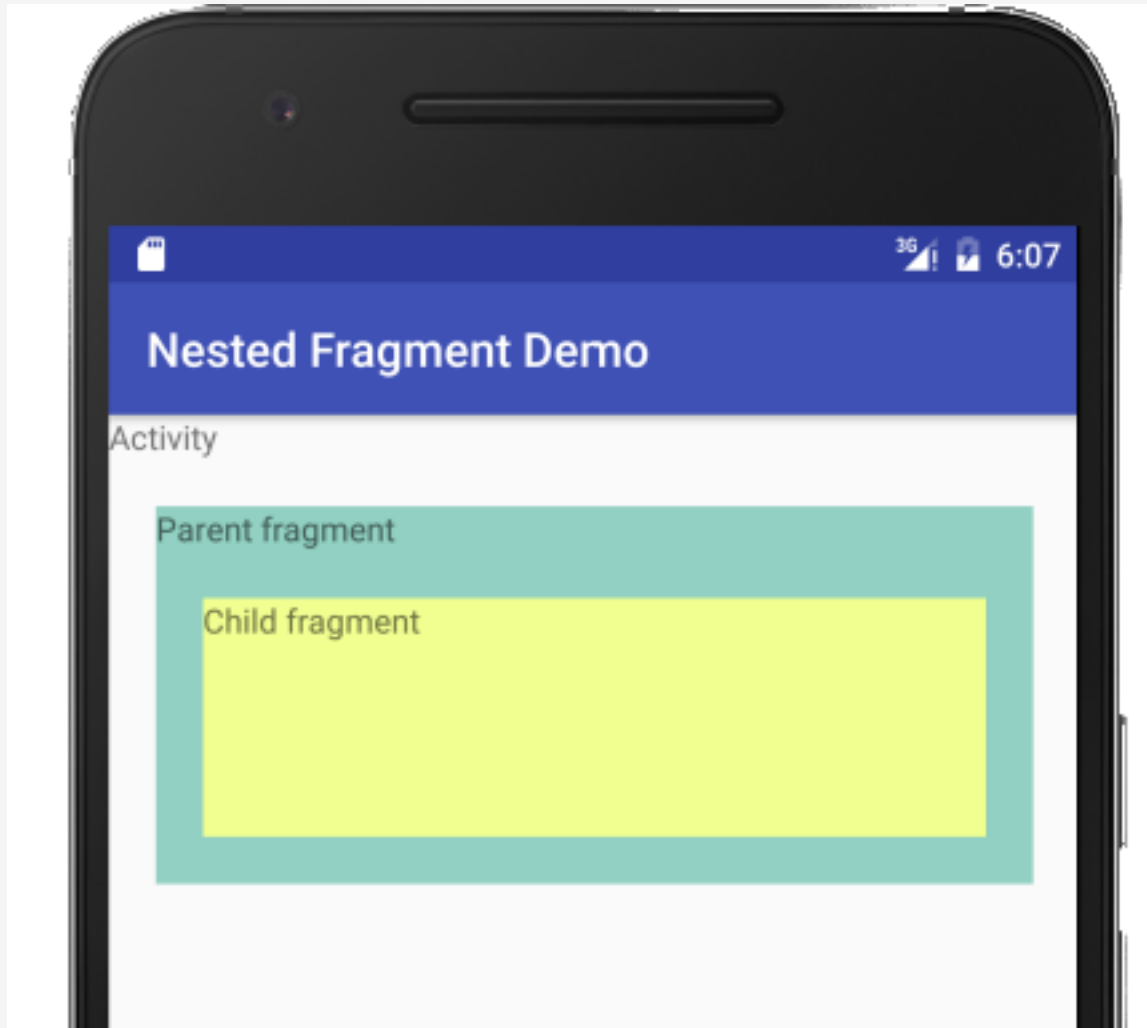


Fragmentos

Clase Nro. 6

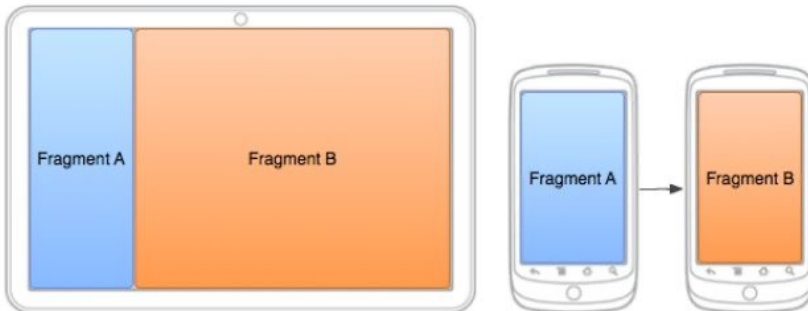
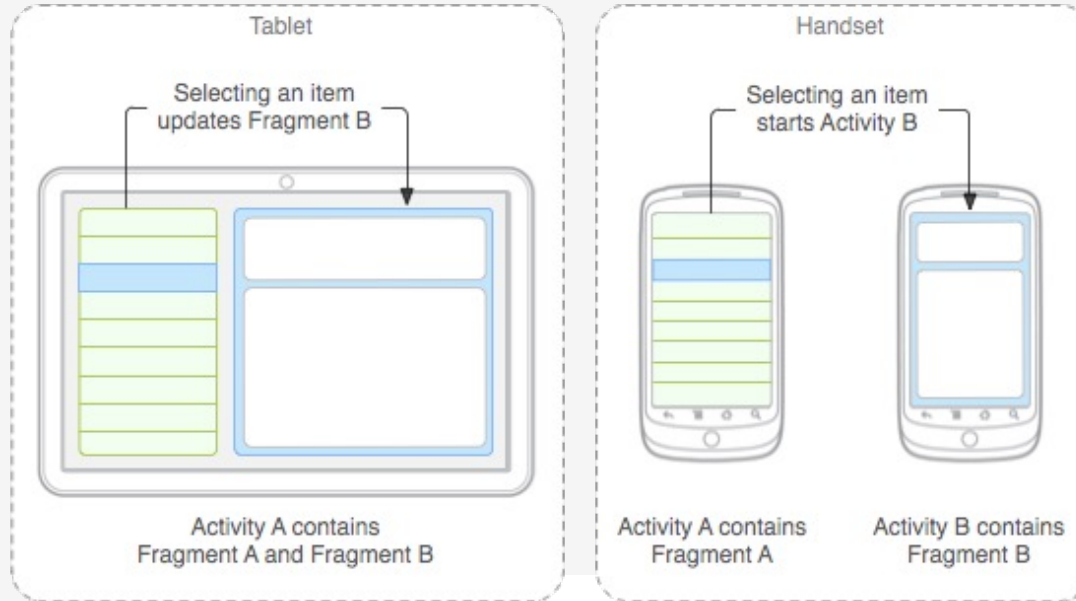
Teófilo Copa F.

Uso de Fragmentos (*Fragment*)



