

**Hoja_UT2_09****Intents explícitos (con valores de retorno)**

Utilizaremos la **App_UT2_05**

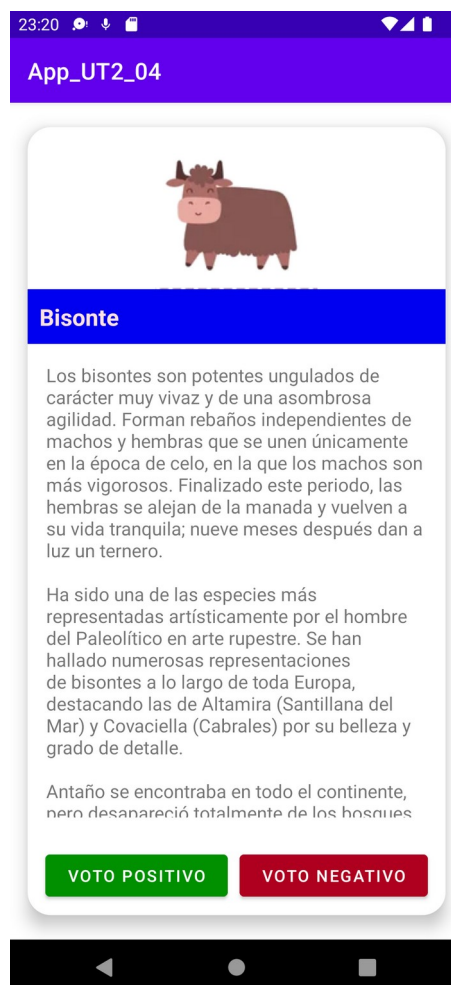
En esta tarea añadiremos un Splash Screen y realizaremos modificaciones para que se devuelva un valor desde DetalleAnimalActivity a MainActivity.

DISEÑO

Modificaremos el diseño del layout del ítem para añadir un contador de votos. También se añadirá un atributo votos de tipo Int en la clase Animal.



En el layout activity_detalle_animal.xml añadiremos 2 botones. El primero servirá para añadir un voto positivo del animal y el segundo para reflejar un voto negativo.





Hoja_UT2_09

RESPUESTA AL PULSAR UN BOTÓN

Cuando pulsemos un botón tenemos que volver a MainActivity pasando 1 si se ha presionado el botón de voto positivo y -1 si ha sido el botón de voto negativo.

```
fun crearIntentAMain(animal: Animal, voto: Int)
{
    val intentDevuelto = Intent()
    intentDevuelto.putExtra("voto", voto)
    intentDevuelto.putExtra("animal", animal.nombre)
    setResult(RESULT_OK, intentDevuelto)
    finish()
}
```

```
binding.botonPositivo.setOnClickListener {
    crearIntentAMain(animal, 1)
}

binding.botonNegativo.setOnClickListener {
    crearIntentAMain(animal, -1)
}
```

Después, en la MainActivity crearemos un launcher para recibir la información devuelta por la activity de detalle:

```
private val segundaActivityLauncher =
    registerForActivityResult(ActivityResultContracts.StartActivityForResult())
    {
        if(it.resultCode == Activity.RESULT_OK){
            val voto = it.data?.getIntExtra("voto", 0)?:0
            val nombre = it.data?.getStringExtra("animal")?:""
            (recyclerView.adapter as AnimalAdapter).cambiarVoto(nombre, voto)
            recyclerView.adapter?.notifyDataSetChanged()
        }
    }
}
```

Recibimos, el voto y el nombre del animal y cambiamos el voto de este animal y le decimos al adaptador que ha habido cambios.

En la clase AnimalAdapter tendremos que crear el método cambiarVoto que modifique el número de votos (+1 o -1) de el animal cuyo nombre se pasa.



Hoja_UT2_09

SPLASH SCREEN

Crea una nueva Empty Activity llamada **SplashScreen**.

Añade la imagen de la pantalla de bienvenida y modifica el layout **activity_splash_screen.xml** para meter un imageView centrado que muestre la imagen importada.



Modifica el archivo **AndroidManifest.xml** para que SplashScreen sea la actividad que se lance. Además, le pondremos el tema **Theme.AppCompat.Light.NoActionBar** para que no se muestre la barra de navegación superior

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="es.ivanlorenzo.app_ut2_05">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
```

**Hoja_UT2_09**

```
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.App_UT2_04"
tools:targetApi="31">
<activity
    android:name=".SplashScreen"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar"
    android:exported="true" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".DetalleAnimalActivity"
    android:exported="false" />
<activity
    android:name=".MainActivity"
    android:exported="false">
</activity>
</application>

</manifest>
```

Si ejecutamos nos mostrará la SplashScreen pero no pasaremos de ahí. Lo que haremos será esperar 5 segundos y redirigir a MainActivity. Para ello crearemos un objeto Handler que ejecute un intent explícito a los 5 segundos:

```
class SplashScreen : AppCompatActivity()
{
    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash_screen)

        Handler(Looper.getMainLooper()).postDelayed({
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
            finish()
        }, 5000)
    }
}
```