Definiciones de Activity y Fragments

Vamos a definir en primer lugar estos conceptos:

Activity

De una forma resumida podríamos definirla como **cada una de las pantallas** que se crean en una **aplicación Android**. Para describir una Activity podemos identificar dos partes bien diferenciadas, la parte lógica y la parte gráfica.

- **La parte lógica** sería la parte programática o código trasero, que es una clase Java encargada de implementar las funcionalidades de la aplicación.
- **La parte gráfica** consiste en un archivo xml, situado en la carpeta*nombreproyecto/res/layout/archivo.xml*, dónde se definen los componentes gráficos que formarán la interfaz gráfica del usuario.

Parte Lógica de una Activity:

```
1 public class MainActivity extends Activity {
2
3    @Override
4    protected void onCreate(Bundle savedInstanceState) {
5        super.onCreate(savedInstanceState);
6        setContentView(R.layout.activity_main);
7    }
8 }
```

Parte Gráfica de una Activity:

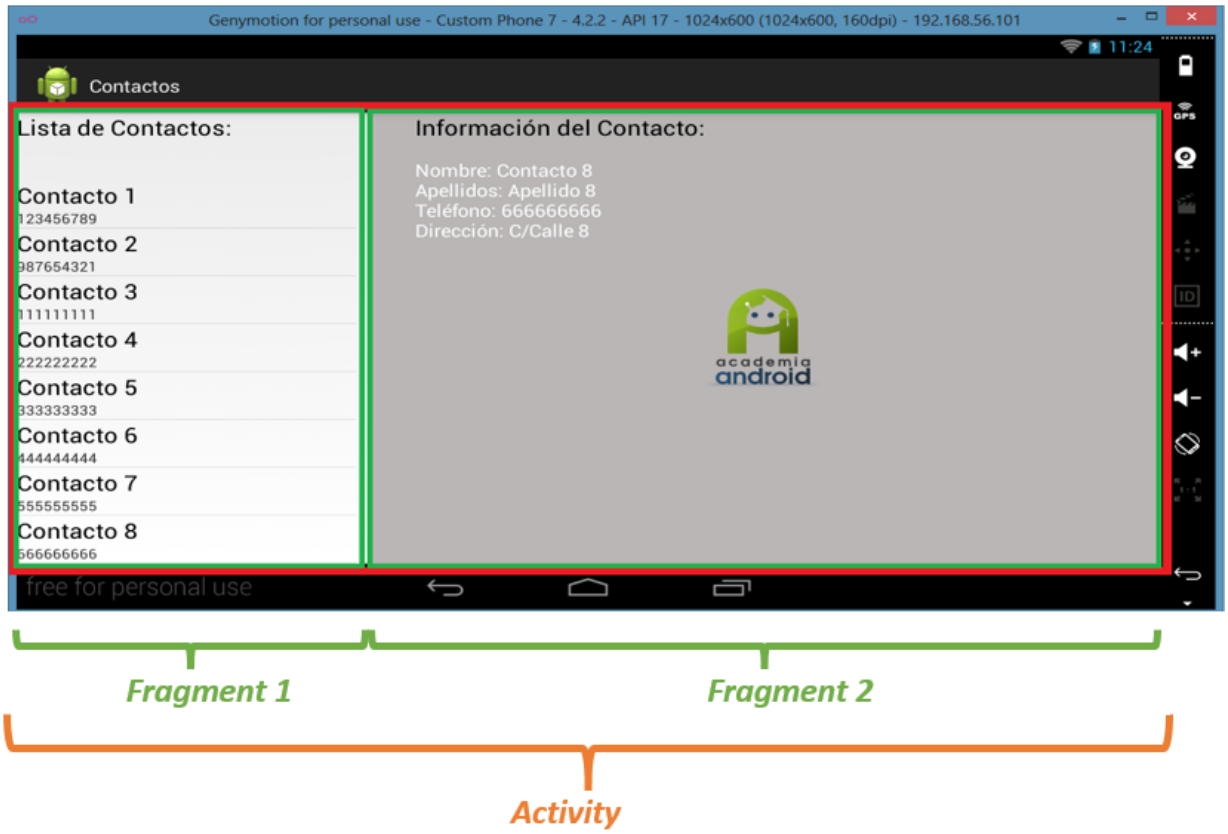
```
 1  <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
 2     xmlns:tools="http://schemas.android.com/tools"
 3     android:id="@+id/container"
 4     android:layout_width="match_parent"
 5     android:layout_height="match_parent"
 6     tools:context="com.academiaandroid.activity.MainActivity"
 7     tools:ignore="MergeRootFrame" >
 8
 9     <ImageView
10        android:id="@+id/imageView1"
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:src="@drawable/g4569" />
14
15 </FrameLayout>
```

