

5. Navegación I

Alberto Luengo



Manifest y Permisos

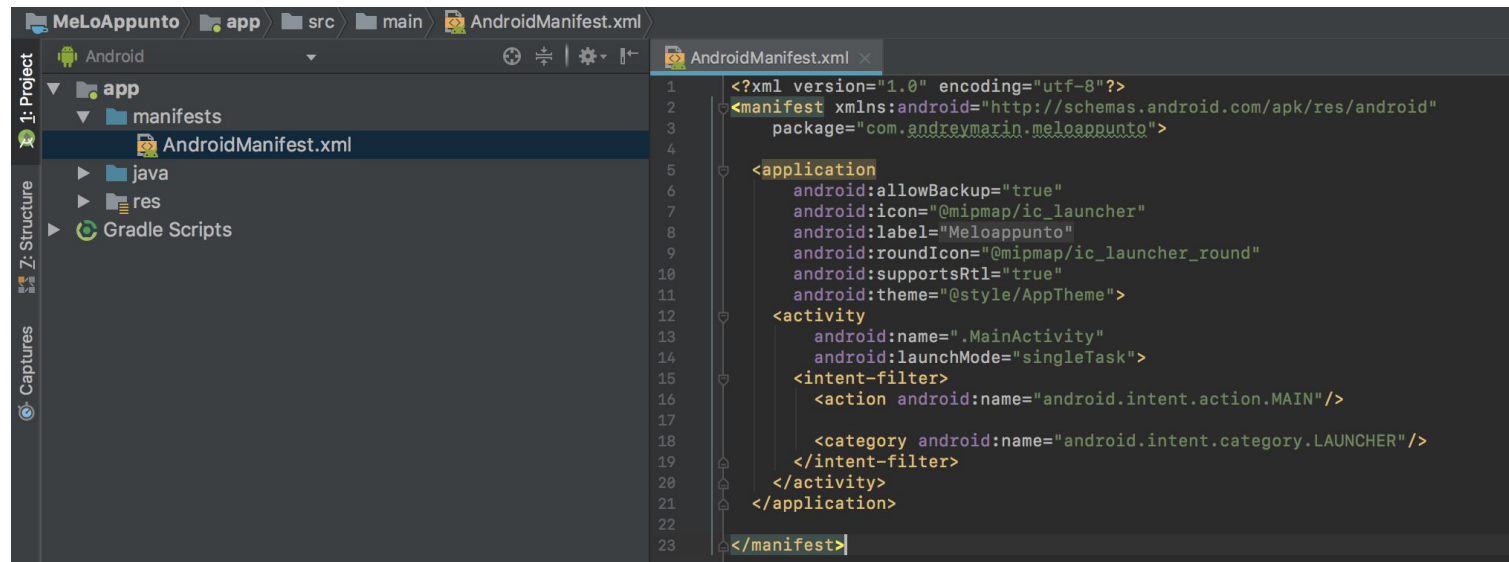
El archivo de manifiesto proporciona información esencial sobre tu aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la app.

Entre otras cosas, el archivo de manifiesto hace lo siguiente:

- Nombra el paquete que sirve como un **identificador único para la aplicación**.
- Describe los componentes de la aplicación, como las **actividades**, los **servicios**, los receptores de mensajes y los proveedores de contenido que la integran.
- Declara los **permisos** que debe tener la aplicación para acceder a las partes protegidas de una API e interactuar con otras aplicaciones. También declara los permisos que otros deben tener para interactuar con los componentes de la aplicación.

Toda la información del manifiesto:

<https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es-419>



Activitys & Intents

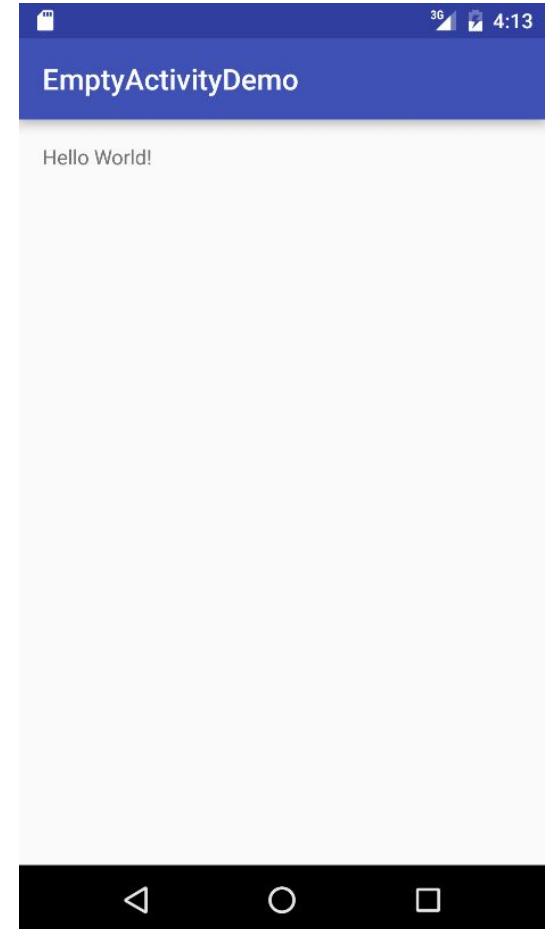
Cuando creas una aplicación de Android, lo primero que debes hacer es crear una actividad . Las actividades son donde sucede toda la acción, porque son las pantallas que permiten al usuario interactuar con su aplicación.

