



Primera App en Android Studio (II)

PROCEDIMIENTO GUIADO

Seguimos en la **App_UT2_01**

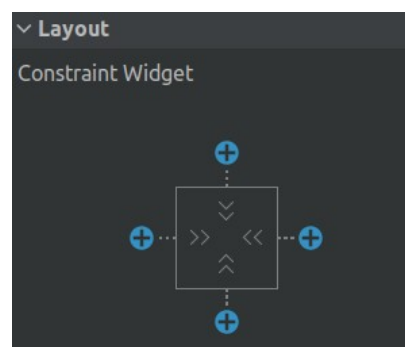
En el editor de diseño, haz clic para seleccionar TextView y bórralo. Ahora sólo tendremos una ConstraintLayout en el editor de diseño y en el Component Tree.

TextView

En Palette busca **TextView** (aparece tanto en la categoría Common como en Text) y arrástrala al editor de diseño.

Nos saldrá un error que indica que no hay ninguna restricción.

Queremos que el componente esté en el medio de la pantalla. Hay que añadirle restricciones para indicar cómo posicionarlo. A la derecha, en Attributes, busca **Constraint Widget** en la sección Layout



Para posicionar el TextView verticalmente podemos hacer clic en el símbolo + de la parte superior. Esto nos añadirá un margen. Si establezco, por ejemplo 200 estará a 200 dp (píxeles independientes de la densidad)

Como yo quiero centrarla verticalmente lo más sencillo es que añada restricciones en la zona superior y en la inferior. Pondré ambos valores a 0 (parent). Prueba a modificar valores para ver los distintos posicionamientos

Con esto lo centraré verticalmente, pero aún falta las restricciones horizontales. También quiero que esté centrado, así que de nuevo establezco a 0 tanto izquierda como derecha.

En el XML vemos que se han añadido las 4 restricciones:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

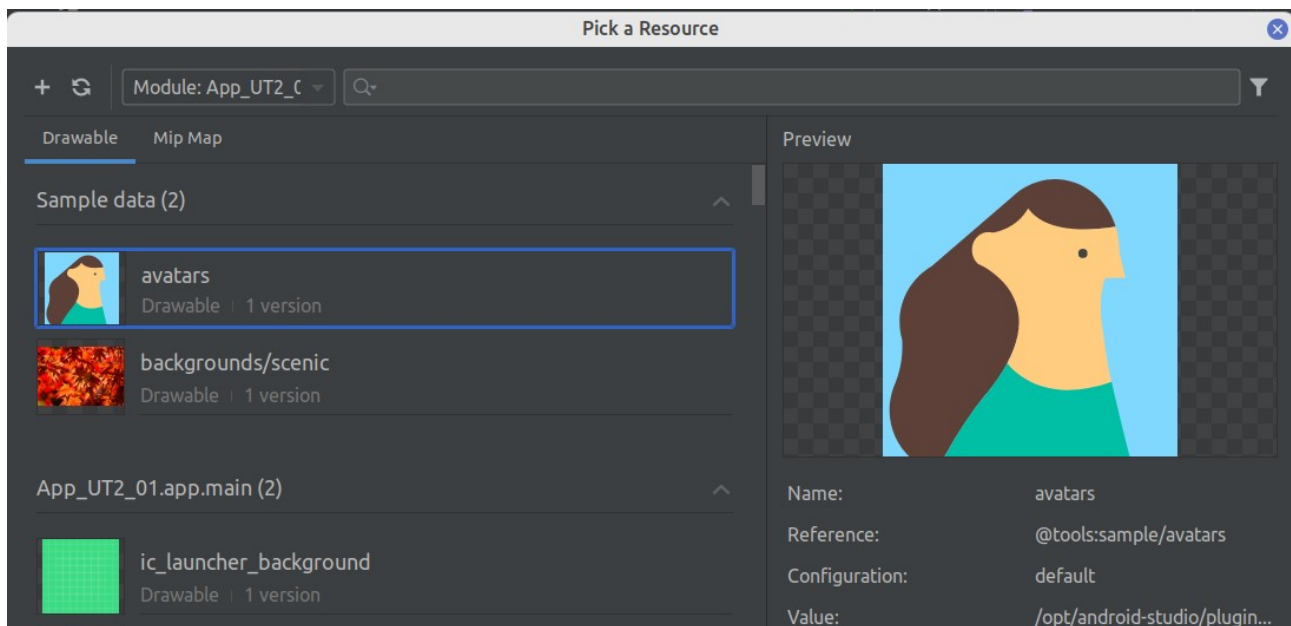


Hoja_UT2_02

ImageView

Añadiremos ahora una imagen y cambiaremos nuestro diseño. La imagen estará centrada y el texto estará posicionado justo debajo de ella, dejando un margen de 32 dp.

El componente es un `ImageView`. Cuando lo arrastremos en la vista de diseño nos mandará elegir una imagen. Podemos elegir avatars de momento.



Ahora la centraremos tal y como hemos hecho antes con el texto.

Si nos fijamos en el diseño que tenemos ahora, no es el deseado. Tenemos que cambiar las restricciones del texto para que se coloque debajo de la imagen (para que dependa de ella).

