

## Guía de ejercicios - Ciclo de vida de componentes (II)



¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

### ¿En qué consiste esta guía?

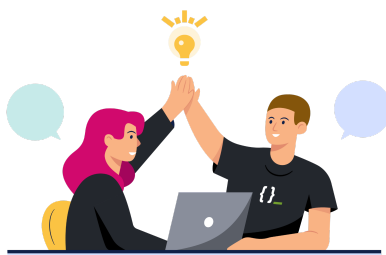
La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase.

**¡Vamos con todo!**



### Tabla de contenidos

Actividad guiada N°: Intents y Bundle	2
¡Manos a la obra! - Usar Intent para abrir el mapa	3
¡Manos a la obra! - User intent para compartir contenido	4
Solución - Usar Intent para abrir el mapa	5
Solución - User intent para compartir contenido	6
Preguntas de proceso	7
Preguntas de cierre	7
Referencias bibliográficas	7



**¡Comencemos!**



## Actividad guiada N°: Intents y Bundle

En la siguiente actividad guiada enviaremos un dato desde un Activity a otro Activity, haciendo uso de Bundle y enviando un objeto parcelable.

¡Comencemos! Primero debemos crear un proyecto con un Empty Activity. Luego debemos editar nuestro build.gradle en el módulo App y agregar el siguiente plugin, el cual nos ayudará a trabajar con objetos parcelables de forma sencilla.

```
id("kotlin-parcelize")
```

Ahora crearemos el objeto que pasaremos entre activities

```
@Parcelize
data class User(
    val id: Int,
    val name: String,
    val age: Int,
) : Parcelable
```

Creamos una instancia de este objeto User, por ejemplo:

```
val user = User(5, "Samuel", 42)
```

Ya que intent no reconoce este tipo de objeto, lo que debemos hacer es, en cierto modo, empaquetarlo con un Bundle. La manera de hacerlo es la siguiente:

```
val bundle = Bundle().apply {
    putParcelable("USER_KEY", user)
}
```

Ahora solo tenemos que iniciar nuestra activity e indicar que pasaremos un bundle en su intent.

```
startActivity(Intent(this, SecondActivity::class.java).apply {
    putExtra("BUNDLE_KEY", bundle)
}))
```

Lo siguiente es leer el dato que estamos enviando en el siguiente Activity, el cual llamamos SecondActivity.

La forma de leer este dato puede ser de la siguiente:

```
val bundle = intent.extras?.get("BUNDLE_KEY") as Bundle
val user = bundle.getParcelable<User>("USER_KEY")
```

O de forma más corta:

```
val user = (intent.extras?.get("BUNDLE_KEY") as
Bundle).getParcelable<User>("USER_KEY")
```



## ¡Manos a la obra! - Usar Intent para abrir el mapa

En la siguiente actividad aprenderemos a usar Intent para abrir otra aplicación y pasar algunos datos, para esto crea un proyecto con un Empty Activity.

1. Usaremos un intent para abrir Google Maps en una ubicación específica, por lo que debemos pasar el String "geo:51.17885270841654,-1.8261525945670793?z=20" a tipo Uri
2. Crea un intent tipo ActionView y pásale el string transformado a tipo Uri.
3. Especifica qué aplicación abrirá el intent, hazlo mediante setPackage y usando el nombre de paquete "com.google.android.apps.maps"
4. Valida que exista el paquete que estamos tratando de llamar (Google Maps).
5. Inicia el activity.
6. Compara el código final.
7. Repetimos los pasos anteriores, pero esta vez parseamos el siguiente string: "google.streetview:cbll=51.17885270841654,-1.8261525945670793"



## ¡Manos a la obra! - User intent para compartir contenido

En este ejercicio aprenderemos cómo compartir un texto desde nuestra app haciendo uso de Intent y de las opciones que Android ofrece. Para esto deberás crear un proyecto con un Empty Activity y luego completar los siguientes pasos:

1. Crea una variable tipo Intent y llámala intent.
2. Define el tipo de Action como Action Send.
3. Agregale un extra tipo texto con Extra Text.
4. Especifica el MIME Type con intent.type
5. Utiliza el Chooser de Android, el cual permite compartir con otras apps dependiendo del tipo de contenido que estás compartiendo, para esto, define una variable Intent.createChooser y llámala shareIntent.
6. Inicia el Activity.
7. Compara el código final.
8. Transformemos ese código a un estilo más Kotlin haciendo uso de "apply".