Las actividades son uno de los **componentes básicos de una aplicación de Android**. Lo más importante que tienes que aprender sobre las actividades es:

1. El proceso para crear, iniciar y detener una actividad y cómo manejar la navegación entre actividades.
2. Las diversas etapas en el ciclo de vida de una actividad y cómo manejar cada etapa..
3. La forma de administrar los cambios de configuración y persistir los datos dentro de su actividad.

Tutorial para aprenderlo todo sobre las activities:

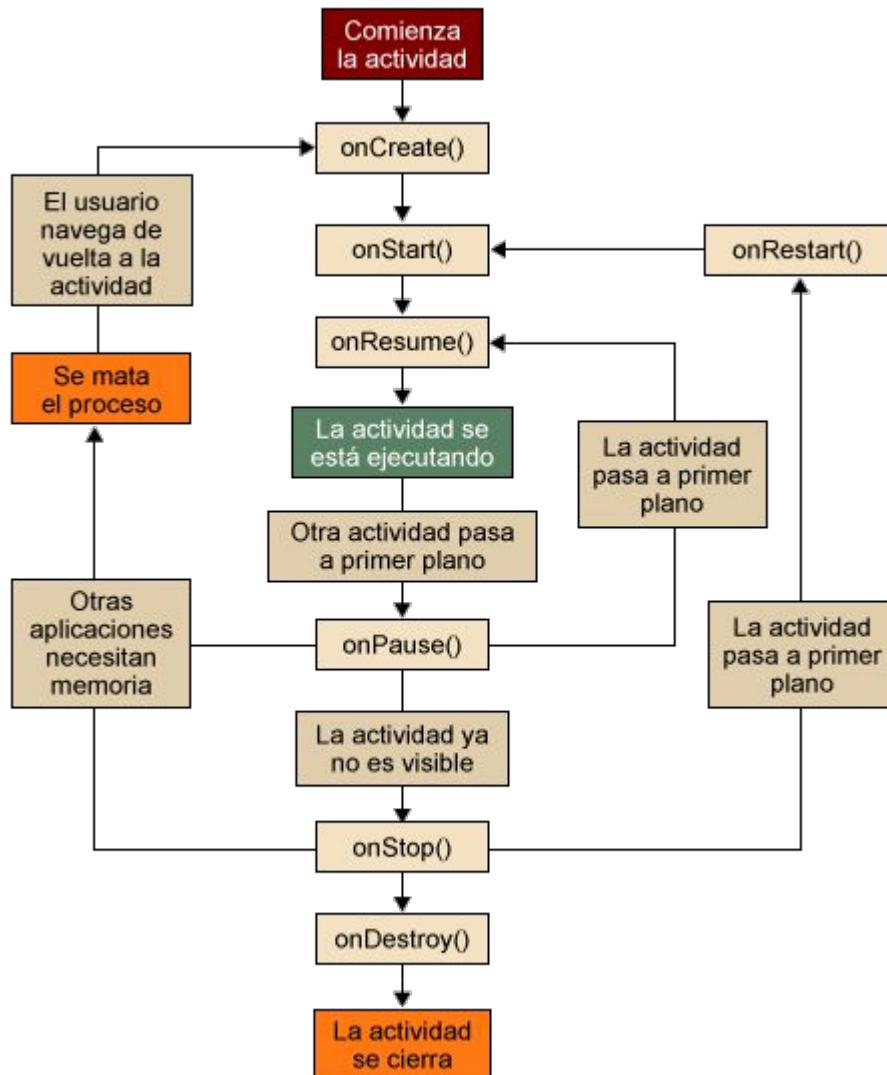
<https://www.raywenderlich.com/165824/introduction-android-activities-kotlin>



