

Bloque 3. Exercici Resolt Dos Activitys

[Descarregar aquests apunts](#)

Crearem un nou projecte que es dirà EjercicioResuelto2Activitys. Com a requisits d'aquest: volem que s'execute en manera apaïxada i que no l'afecte el canvi d'orientació del dispositiu. La versió mínima suportada serà la 19. L'aplicació constarà de dues activitats, en la primera l'usuari introduirà el seu nom en un EditText i quan l'usuari preme el botó Salutació, s'obrirà una altra finestra on saludarem de forma personalitzada a l'usuari.

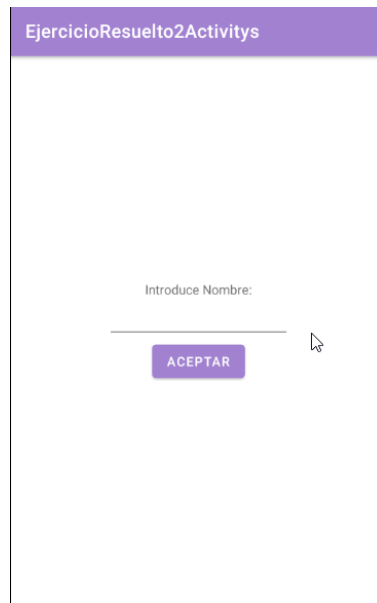
PAS1. Crearem el projecte seguint els passos ja realitzats en els exercicis anteriors

PAS2. Passem a definir el layout de la activity principal

Per a això obri l'arxiu **activity_main.xml** i còpies el següent codi, en temes posteriors es descriuran els elements que s'usen:

```
<?xml version="1.0" encoding="utf-8"?>
<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:gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Introduce Nombre:"/>
    <EditText
        android:layout_gravity="center"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:id="@+id/datos" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:text="Saludo"/>
</LinearLayout>
```

El resultat visual de la primera finestra de la nostra aplicació seria alguna cosa semblança al següent:



PAS3. Passem a definir la interfície gràfica de la segona finestra.

Per a això creem un nou fitxer anomenat activity2.xml, i en ell afegim únicament una etiqueta TextView on mostrarem el missatge personalitzat a l'usuari.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textoActivity2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="132dp"
        android:layout_marginTop="320dp"
        android:text="textoActivity2"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

PAS4. Crear una activity per a la pantalla secundària.

Crearem una classe nova derivada de AppCompatActivity. En ella implementarem el mètode onCreate, igual que es va fer per a l'anterior.

```

1 | class Activity2 : AppCompatActivity() {
2 |     override fun onCreate(savedInstanceState: Bundle?) {
3 |         super.onCreate(savedInstanceState)
4 |         setContentView(R.layout.activity2)
5 |         val textView = findViewById<TextView>(R.id.textoActivity2);
6 |         val intento = this.intent
7 |         textView.setText("Hola "+intento.getStringExtra("DATO")+ ", ¿como estas?")
8 |     }
9 | }

```

🚩 **Nota:** la línia 4 el que fa és carregar el layout que volem associar amb l'activitat principal i assignar-ho al context de l'aplicació, **Línia 5** el mètode `findViewById` cerca la vista en el layout del context i la seguidora (crear l'objecte del tipus corresponent associat a aqueixa vista, en aquest cas `TextView`). **Línia 6** recuperem la instància de intent que arriba a l'activitat. **Línia 7** extraïem la informació del intent mitjançant el get corresponent, en aquest cas `getStringExtra` passant com a paràmetre la clau que ho identifica, i s'afeg al `textView`.

👉 **Important:** recordeu que hem de declarar en el Manifest totes les activitats que s'afeg a la nostra aplicació, en el nostre cas:

```

1 | <application
2 |     android:allowBackup="true"
3 |     android:icon="@mipmap/ic_launcher"
4 |     android:label="@string/app_name"
5 |     android:roundIcon="@mipmap/ic_launcher_round"
6 |     android:supportsRtl="true"
7 |     android:screenOrientation="landscape"
8 |     android:configChanges="orientation|screenSize"
9 |     android:theme="@style/Theme.EjercicioResuelto2Activitys">
10 |     <activity android:name=".MainActivity">
11 |         <intent-filter>
12 |             <action android:name="android.intent.action.MAIN" />
13 |             <category android:name="android.intent.category.LAUNCHER" />
14 |         </intent-filter>
15 |     </activity>
16 |     <activity android:name=".Activity2"/>
17 | </application>

```

PAS5. Implementarem la lògica de funcionament de la nostra aplicació.

El codi és el següent, només s'expliquen els elements nous del codi.

```

1  class MainActivity: AppCompatActivity() {
2      override fun onCreate(savedInstanceState: Bundle?) {
3          super.onCreate(savedInstanceState)
4          setContentView(R.layout.activity_main)
5          findViewById<Button>(R.id.button).setOnClickListener {
6              val intento=Intent(applicationContext, Activity2::class.java)
7              val datos=findViewById<EditText>(R.id.datos).text.toString()
8              intento.putExtra("DATO", datos)
9              startActivity(intento)
10         }
11     }
12 }

```

📌 **Nota:** En la **línia 5** s'unfla el botó i amb el seu mètode listener de Clic afegim el codi que volem que ocorregi en prémer el botó. En la **Línia 6** es crea un intent redirigit a la Activity2. **Línia 7** s'extrau la informació, introduïda per l'usuari, del EditText. **Línia 8** s'afeg l'extra al intent usant la clau identificativa "DATO", per a posteriorment extraure aqueixa informació. **Línia 9** s'inicia l'activitat amb el intent creat.