

# INTENTS EXPLICITOS

## 1. Graddle app

```
buildFeatures {  
    viewBinding = true  
}
```

## 2. MainActivity o SecondActivity o lo que sea

```
private lateinit var binding : ActivityMainBinding  
  
//En caso de SecondActivity  
//private lateinit var binding: ActivitySecondBinding  
  
binding = ActivityMainBinding.inflate(layoutInflater)  
setContentView(binding.root)
```

## 3. Creamos XML (los que sean)

## 4. AndroidManifest.xml

```
<activity  
    android:name=".SecondActivity"  
    android:label="Second Activity"  
    android:parentActivityName=".MainActivity"  
    android:exported="false" />
```

label : cambia el titulo

parent : le indica de quien es hija

## 5. Enviar Datos - MainActivity y recibir

a) Companion Object → En las variables de la clase

```
companion object {  
    const val TAG_APP = "myExplicitIntent"  
    const val EXTRA_TITULO = "myTitulo"  
    const val EXTRA_AUTOR = "myAutor"  
}
```

b) Intent.putExtra en → Creamos un listener contante de INTENT y metemos los datos y los lanzamos con startActivity()

```
binding.btEnviar.setOnClickListener() {  
    lanzarSegundaActividad()  
}
```

//Creamos por un lado la funcion de lanzarSegundaActividad y por otro el getResult para recoger los datos de la segunda actividad

```
// -- Enviar y recibe en parte con getResult  
private fun lanzarSegundaActividad() {  
    // Se crea un objeto de tipo Intent  
    val myIntent = Intent(this, SecondActivity::class.java).apply {  
        // Se añade la información a pasar por clave-valor  
        putExtra(EXTRA_AUTOR, binding.editAutor.text.toString())  
        putExtra(EXTRA_TITULO, binding.editTitulo.text.toString())  
    }  
    getResult.launch(myIntent)  
}
```

// Se crea la constante de getResult que es un Registro de ActivityResult que nos da los mecanismos a partir de it para discernir que actividad nos llama  
// -- Recibe

```
private val getResult = registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {  
    if (it.resultCode == Activity.RESULT_OK) {  
        binding.txtResult.text = "Se ha aceptado."  
        binding.txtResult.visibility = View.VISIBLE  
    }  
    if (it.resultCode == Activity.RESULT_CANCELED)  
    {  
        binding.txtResult.text = "Se ha cancelado"  
        binding.txtResult.visibility = View.VISIBLE  
    }  
}
```

## 6 . Recibir Datos del Main - Second

Funcion onCreate

```
val mytitulo = intent.getStringExtra(MainActivity.EXTRA_TITULO)
val myautor = intent.getStringExtra(MainActivity.EXTRA_AUTOR)

//binding.txtDatos.text = "Titulo: " + mytitutlo + "\nAutor: " + myautor
binding.txtDatos.text = "Titulo: $mytitulo \nAutor: $myautor"
```

## 7. Devolver datos desde Second - Main

Lo primero seria crera los EXTRA en Companion Objects para que existieran

```
companion object {
    const val TAG_APP = "myExplicitIntent"
    const val EXTRA_TITULO = "myTitulo"
    const val EXTRA_AUTOR = "myAutor"
    const val EXTRA_COMENTARIO = "myComentario"
    const val EXTRA_RATING = "myRating"
}
```

Ahora vamos al SecondActivity modificamos el listener con los datos que queremos mete puExtra en el Main, donde declaramos un Intent y le aplicamos los cambios pasando el inetnt al Main mediante setResult

```
binding.btAceptar.setOnClickListener() {
    val intentResult :Intent = Intent().apply {
        //Metemos los datos en el MainActivity
        putExtra(MainActivity.EXTRA_COMENTARIO, binding.editOpinion.text.toString())
        putExtra(MainActivity.EXTRA_RATING, binding.ratingBar.rating)
    }

    Log.d(MainActivity.TAG_APP, "Pulsado boton Aceptar")
    //Importatnte meter el intentResult para pasarlo a la MainActivity
    setResult(Activity.RESULT_OK, intentResult)
    finish()
}
```