Activitys & Intents

- **onCreate ()** : cuando se crea la actividad por primera vez, aquí es donde se inicializan los elementos de la interfaz de usuario (layouts) u objetos de datos.
- **onStart ()** : justo antes de presentar al usuario una actividad, se llama a este método. Siempre es seguido por `onResume()`. Por lo general, debe iniciar animaciones de IU, audio, etc.
- **onResume ()** : cuando una actividad entra en primer plano, se llama a este método.
- **onPause ()** : se llama a este método antes de pasar al fondo. Aquí debe detener cualquier imagen o audio asociado con la actividad, como las animaciones de la interfaz de usuario..
- **onStop ()** : Este método se llama inmediatamente después `onPause()`, cuando la actividad ya no es visible para el usuario, y es un buen lugar para guardar los datos.
- **onRestart ()** : se llama después de detener una actividad, pero justo antes de comenzar de nuevo. Siempre es seguido por `onStart()`.
- **onDestroy ()** : esta es la llamada final que recibirá del sistema operativo antes de que se destruya la actividad.

Activities & Intents



Las solicitudes para lanzar nuevas actividades, o recuperar las que no son visibles, se llaman [Intents](#) (intentos, solicitudes). Un *Intent* es el responsable de que le lance la primera actividad de una aplicación, y nuevos intentos serán responsables de ir lanzando las nuevas pantallas que sean necesarias.

Cada *Activity* puede encontrarse en diferentes estados:

- Activa o en ejecución cuando ocupa el primer plano. La pantalla está visible y tiene el foco de la interacción del usuario.
- Pausada, cuando ha perdido el foco pero todavía es parcialmente visible. Típicamente ocurre cuando se abre un diálogo encima de la pantalla.
- Parada o detenida, cuando nada de ella es visible.

El sistema puede terminar una *Activity* en cualquier momento, bien invocando su método [finish\(\)](#) o matando su proceso. Pero cuando escaseen los recursos y haya que terminar alguna *Activity*, acabará primero con aquellas que están paradas, luego con las pausadas y, ya en una situación crítica, con aquellas activas.

En cada caso cuando una *Activity* termina debe guardar su estado para que cuando sea lanzada lo pueda recuperar. Cuando se realizan cambios de estado la plataforma invoca determinados métodos de la actividad para que esta pueda realizar las operaciones oportunas.