- Es una porción «**modular**» de la UI embebida dentro de una **actividad**.
- Tiene su **propio diseño** (un recurso layout propio) *y ciclo de vida propia*.

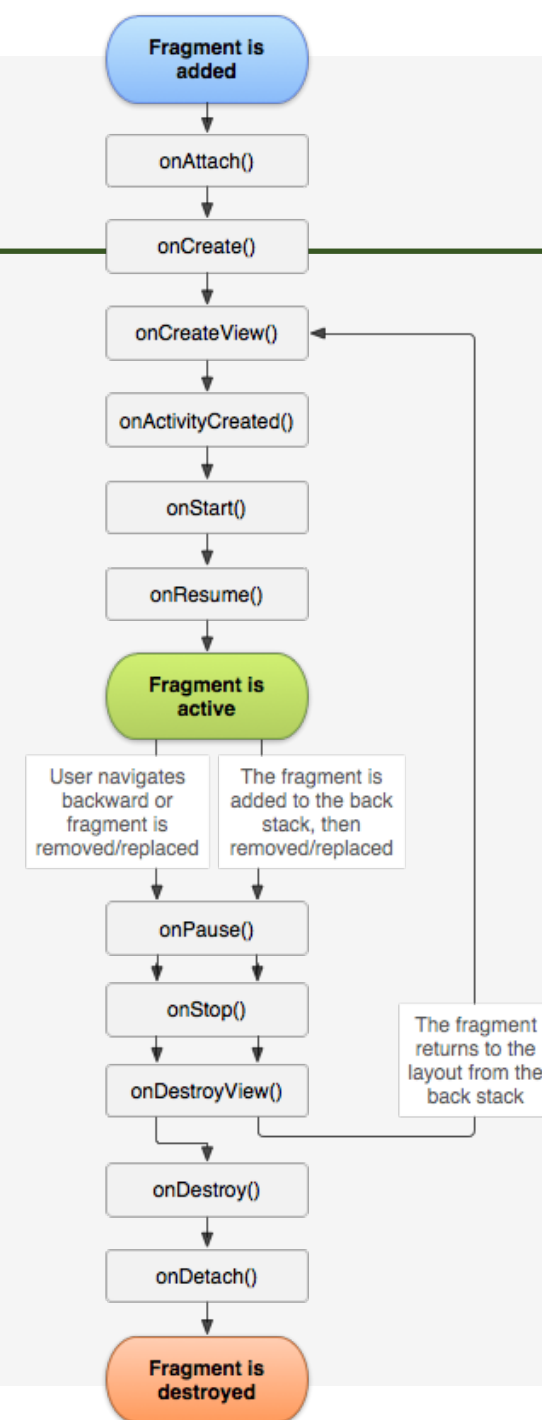
Uso de Fragmentos (*Ventajas*)



