

Tutorial 3

3. Diseño de la interfaz de usuario



*Curso de perfeccionamiento: Programación de Android
2013*



3. Diseño de la interfaz de usuario

- Creación de una interfaz de usuario usando XML
 - Fichero **res/layout/main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

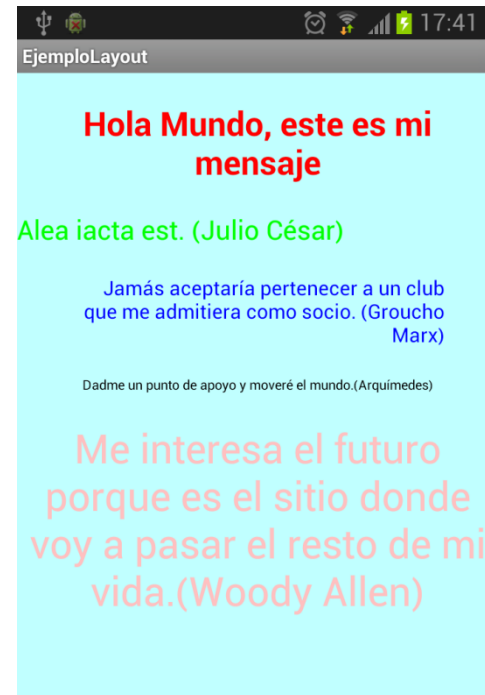
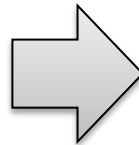
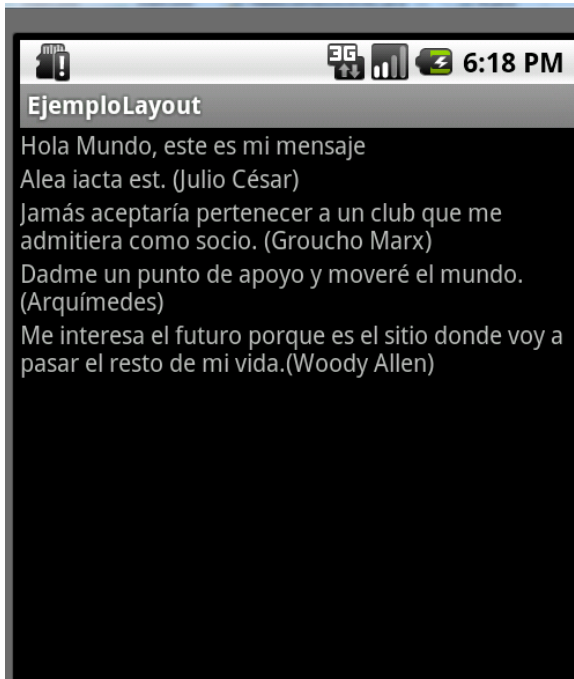
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```



3. Diseño de la interfaz de usuario

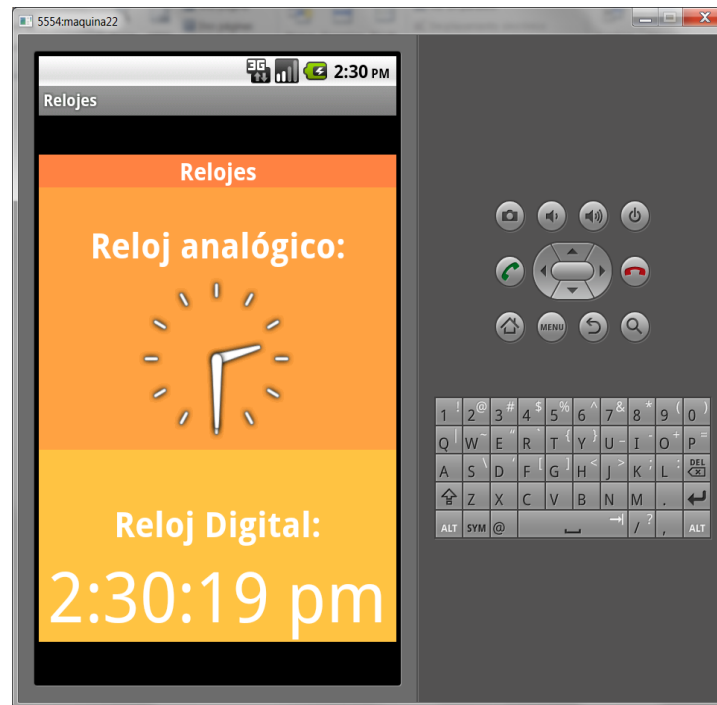
Proyecto ejemplo: EjemploLayout



Sigue las instrucciones del profesor para crear este ejemplo.

