagen o ambos que nos ayudan a establecer comunicación con acciones cuando el usuario lo toca.

Button

```
public class Button  
extends TextView
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```

```
↳ android.widget.Button
```



```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_text"  
    ... />
```

```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/button_icon"  
    ... />
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_text"  
    android:drawableLeft="@drawable/button_icon"  
    ... />
```

<https://developer.android.com/reference/android/widget/Button.html>

II. Widgets básicos

c. Button

Este componente hereda del componente TextView. Sus propiedades son las mismas que las del componente TextView. Las principales propiedades utilizadas por el componente son:

Propiedad	Descripción
<code>android:text</code>	Texto a mostrar sobre el botón.
<code>android:onClick</code>	Nombre del método de la actividad a ejecutar cuando el usuario hace clic sobre el botón.

Sintaxis 1

`public void nombreMétodo (View vista Asociada)`

Sintaxis 2

`public void setOnClickListener (View.OnClickListener l)`

Práctica



- Realice una aplicación móvil para registrar una **Personas** que almacene **nombres, apellidos, edad, DNI, peso y altura**. Los métodos que se implementaran son:
- calcularIMC():** calculará si la persona está en su peso ideal ($\text{par} = \text{peso en kg} / (\text{altura}^2 \text{ en m})$), devuelve un -1 si está por debajo de su peso ideal ($\text{par} < 20$), un 0 si está en su peso ideal ($\text{par} \geq 20$ y $\text{par} \leq 25$) y un 1 si tiene sobrepeso ($\text{par} > 25$).
- esMayorDeEdad():** indica si es mayor de edad, devuelve un booleano.
- toString():** devuelve toda la información del objeto.
- verificarDNI():** verifica si el DNI es correcto y tiene sólo números de longitud 8 dígitos.
- Tiene dos botones:
- Registrar, registra el objeto en un array dinámico si el DNI es correcto y luego muestra un mensaje de la persona sobre el tipo de peso que tiene y si es mayor o menor de edad, caso contrario sale un mensaje el DNI es incorrecto.
 - Listar, listar todas las personas registradas en una nueva actividad mediante un LISTVIEW con los datos nombre completo, tipo de peso, si es mayor de edad o no.