Documentación Oficial Android Activity

Fragment

Son componentes que funcionan **dentro del ámbito de una Activity**. Su finalidad es la de ampliar parte de la lógica utilizada para la **navegación entre pantallas** o Activities, pudiendo definir **varios Fragments** dentro de una misma Activity, **interaccionando** entre ellos. Fueron introducidos con la versión de **Android 3.0** (API level 11), y representan el comportamiento de una porción de la interfaz de usuario asociada a una Activity.
Todo Fragment debe estar embebido dentro de una Activity, por lo que el ciclo de vida de un Fragment está ligado al de la Activity dónde se ha definido:

- Al pausar una Activity se pausarán todos los Fragments definidos dentro de esta.
- Si se destruye una Activty, todos los Fragments que contenga serán también destruidos.
- Si la Activity se encuentra en ejecución, es posible manipular de manera independiente cada Fragment, inclusive destruirlo.



** Imagen: una Activity con dos Fragments embebidos en ella*

Documentación Oficial Android Fragment:

Ciclo de Vida de un Fragment

Para poder **crear un Fragment** es necesario crear una subclase de Fragment. La **clase Fragment**, al igual que la clase Activity, contiene funciones de tipo **callback**, (cuando una función es pasada como argumento a otra función) como pueden ser las funciones **onCreate()**, **onStart()**, **onStop()** y **onPause()**.
Al menos, en todo ciclo de vida de un Fragment, se recomienda utilizar las siguientes funciones:

- onCreate() : El sistema llama a esta función cuando se crea el Fragment.
- onCreateView() : El sistema llama a esta función cuando se dibuja por primera vez el Fragment en la interfaz de usuario.
- onPause() : El sistema llama a esta función cuando el usuario deja de utilizar el Fragment (no implica que éste sea destruido).

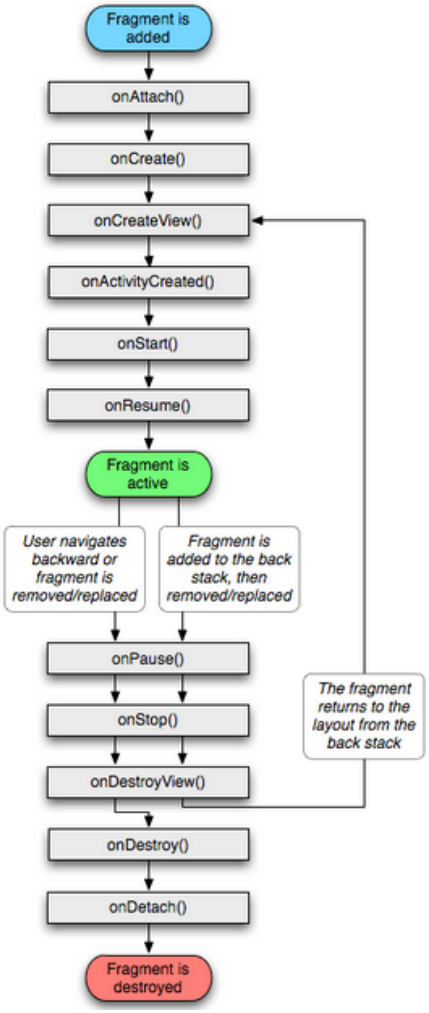


**Imagen: ciclo de vida de un Fragment mientras su Activity está en ejecución.*

La primera etapa del ciclo de vida de un fragment se corresponde con el instante en que el fragment se adjunta a la actividad que lo contiene (método onAttach). No puede existir un fragment sin estar vinculado a una actividad.