3. Diseño de la interfaz de usuario

Ejercicio: Relojes



Intenta realizar esta sencilla app, modificando sólo main.xml y strings.xml.

3. Diseño de la interfaz de usuario

LAYOUT:

- Se denomina “layout” a la arquitectura de la interfaz de usuario de una actividad.
- Puede declararse de 2 formas:
 - Declarando elementos en XML denominados vistas (*Views*) en ***main.xml***.
 - Creando instancias de los objetos (*Views*) en tiempo de ejecución dentro del código Java.

<http://developer.android.com/guide/topics/ui/declaring-layout.html>

3. Diseño de la interfaz de usuario

LAYOUT en *main.xml*:

- Cada fichero XML debe tener un elemento contenedor raíz que agrupe en su interior a los objetos y widgets (*Views*).
- Por ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

- Se ubica en *res/layout/main.xml*

3. Diseño de la interfaz de usuario

Carga del recurso *main.xml*:

- El fichero XML debe cargarse en el código de la aplicación, dentro del método ***onCreate()*** llamando a la función ***setContentView()*** y pasándole como referencia el nombre del fichero XML.
- Por ejemplo:

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main_layout);  
}
```

3. Diseño de la interfaz de usuario

- **Tipos de Layouts:**

- ***LinearLayout***: elementos en fila o columna
- ***TableLayout***: elementos en una tabla
- ***RelativeLayout***: en relación uno con otro
- ***AbsoluteLayout***: posicionamiento absoluto
- ***FrameLayout***: cambio dinámico de los elementos
- Otros layouts más avanzados: ***ScrollView***, ***ListView***, ***GridView***, ***TabHost***, ***ViewFlipper***,...

<http://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts>

3. Diseño de la interfaz de usuario

- **Tipos de Layouts:**

Linear Layout



A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

Relative Layout



Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

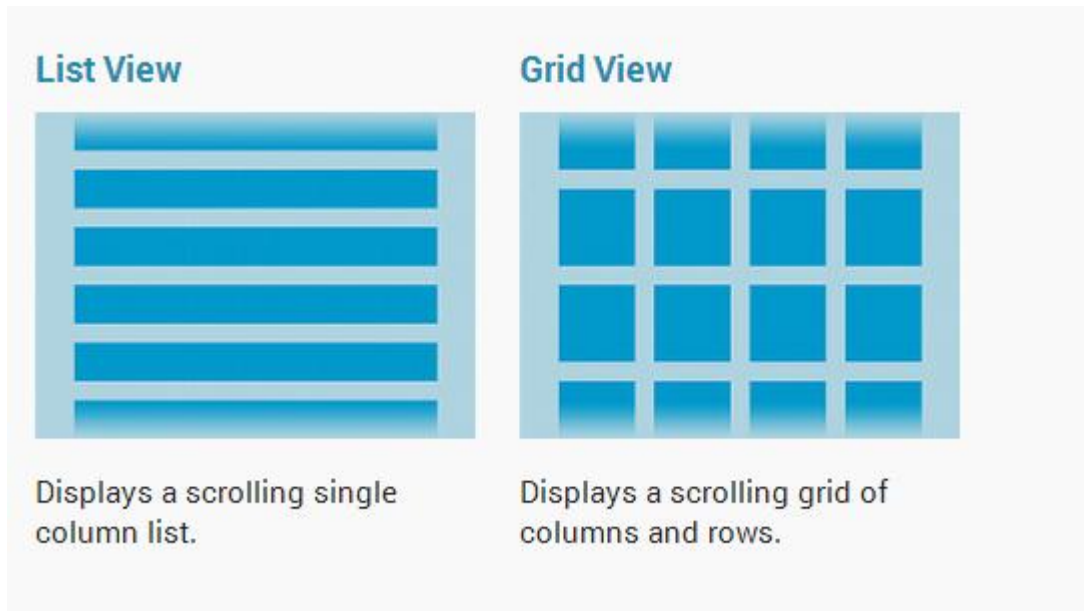
Web View