Navegación



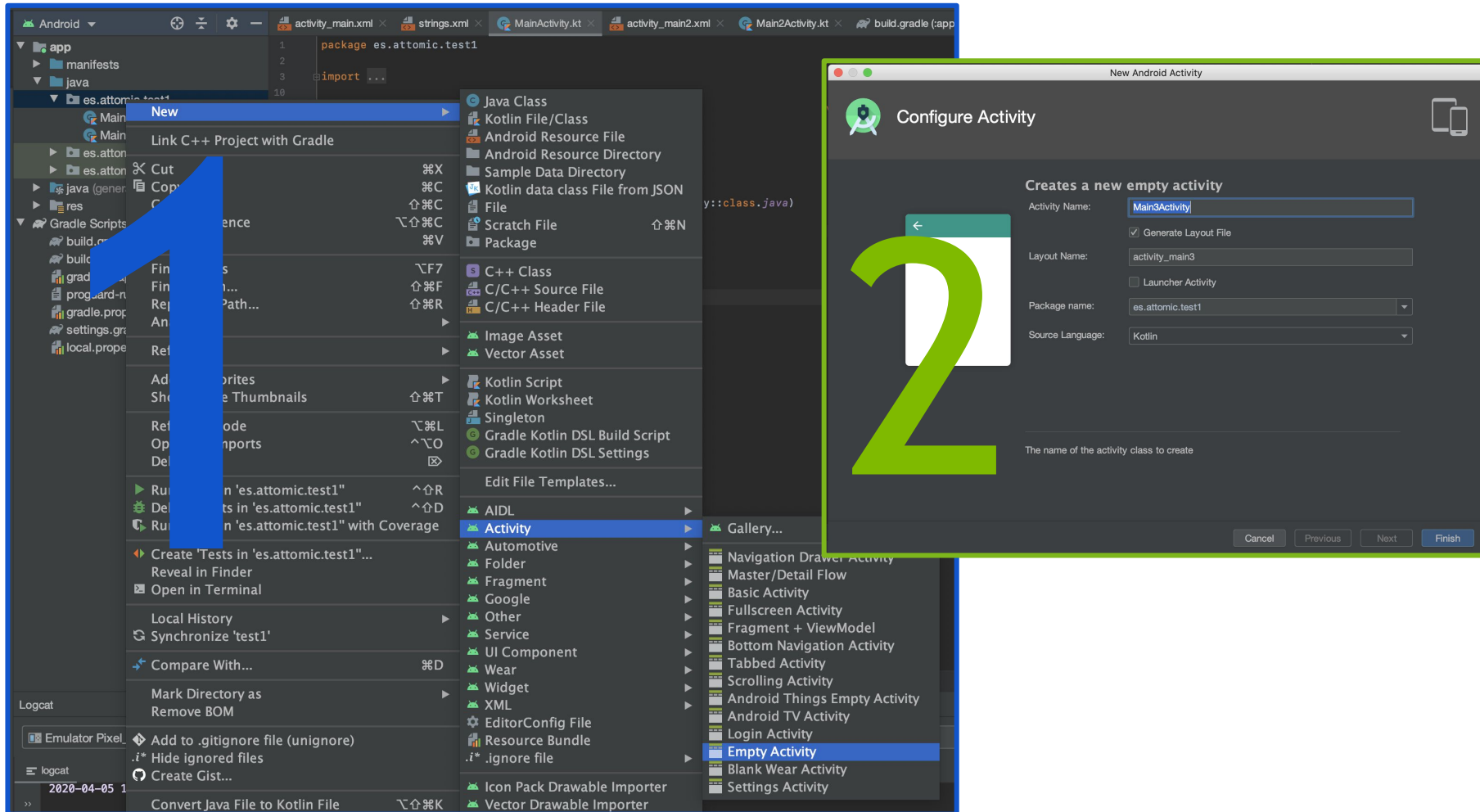
Navegación

Uno de los primeros pasos para diseñar y desarrollar una aplicación de Android es ***determinar lo que los usuarios pueden ver y hacer con la aplicación.***

Una vez que sepa con qué tipos de datos están interactuando los usuarios en la aplicación, el siguiente paso es diseñar las interacciones que les permitan a los usuarios navegar, entrar y salir de las diferentes partes del contenido dentro de la aplicación.

Navegación con Activitys

¿Cómo crear una Activity?



Navegación con Activitys

¿Cómo puedo abrir una activity desde otra activity?

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val intent = Intent(this, Main2Activity::class.java)  
        startActivity(intent)  
  
    }  
  
}
```

Navegación con Activitys

Hay varias formas de añadir el botón de atrás de una activity, la más sencilla y más rápida es la siguiente:

- 1) Crea una activity
- 2) En el manifest, añade lo siguiente en la activity en la que quieres que tenga el botón de atrás.

android:parentActivityName=".NombreDeLaActivityPadre"

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 package="es.attomic.tema3navegacionactivities">
4
5 <application
6     android:icon="@mipmap/ic_launcher"
7     android:label="Tema3NavegacionActivities"
8     android:roundIcon="@mipmap/ic_launcher_round"
9     android:supportsRtl="true"
10    android:theme="@style/AppTheme">
11    <activity android:name=".MainActivity">
12        <intent-filter>
13            <action android:name="android.intent.action.MAIN"/>
14
15            <!-- Para que google indexe nuestra app-->
16            <action android:name="android.intent.action.VIEW"/>
17            <category android:name="android.intent.category.DEFAULT"/>
18            <category android:name="android.intent.category.BROWSABLE"/>
19            <category android:name="android.intent.category.LAUNCHER"/>
20            <data android:host="attomic.es"
21                  android:scheme="attomic"/>
22            <!-- /Para que google indexe nuestra app-->
23
24        </intent-filter>
25    </activity>
26
27    <activity android:name=".ProfileActivity"
28              android:parentActivityName=".MainActivity">
29    </activity>
30
31    <activity android:name=".SettingsActivity"
32              android:parentActivityName=".ProfileActivity">
33    </activity>
34
35    </application>
36
37 </manifest>
```

Pasar datos

¿Cómo puedo enviar datos de una activity a otra?