A continuación, el fragment se rige por un ciclo de creación que se corresponde con los siguientes métodos:

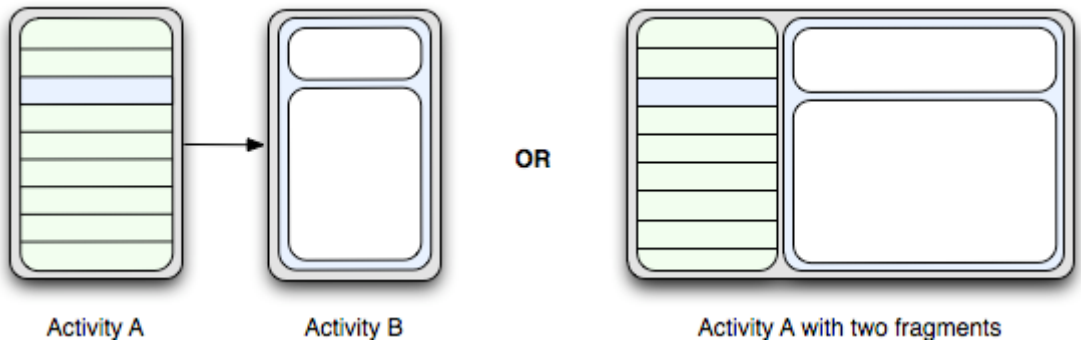
- **onCreate** : se invoca durante la creación del fragmento. Puede inicializar las variables esenciales para el funcionamiento de sus fragments y aquellas que desea salvaguardar cuando el fragment pase al estado de pausa y, a continuación, de nuevo al estado de ejecución.
- **onCreateView** (de implementación obligatoria): se invoca cuando el fragment dibuja su contenido. Para ello, debe cargar una vista y devolverla al final del método (clase view).
- **onActivityCreated**: se invoca cuando el fragment termine de crearse y de dibujarse (para, por ejemplo, implementar las interacciones con el usuario…).

El método **onStart** coincide con el paso del fragment **a primer plano**, seguido de la llamada al método **onResume** que indica que el fragment **ya puede interactuar con el usuario** (mismo funcionamiento que para una actividad - véase el capítulo Principios de programación - Ciclo de vida de una actividad).
Cuando un fragment pasa a estar inactivo, la llamada al método onPause permite ejecutar las acciones adecuadas (deshabilitar actualizaciones, listener...). Después de esta llamada, se invoca al método onStop (el fragmento deja de estar en primer plano).
Si se destruye un fragment, se invocará respectivamente a los siguientes métodos:

- **onDestroyView**: destrucción de la vista.
- **onDestroy**: destrucción del fragment.
- **onDetach**: el fragment se desvincula de la actividad que lo contiene.

Cambio de Filosofía

Tras la introducción de los Fragments en el S.O. Android (en la versión 3.0 como se ha comentado anteriormente), permitió la **división de una Activity en Fragments**, posibilitando la modificación de la apariencia de una Activity en tiempo de ejecución y preservando los cambios en la pila de procesos de la Activity:



**En esta imagen podemos apreciar como dos módulos de una interfaz de usuario dividida en dos Activities pueden combinarse en una sola Activity mediante el uso de dos Fragments.*

Ventajas del uso de Fragments

- Proporciona diseños más dinámicos y flexibles para pantallas más grandes.
- No son necesarios cambios muy profundos en la jerarquía de vistas.
- Cada Fragment definido en una Activity es independiente del resto de Fragments, y por lo tanto reutilizable.

- Facilita la tarea de desarrollo de Apps que funcionen correctamente tanto en tablets como en dispositivos móviles (multidispositivo).
- Un Fragment tiene su propio Layout y su propio ciclo de vida.