Para eso, nos iremos al código y cambiaremos `app:layout_constraintTop_toTopOf="parent"` por `app:layout_constraintTop_toBottomOf="@id/imageView"` (esto significa que el borde superior del texto estará justo debajo de...) y añadimos el identificador de la imagen.

Ahora lo centra entre el fondo de la pantalla y la imagen. Para ponerlo justo debajo borro la restricción `layout_constraintBottom_toBottomOf`

Ya casi lo tenemos. Le añadimos un `marginTop` a la imagen y listo:

`android:layout_marginTop="32dp"`

El código de los componentes es el siguiente:

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```



Hoja_UT2_02

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
tools:srcCompat="@tools:sample/avatars" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/imageView" />
```

Si quisiésemos que todo estuviese más arriba podríamos “jugar” con el **verticalBias**. Si le establecemos valor 0.25 nos quedará al 25% del borde superior:

```
app:layout_constraintVertical bias="0.25"
```

Para finalizar la tarea vamos a modificar el texto (poniendo tu nombre) y el estilo de TextView, añadir un botón en la parte inferior y a modificar la imagen

Cambiar texto y apariencia TextView

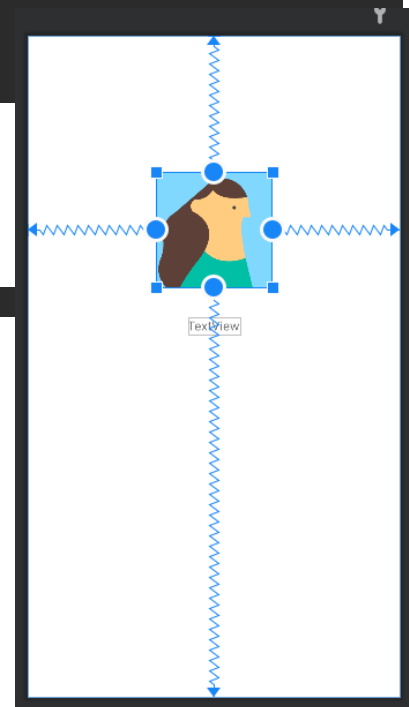
Podríamos cambiar el texto del TextView directamente en la propiedad text, pero no sería lo más adecuado. Lo ideal es añadir textos en el fichero **strings.xml** (dentro de la carpeta src/main/res) y asociarlos luego a componentes. Esto se debe hacer así ya que, de este modo, podremos realizar apps que se adapten a varios idiomas.

Tu archivo strings.xml deberá ser similar al siguiente:

```
<resources>
    <string name="app_name">App_UT2_01</string>
    <string name="ivan">Iván</string>
</resources>
```

Después, asociaremos el string al atributo **text** del TextView:

```
android:text="@string/ivan"
```





Hoja_UT2_02

Para cambiar la apariencia del texto podemos modificar los atributos **textSize**, **textColor** y **fontFamily**.

Ponemos el **textSize** a 34sp (sp son píxeles escalables y es la medida que se usa para el texto) y **textColor** a `@color/teal_200`

Añadir imágenes

Para añadir una imagen vamos a la pestaña Resource Manager (debajo de Project) y hacemos clic en + y seleccionamos **Import Drawables** y seleccionaremos las imágenes proporcionadas por el profesor.

Estableceremos en `srcCompat` cualquiera de las 4 imágenes proporcionadas.

Button

Añadimos un botón y lo posicionamos para que nos quede en posicionado tal y como se ve en la siguiente imagen.

En el `MainActivity` tendremos un mapa en el que se asocia un nombre y un avatar.

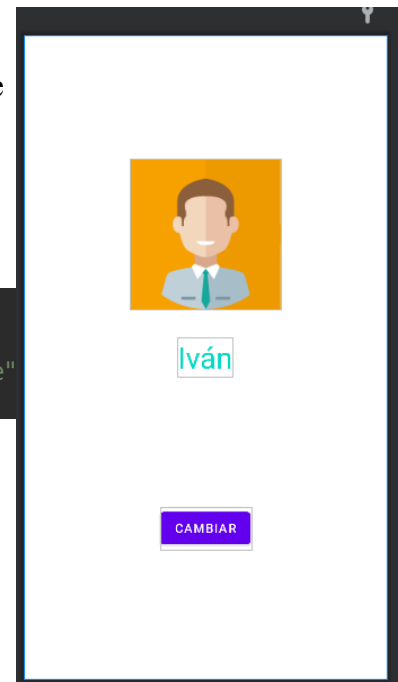
Por ejemplo: Iván corresponde al `avatar3`, María al `avatar1`, Rosa al `avatar2` y Pepe al `avatar4`

```
private val datos = linkedMapOf<String, Int>("Iván" to  
R.drawable.avatar3,  
"María" to R.drawable.avatar1, "Rosa" to R.drawable.avatar2, "Pepe"  
to R.drawable.avatar4)
```

Investiga ahora cómo cambiar el texto del `TextView` y el avatar al pulsar en el botón “Cambiar”. Empezaremos mostrando el primero de la colección y cada vez que se pulse el botón cambiar deberá ir avanzando en esta y mostrando los siguientes valores.

Además, mostraremos un `Toast` (mensaje emergente) de duración corta en la aplicación (también cuando se pulse el botón)

```
Toast.makeText(this, "Has cambiado el valor a $texto", Toast.LENGTH_SHORT).show()
```





Hoja_UT2_02

