

INTERFACE DE USUARIO: COMPONENTES BÁSICOS



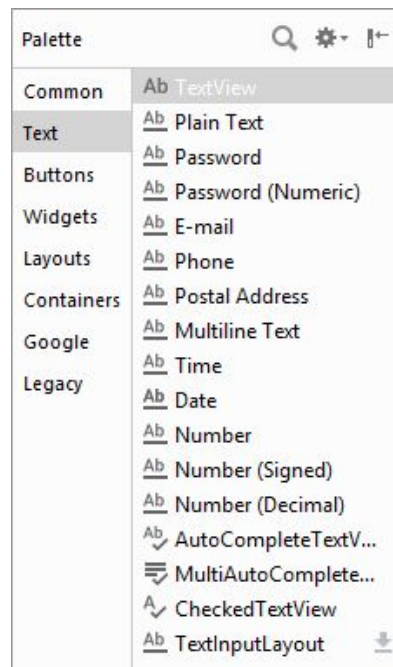
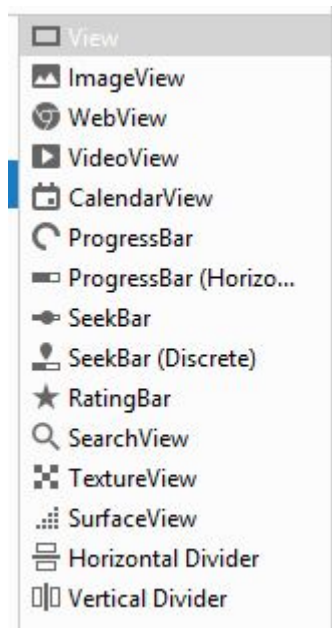
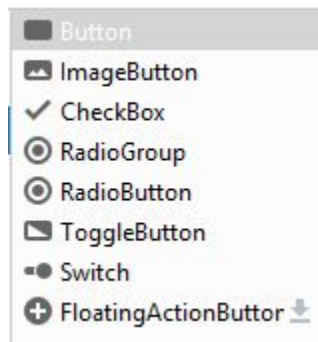
Programación multimedia
y dispositivos móviles
2º DAM

ÍNDICE DE CONTENIDOS

1. Introducción
2. Propiedades obligatorias
3. Propiedad id
4. Comunicación de componentes
5. Propiedad onClick
6. Otros eventos

1. INTRODUCCIÓN

- Una vez que se ha realizado la distribución de la actividad podremos comenzar a colocar los distintos elementos que la componen, botones, textView, editText...
- Existen gran multitud de variedades.



2. PROPIEDADES OBLIGATORIAS

Cada componente que añadimos al Layout viene identificado con el tipo de componente que es, y sus propiedades.

- Todo elemento tiene dos propiedades obligatorias: `layout_width` y `layout_height` (ancho y alto del elemento)
- Estas medidas pueden ser:
 - `wrap_content` → lo que necesite (se va adaptando al contenido)
 - `match_parent` → ocupa lo mismo que el contenedor en el que se encuentra .

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/hola"/>
```

3. PROPIEDAD ID

El resto de los atributos son opcionales, aunque hay un atributo bastante necesario.

- id -> permite identificar al elemento. Sólo puede aparecer cada nombre una única vez en un layout.

```
<TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hola"/>
```

Para declarar el id de un componente necesitamos utilizar el @+id junto con el nombre deseado

3. PROPIEDAD ID

La propiedad id tiene dos usos:

- Utilizar el componente como referencia dentro de un layout para otro componente:

```
<TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hola"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botón 1"
    android:id="@+id/boton"
    android:layout_alignLeft="@id/text"
    android:layout_below="@id/text"/>
```

Como se puede observar, para hacer referencia al componente se realiza sin el "+"
@id/nombre

4. COMUNICACIÓN DE COMPONENTES

El layout principal (el primero en jerarquía) nos indicará a qué actividad pertenece con la propiedad `tools:context`.

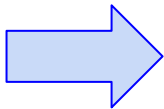
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```

4. COMUNICACIÓN DE COMPONENTES

- Para poder manejarlo desde el código java de la aplicación: Se crean objetos de la clase del componente, y mediante el método findViewById se indica el identificador del mismo.

```
public class MainActivity extends AppCompatActivity {
```

```
    EditText editText;  
    Button b1;  
    Button b2;
```



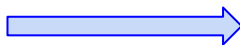
Se crean los objetos, se puede hacer de forma global, o dentro del método onCreate

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

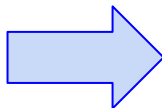
```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```



Se indica que layout debe cargar.

```
    editText = (EditText) findViewById(R.id.edit);  
    b1 = (Button) findViewById(R.id.b1);  
    b2 = (Button) findViewById(R.id.b2);
```



Se indica a qué componente hace referencia

Nota: En el tema 3 se verá el código de las actividades

5. PROPIEDAD ONCLICK

Hay una propiedad, que nos permite crear los eventos que se lanzan al interactuar un usuario con nuestra aplicación, la propiedad `onClick`. Es un evento que permite realizar una acción al pulsar sobre el componente, y se puede implementar en todos los componentes (aunque lo más común es su uso en los botones).

Esta propiedad nos permite crear el evento en el código Java, aunque veremos otras formas de implementarlo.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botón 1"
    android:id="@+id/b1"
    android:layout_alignLeft="@id/edit"
    android:layout_below="@id/edit"
    android:onClick="pulsarBoton"/>
```



Esta propiedad nos creará este método en la actividad correspondiente

```
public void pulsarBoton(View view) {
}
```

6. OTROS EVENTOS

A parte del evento **OnClick** para trabajar al pulsar un componente, existen otros eventos que se indican desde el código Java.

- **OnLongClickListener** → dejar pulsado
- **OnKeyListener** → en los editText, al escribir se pueden detectar las letras.
- **OnDragListener** → evento al arrastrar. <https://developer.android.com/guide/topics/ui/drag-drop>
- **OnFocusChangeListener** → evento al cambiar el foco (recibirlo o perderlo)
- **OnTouchListener** → evento táctil sobre la vista

6. DUDAS Y PREGUNTAS

