

## Vistas (1)

Las interfaces en Android tienen dos partes:

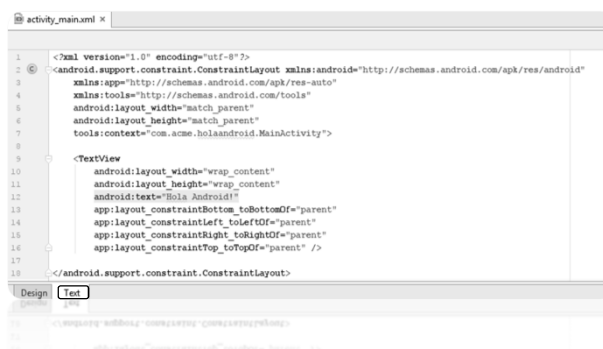
- **Estática:** uno o varios archivos XML en el que se definen los distintos elementos (botones, etiquetas, campos de texto, etc.). Todos heredan de la clase *View*.
- **Dinámica:** archivo java asociado a una *Activity* en el que se definen las interacciones del usuario y sus tratamientos.

2

## Vistas (2)

Podemos modificar una vista de dos formas:

- **Modo edición** (pestaña **Text**):  
Modificando el código XML de la interfaz.

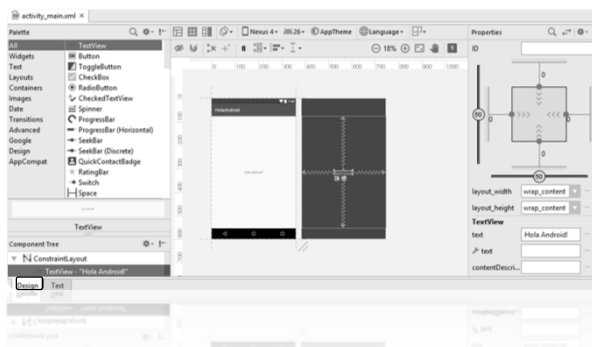


3

## Vistas (3)

### ■ **Modo diseño** (pestaña *Design*):

Utilizando el editor de interfaces para añadir o quitar elementos (*drag & drop*), modificar sus propiedades, etc.

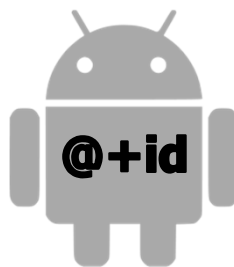


4

## Vistas (4)

### **Identificadores**

Un identificador es un nombre único asignado a un elemento de una vista que permite referenciarlo para implementar sus interacciones y tratamientos.



5

## Vistas (5)

### Identificadores

#### ■ Declaración:

```
android:id="@+id/nombre_id"
```

- **android:id**: nombre del atributo.
- **@+id**: declaración de nuevo identificador.
- **nombre\_id**: identificador del elemento

6

## Vistas (6)

### Identificadores

#### ■ Acceso desde XML:

```
@id/nombre_id
```

#### ■ Acceso desde Java:

```
R.id.nombre_id
```

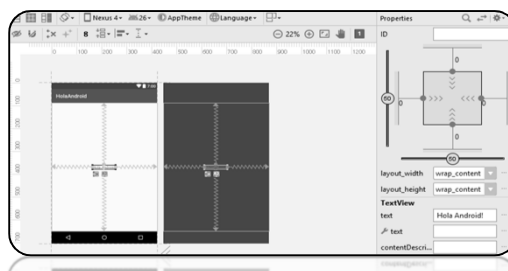
7

## Vistas (7)

### Tamaño de los elementos

Cada vez que declaremos un elemento de una vista (contenedor o componente) debemos especificar sus dimensiones:

- Altura: **android:layout\_height**
- Anchura: **android:layout\_width**



8

## Vistas (8)

### Tamaño de los elementos

Estos valores se pueden especificar de varias formas:

- **match\_parent**: el tamaño del elemento será igual que el de su elemento padre.
- **wrap\_content**: el tamaño del elemento será igual que el de su contenido.
- **valor**: se establece el tamaño mediante un valor numérico fijo. Hemos de utilizar **dp** (*density-independent pixels*).

9

## Vistas (9)

### Asociar una vista a una actividad

Una vez creada la parte estática de la interfaz (XML), hay que crear una clase Java que represente su actividad.

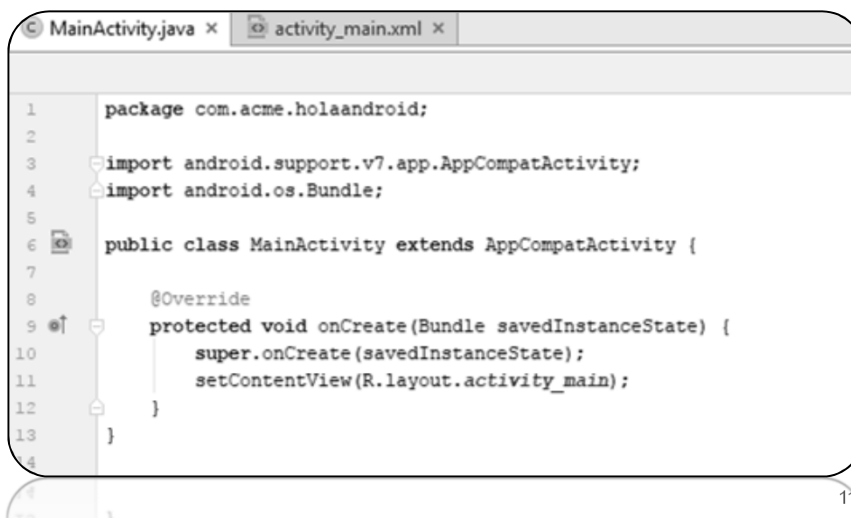
Esta clase debe:

- Heredar de la clase ***AppCompatActivity***.
- Sobrecargar el método ***onCreate*** (al menos).
- Enlazar la actividad con su interfaz mediante el método ***setContentView***.

10

## Vistas (10)

### Asociar una vista a una actividad



```
1 package com.acme.holaandroid;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
```

11

## Vistas (11)

### Asociar una vista a una actividad

Por último tendremos que declarar la actividad dentro de **<application>** del archivo "AndroidManifest.xml".

```
<activity android:name=".MainActivity">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>
```

```
</activity>
```

12