```
<html>  
  <!-- web page -->  
</html>
```

Displays web pages.

3. Diseño de la interfaz de usuario

- **Tipos de Layouts:**



3. Diseño de la interfaz de usuario

- **Algunos elementos (*Views*):**
 - *TextView*
 - *Button*
 - *CheckBox*
 - *AnalogClock*
 -

<http://developer.android.com/reference/android/widget/package-summary.html>

3. Diseño de la interfaz de usuario

- **Atributos de las vistas (*Views*):**
 - Cada objeto tipo *View* tiene una serie de atributos o propiedades que pueden modificarse desde el fichero XML o bien desde el código en tiempo de ejecución.

```
<TextView
    android:id="@+id/titulo"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#845C71"
    android:gravity="center"
    android:text=" Citas Célebres "
    android:textColor="#FDFDD9" />
```

3. Diseño de la interfaz de usuario

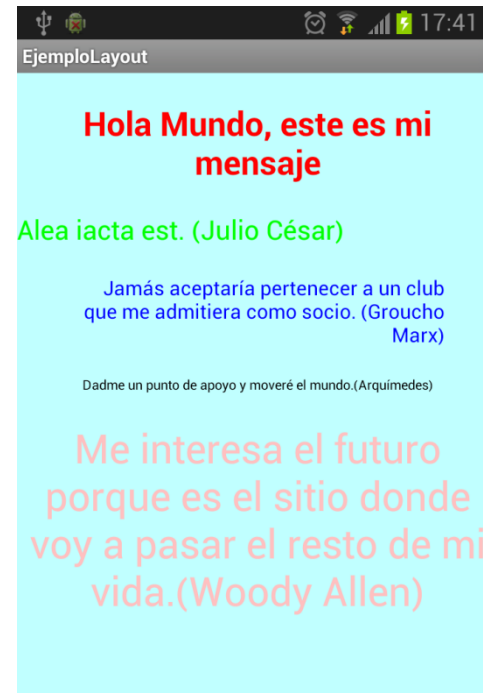
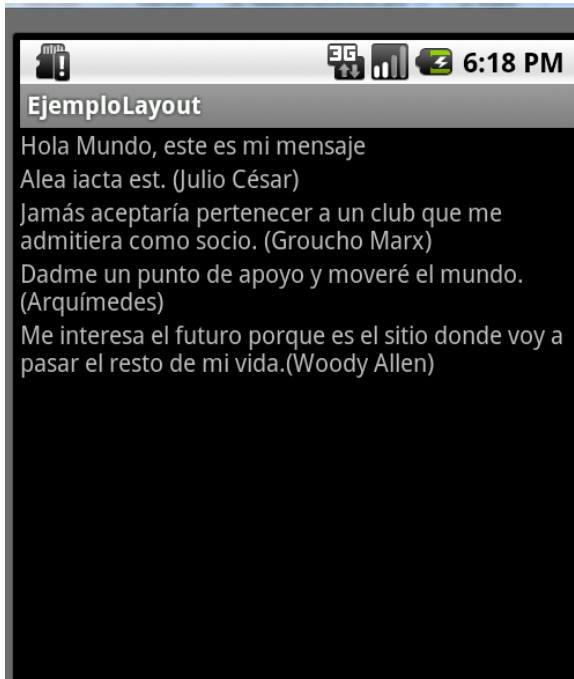
- **Creación de instancias en el código:**

- Para modificar desde el código los atributos de los elementos **View** hay que crearse una instancia del objeto dentro del método *onCreate()*:

