10. Gravedad

Los atributos de gravedad permiten controlar la alineación del contenido de las vistas (`gravity`), así como la alineación de las vistas dentro de sus contenedores (`layout_gravity`):

- `android:gravity`, en un contenedor tal y como `LinearLayout`, alinea los elementos situados dentro del contenedor según el valor asignado (derecha, izquierda, ...). Dentro de una vista como `Button`, alinea el texto dentro del botón.
- `android:layout_gravity` en una vista como `Button`, por ejemplo, alinea el botón dentro del contenedor en el que se encuentra, según el valor asignado.

Los valores que pueden tomar estos atributos son los siguientes:

- `top`
- `bottom`
- `left`
- `right`
- `center_vertical`
- `fill_vertical`
- `center_horizontal`
- `fill_horizontal`
- `center`
- `fill`
- `clip_vertical`
- `clip_horizontal`
- `start`
- `end`

A continuación se utilizan cuatro proyectos para ilustrar el significado de algunos de estos valores. Todos los proyectos comparten el fichero Java salvo por el nombre del paquete. Este fichero se encarga de inflar la interfaz gráfica especificada en el fichero de diseño `activity_main.xml`:

`/src/MainActivity.java`

```
package es.uam.eps.dadm.ccc10_1;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

En el siguiente proyecto (ccc10_1) utilizamos el atributo `gravity` para alinear el texto de los botones: Burgos queda alineado a la izquierda y Madrid a la derecha. Fíjate en que el atributo `android:layout_width` de los botones debe tener el valor `match_parent` para conseguir el efecto deseado (utilizamos por claridad una cadena en lugar de un recurso en el valor del atributo `android:text`):

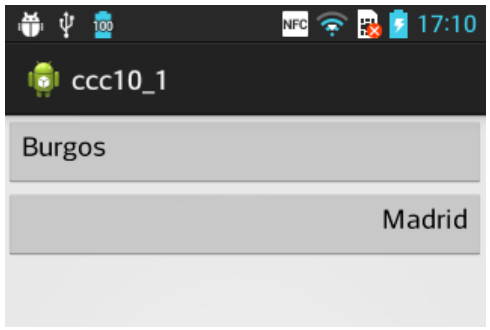
/res/layout/activity_main.xml

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

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="Burgos" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="Madrid" />

</LinearLayout>
```



En el siguiente proyecto (ccc10_2) son los botones, no su contenido, los que se alinean a la izquierda y derecha dentro del contenedor `LinearLayout` gracias al atributo `android:layout_gravity`. El atributo `android:layout_width` de los botones se iguala a `wrap_content` en esta ocasión (comprueba lo que ocurre si dejas el valor `match_parent`):

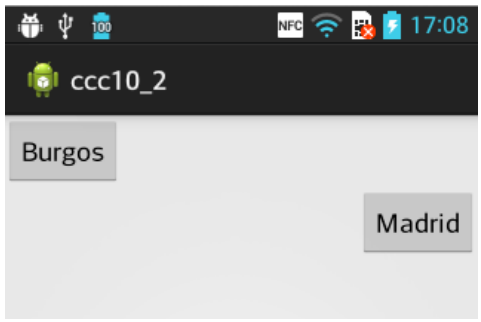
/res/layout/activity_main.xml

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

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:text="Burgos" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Madrid" />

</LinearLayout>
```



El valor `android:layout_gravity="center_horizontal"` del botón número 1 del ejemplo ccc10_3 hace que el botón quede centrado dentro del `LinearLayout` que lo contiene. El `LinearLayout` interior con `android:layout_gravity="left"` queda alineado a la izquierda del `LinearLayout` principal, con lo que el botón número 2 aparece a la izquierda del todo:

/res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="1" />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_gravity="left">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="2" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="3" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="4"/>
    </LinearLayout>
</LinearLayout>
```

En el último ejemplo (ccc10_4), el valor `android:gravity="center_horizontal"` del `LinearLayout` principal fuerza a que tanto el botón número 1 como el `LinearLayout` interior, que contiene a los botones 2, 3 y 4, queden centrados horizontalmente:

/res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1" />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="2" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="3" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="4" />
    </LinearLayout>
</LinearLayout>
```