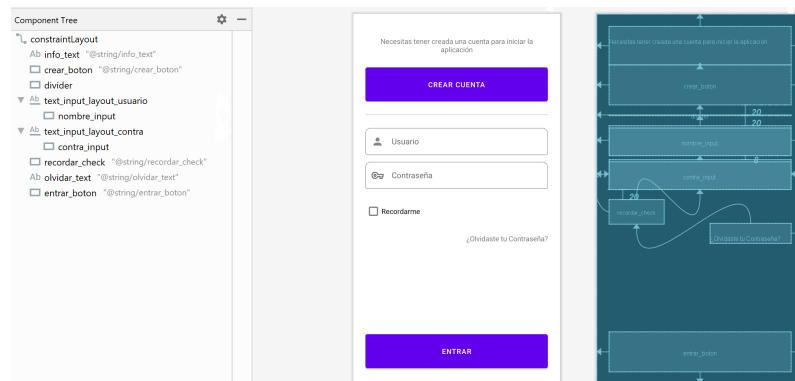


Ejercicio resuelto diálogo personalizado

[Descargar estos apuntes](#)

Vamos a desarrollar una aplicación (`EjemploDialogoPersonalizado`) que abra un diálogo diseñado por nosotros, como se muestra abajo, tras pulsar un botón en la pantalla de la actividad principal. Debéis añadir un botón en el layout `main_activity.xml`, y crear el xml para el diálogo.



Daremos de alta los siguientes recursos en el archivo `res/values/strings.xml` :

```
<resources>
    <string name="app_name">EjemploDialogoPersonalizado</string>
    <string name="info_text">Necesitas tener creada una cuenta para iniciar la aplicación</string>
    <string name="crear_boton">CREAR CUENTA</string>
    <string name="nombre_input">Nombre</string>
    <string name="contrasena_input">Contraseña</string>
    <string name="recordar_check">Recordarme</string>
    <string name="olvidar_text">¿Olvidaste tu Contraseña?</string>
    <string name="entrar_boton">ENTRAR</string>
</resources>
```

Y para los márgenes y paddings, creamos `res/values/dimens.xml` :

```
<resources>
    <!-- Medidas -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="dialog_body">24dp</dimen>
    <dimen name="padding_between">20dp</dimen>
    <dimen name="edit_text_padding">16dp</dimen>
    <dimen name="normal_padding">8dp</dimen>
    <dimen name="button_padding">24dp</dimen>
</resources>
```

El archivo xml correspondiente a la interfaz **dialogo.xml** sería:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/constraintLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/white"
    android:padding="@dimen/dialog_body">

    <TextView
        android:id="@+id/info_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:paddingTop="@dimen/padding_between"
        android:paddingBottom="@dimen/padding_between"
        android:text="@string/info_text"
        android:textAppearance="?android:attr/textAppearanceSmall"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/crear_boton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/button_padding"
        android:paddingBottom="@dimen/button_padding"
        android:text="@string/crear_boton"
        android:textColor="@android:color/white"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/info_text" />

    <View
        android:id="@+id/divider"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:layout_marginTop="@dimen/padding_between"
        android:layout_marginBottom="@dimen/padding_between"
        android:background="#C8C9CB"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/crear_boton" />

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/text_input_layout_usuario"
        style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/padding_between"
        android:hint="Usuario"

```

```

app:boxCornerRadiusTopEnd="10dp"
app:boxCornerRadiusTopStart="10dp"
app:startIconDrawable="@drawable/account"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toBottomOf="@+id/divider">

<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/nombre_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:padding="@dimen/edit_text_padding"
    android:singleLine="true"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/text_input_layout_contra"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/normal_padding"
    android:layout_marginBottom="@dimen/padding_between"
    android:hint="Contraseña"
    app:boxCornerRadiusTopEnd="10dp"
    app:boxCornerRadiusTopStart="10dp"
    app:startIconDrawable="@drawable/outline_vpn_key_24"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/text_input_layout_usuario">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/contra_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:padding="@dimen/edit_text_padding"
        android:singleLine="true"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.checkbox.MaterialCheckBox
    android:id="@+id/recordar_check"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/padding_between"
    android:checked="false"
    android:text="@string/recordar_check"
    app:layout_constraintLeft_toLeftOf="@+id/text_input_layout_contra"
    app:layout_constraintStart_toStartOf="@+id/text_input_layout_contra"
    app:layout_constraintTop_toBottomOf="@+id/text_input_layout_contra" />

<TextView

```

```

        android:id="@+id/olvidar_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/padding_between"
        android:text="@string/olvidar_text"
        android:textAppearance="?android:attr/textAppearanceSmall"
        app:layout_constraintEnd_toEndOf="@+id/text_input_layout_contra"
        app:layout_constraintRight_toRightOf="@+id/text_input_layout_contra"
        app:layout_constraintTop_toBottomOf="@+id/recordar_check" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/entrar_boton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="@dimen/button_padding"
    android:paddingBottom="@dimen/button_padding"
    android:text="@string/entrar_boton"
    android:textColor="@android:color/white"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/olvidar_text" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Solamente nos falta analizar el código de nuestra aplicación, donde iremos visualizando los elementos de la vista y las acciones de cada uno de los botones.

```

1  class MainActivity : AppCompatActivity() {
2      private lateinit var binding: ActivityMainBinding
3      private var dialog : AlertDialog? = null
4
5      override fun onCreate(savedInstanceState: Bundle?) {
6          super.onCreate(savedInstanceState)
7          binding = ActivityMainBinding.inflate(layoutInflater)
8          val view = binding.root
9          setContentView(view)
10
11         binding.button1.setOnClickListener( ) {
12             val dialogBuilder = createLoginDialogo()
13             dialog = dialogBuilder.show()
14         }
15     }
16
17     fun createLoginDialogo(): MaterialAlertDialogBuilder {
18         val builder = MaterialAlertDialogBuilder(this)
19         val view = this.layoutInflater.inflate(R.layout.dialogo, null)
20         // val view = LayoutInflater.from(this).inflate(R.layout.dialogo, null)
21         builder.setView(view)
22
23         val crear = view.findViewById<View>(R.id.crear_boton) as Button
24         val entrar = view.findViewById<View>(R.id.entrar_boton) as Button
25         crear.setOnClickListener {
26             Toast.makeText(applicationContext, "Pulsaste CREAR", Toast.LENGTH_SHORT).show()
27         }
28         entrar.setOnClickListener {
29             Toast.makeText(applicationContext, "Pulsaste ENTRAR", Toast.LENGTH_SHORT).show()
30             dialog?.dismiss() // Cerrará el diálogo
31         }
32         return builder
33     }
34 }

```

Aclaraciones:

- Líneas 18-21:** la idea es que tenemos que superponer la vista de nuestro diálogo sobre la vista de nuestra actividad, es decir, inflar la vista. Para ello utilizamos un objeto **inflater** que hace referencia a la vista principal (su xml) con **this@MainActivity.layoutInflater** . Posteriormente añadimos o inflamos sobre esa vista el diseño de nuestro diálogo, obteniendo una nueva jeraquía de vista (**línea 19**). Finalmente esa vista se asigna al diálogo que estamos construyendo (**línea 21**).
- Líneas 23-32:** referenciamos las vistas del diálogo que queremos tratar y escribimos los escuchadores correspondientes.
- Líneas 30:** el botón '**ENTRAR**' además de realizar la acción, cerrará el dialogo con **dialog?.dismiss()** . Siendo **dialog** el objeto **AlertDialog** que crearemos

posterioroemente en el **show()** con el '*builder*'.