```
TextView mititulo;  
  
mititulo = (TextView)findViewById(R.id.titulo);  
mititulo.setText("CITAS CÉLEBRES") ;
```

3. Diseño de la interfaz de usuario

Proyecto ejemplo: EjemploLayout



Sigue las instrucciones del profesor para crear este ejemplo.

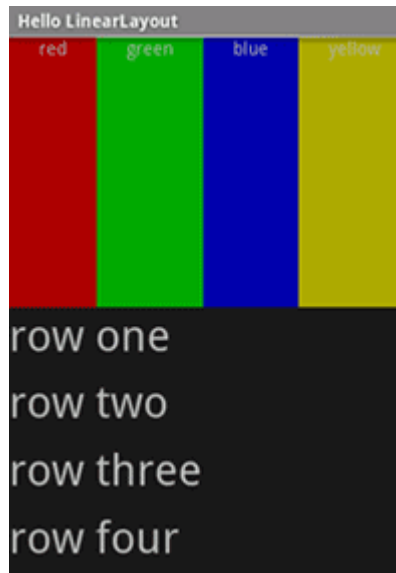
3. Diseño de la interfaz de usuario

Tutoriales: Hello Views

A continuación realizaremos una colección de tutoriales tipo "*Hello World*" diseñados para conocer los principales layouts y widgets de Android.

3. Diseño de la interfaz de usuario

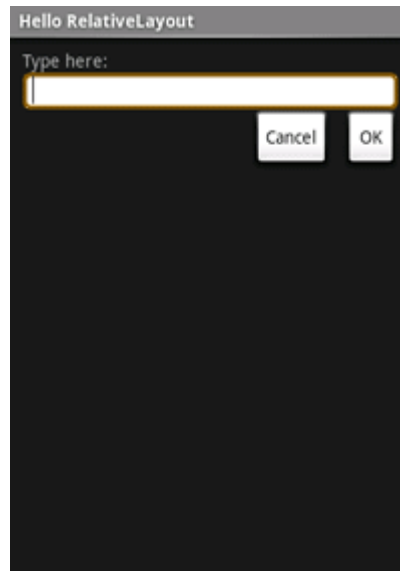
*Proyecto ejemplo: **HelloLinearLayout***



Tutorial_HelloLinearLayout.pdf

3. Diseño de la interfaz de usuario

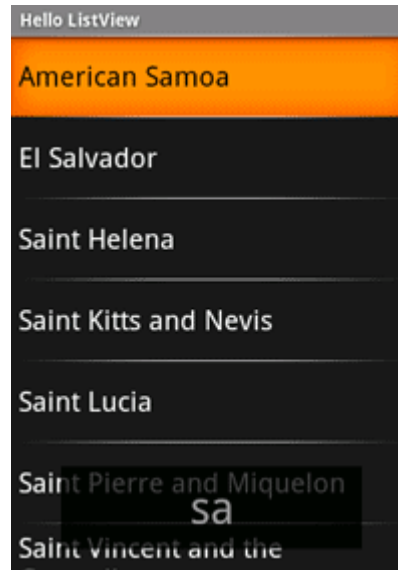
*Proyecto ejemplo: **HelloRelativeLayout***



Tutorial_HelloRelativeLayout.pdf

3. Diseño de la interfaz de usuario

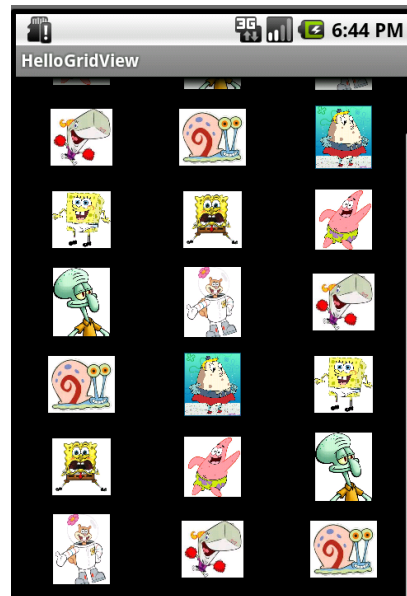
Proyecto ejemplo: HelloListView



Tutorial_HelloListView.pdf

3. Diseño de la interfaz de usuario

Proyecto ejemplo: HelloGridView



Tutorial_HelloGridView.pdf

Y no olvides descargarte el zip con las imágenes: **sample_images_bobesponja.zip**