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val intent = Intent(this, Main2Activity::class.java)  
        intent.putExtra("usuario", "Marin")  
        startActivity(intent)  
  
    }  
  
}
```

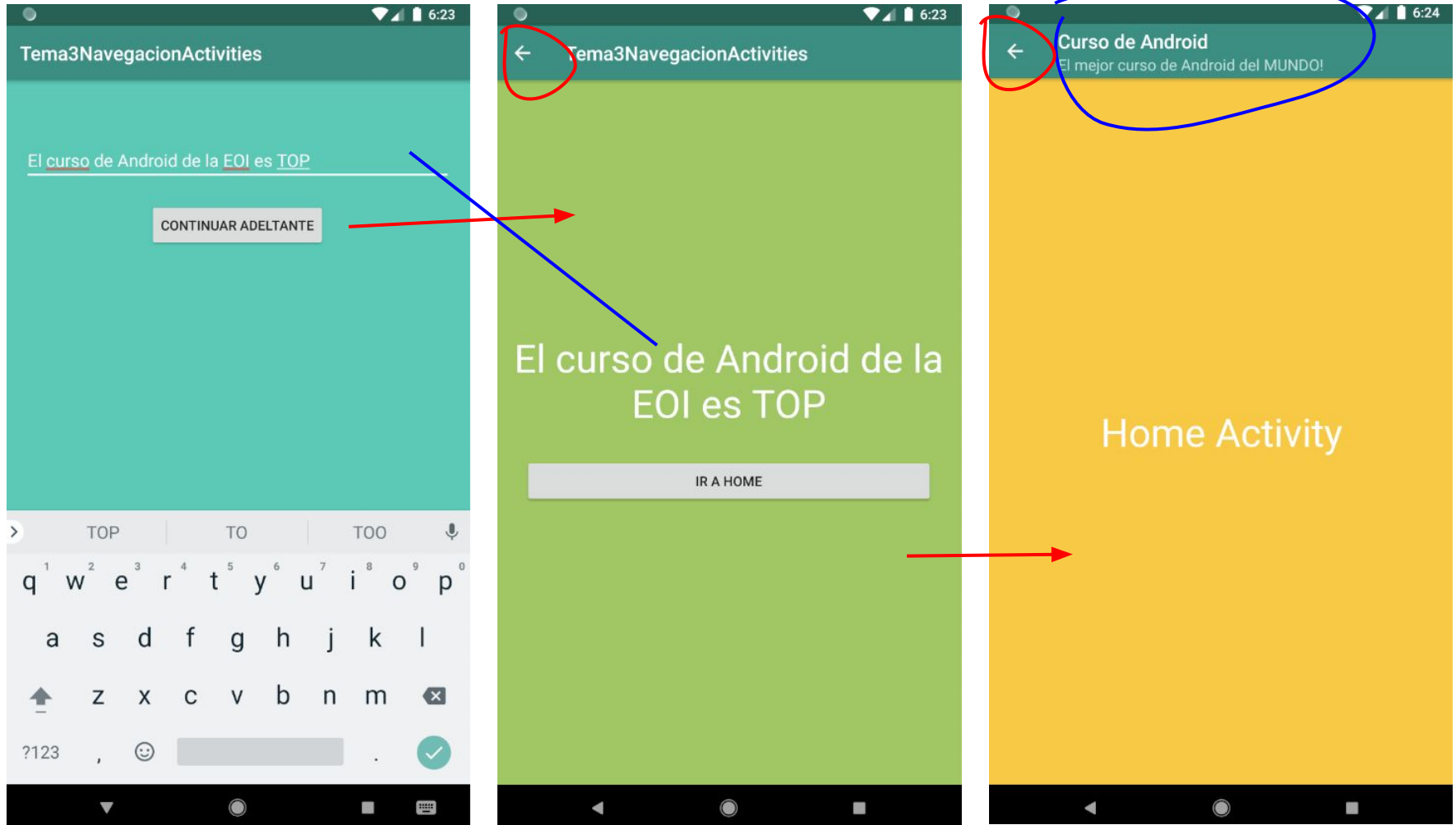
Recibir datos

¿Cómo puedo leer los datos que vienen de otra activity?

```
class Main2Activity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main2)  
  
        val bundle: Bundle? = intent.extras  
        val myUser: String? = bundle?.getString("usuario")  
  
        Log.v("miapp", "$myUser")  
  
    }  
}
```


Ejercicios

Ejercicios



Ejercicios

1. Hay que crear un proyecto nuevo, con el nombre de “TActivities”. En el proyecto, tienes que crear 2 Activity más, una que se llame “ProfileActivity” y otra “HomeActivity”.
2. En el activity_main.xml crea un EditText para introducir texto y un botón para que te lleve a la pantalla de “ProfileActivity”
3. En el xml de “ProfileActivity” crea un botón que te lleve a “HomeActivity”.
4. Añade el botón de volver atrás típico de Android en “ProfileActivity” y en “HomeActivity”.
5. Ahora, es el momento de utilizar el EditText creado antes, haz que el texto escrito en el campo de texto el activity_main.xml pase a la pantalla de “ProfileActivity” con el Bundle Extras de Android.
6. Después, en “ProfileActivity” obten ese texto con Bundle Extras.

Gracias.