## 8. Recoger Datos en Main

En el lugar que queremos recoger los datos dentro del getResult

```
if (it.resultCode == Activity.RESULT_OK) {
    val myComentario = it.data?.getStringExtra(EXTRA_COMENTARIO)
    val myRating = it.data?.getFloatExtra(EXTRA_RATING,0.0f).toString()

    binding.txtResult.text = "Valoracion del Usuario: \nRating = $myRating \nComentario:\n $myComentario "
    binding.txtResult.visibility = View.VISIBLE
}
```

## CODIGO MAIN

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding

    //Creamos un companion object para pasarle todos los datos a la segunda actividad
    // Serian tres variables 1 el tag para controlar el Log cat y las otras dos para los dos
    // campos, tener en cuenta que si pusieramos tipo string despues tendriamos muchos problemas
    // De aqui nos vamos al interior d la funcion onCreate para pasarle los datos con el intent
    companion object {
        const val TAG_APP = "myExplicitIntent"
        const val EXTRA_TITULO = "myTitulo"
        const val EXTRA_AUTOR = "myAutor"
        const val EXTRA_COMENTARIO = "myComentario"
        const val EXTRA_RATING = "myRating"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btEnviar.setOnClickListener() {
            lanzarSegundaActividad()
        }
    }

    private fun lanzarSegundaActividad() {
        // Se crea un objeto de tipo Intent
        val myIntent = Intent(this, SecondActivity::class.java).apply {
            // Se añade la información a pasar por clave-valor
            putExtra(EXTRA_AUTOR, binding.editAutor.text.toString())
            putExtra(EXTRA_TITULO, binding.editTitulo.text.toString())
        }
        getResult.launch(myIntent)
    }

    private val getResult = registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
        if (it.resultCode == Activity.RESULT_OK) {
            val myComentario = it.data?.getStringExtra(EXTRA_COMENTARIO)
            val myRating = it.data?.getFloatExtra(EXTRA_RATING, 0.0f).toString()

            binding.txtResult.text = "Valoracion del Usuario: \nRating = $myRating \nComentario:\n $myComentario "
            binding.txtResult.visibility = View.VISIBLE
        }
        if (it.resultCode == Activity.RESULT_CANCELED) {
            binding.txtResult.text = "Se ha cancelado"
            binding.txtResult.visibility = View.VISIBLE
        }
    }
}
```

## CODIGO SECOND

```
class SecondActivity : AppCompatActivity() {
    private lateinit var binding : ActivitySecondBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivitySecondBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val mytitulo = intent.getStringExtra(MainActivity.EXTRA_TITULO)
        val myautor = intent.getStringExtra(MainActivity.EXTRA_AUTOR)
        //binding.txtDatos.text = "Titulo: " + mytitutlo + "\nAutor: " + myautor
        binding.txtDatos.text = "Titulo: $mytitulo \nAutor: $myautor"

        // Creamos los listeners
        //Importante ha de llevar el finish para decirle que efectivamente se finaliza actividad
        binding.btAceptar.setOnClickListener() {
            val intentResult :Intent = Intent().apply {
                //Metemos los datos en el MainActivity
                putExtra(MainActivity.EXTRA_COMENTARIO, binding.editOpinion.text.toString())
                putExtra(MainActivity.EXTRA_RATING, binding.ratingBar.rating)
            }

            Log.d(MainActivity.TAG_APP, "Pulsado boton Aceptar")
            setResult(Activity.RESULT_OK, intentResult)
            finish()
        }
        binding.btCancelar.setOnClickListener() {
            Log.d(MainActivity.TAG_APP, "Se ha pulsado Cancelar")
            setResult(Activity.RESULT_CANCELED)
            finish()
        }
    }
}
```