## Solución - Usar Intent para abrir el mapa

1.

```
val gmmIntentUri =  
Uri.parse("geo:51.17885270841654,-1.8261525945670793")
```

2.

```
val mapIntent = Intent(Intent.ACTION_VIEW, gmmIntentUri)
```

3.

```
mapIntent.setPackage("com.google.android.apps.maps")
```

4.

```
mapIntent.resolveActivity(packageManager)?.let {  
  
}
```

5.

```
startActivity(mapIntent)
```

6.

```
val gmmIntentUri =  
Uri.parse("geo:51.17885270841654,-1.8261525945670793")  
    val mapIntent = Intent(Intent.ACTION_VIEW, gmmIntentUri)  
    mapIntent.setPackage("com.google.android.apps.maps")  
    mapIntent.resolveActivity(packageManager)?.let {  
        startActivity(mapIntent)  
    }
```

7.

```
val gmmIntentUri =  
Uri.parse("google.streetview:cbll=51.17885270841654,-1.8261525945670793"  
)  
val mapIntent = Intent(Intent.ACTION_VIEW, gmmIntentUri)  
mapIntent.setPackage("com.google.android.apps.maps")  
mapIntent.resolveActivity(packageManager)?.let {  
    startActivity(mapIntent)  
}
```

## Solución - User intent para compartir contenido

1.

```
val intent = Intent()
```

2.

```
intent.action = Intent.ACTION_SEND
```

3.

```
intent.putExtra(Intent.EXTRA_TEXT, "Hola Android!!!")
```

4.

```
intent.type = "text/plain"
```

5.

```
val shareIntent = Intent.createChooser(intent, "Compartiendo ... ")
```

6.

```
startActivity(shareIntent)
```

7.

```
val intent = Intent()
intent.action = Intent.ACTION_SEND
intent.putExtra(Intent.EXTRA_TEXT, "Hola Android!!!")
intent.type = "text/plain"

val shareIntent = Intent.createChooser(intent, "Compartiendo ... ")
startActivity(shareIntent)
```

8.

```
val shareIntent = Intent.createChooser(
    Intent().apply {
        action = Intent.ACTION_SEND
        putExtra(Intent.EXTRA_TEXT, "Hola Android!!!")
        type = "text/plain"
    }, "Compartiendo ... "
)
startActivity(shareIntent)
```

## Preguntas de proceso

### Reflexiona:

- A partir de lo aprendido hasta aquí ¿Existe algún ejercicio de los resueltos previamente que deba repetir o desarrollar otra vez para poder dominarlo mejor?
- ¿Qué contenido se me ha hecho más difícil? ¿Y cuál más sencillo?
- ¿Crees que lo aprendido será de utilidad a futuro? Nombra al menos 1 uso.



## Preguntas de cierre

- Respecto a Intents, cuando se comparte un texto, el MIME Type es `text/plain`, ¿cual seria el MIME Type si quiero compartir una imagen?
- ¿A simple vista, te parece que el uso de Intents y Bundle, soluciona el envío de datos dentro de una aplicación?
- ¿Qué método utilizarías si tuvieras que enviar un arreglo de User entre activities?
- ¿Qué pasa si envías un dato tipo String desde el Activity A y en el Activity B lo tratas de leer como tipo Boolean?

## Referencias bibliográficas

<https://developer.android.com/docs>