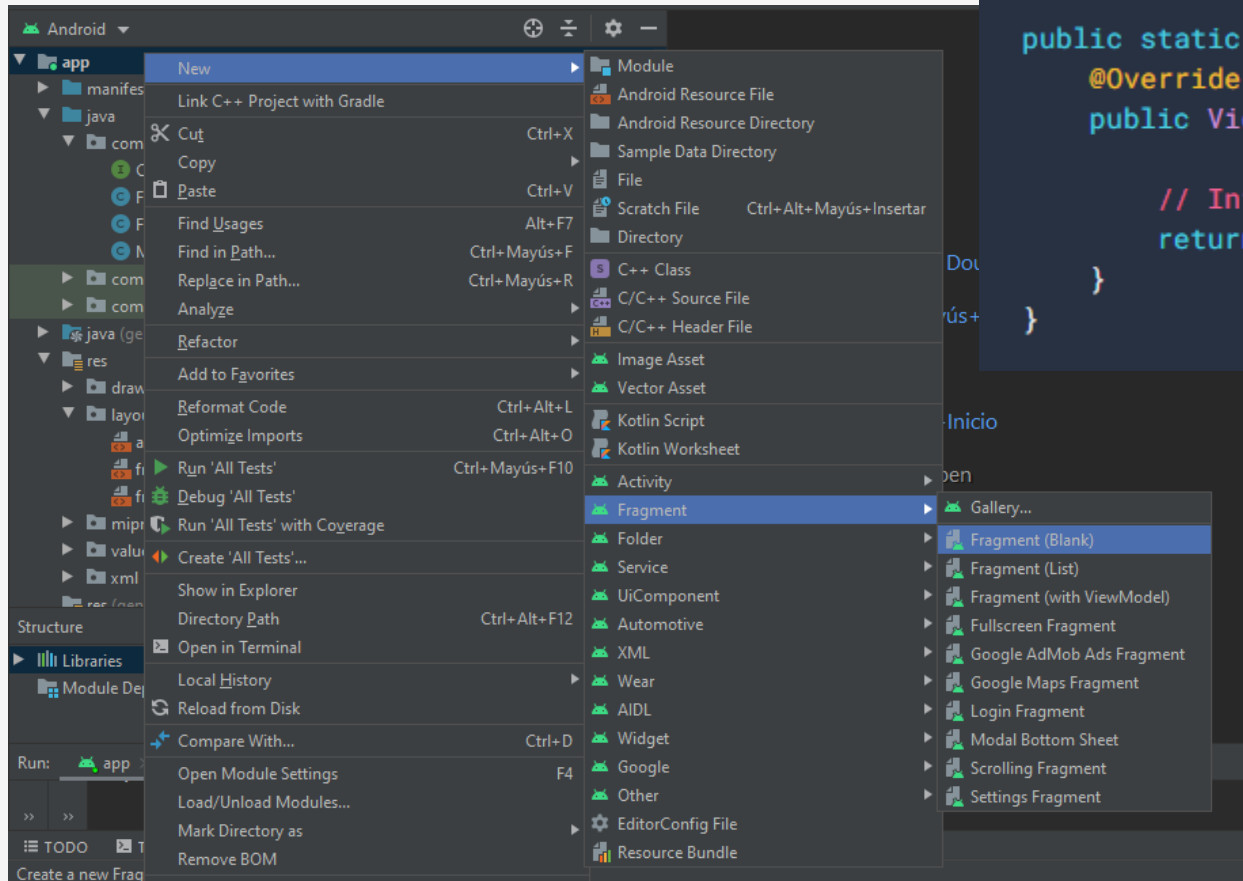
- Permiten **reusar código** y **ahorrar tiempo** de diseño a la hora de desarrollar una aplicación.
- Permiten crear diseños de interfaces de usuario de **múltiples vistas**.

Ciclo de vida de un Fragmento

- El ciclo de vida de un fragmento se ve **afectado directamente por el ciclo de vida de la actividad.**
- Por ejemplo si la actividad está pausada, todos los fragmentos también lo están.
- Un fragmento siempre se **aloja en una actividad.**
- Puedes reusar **un mismo fragmento** en **diferentes actividades**



Como crear un Fragmento en Android Studio



```
public static class ExampleFragment extends Fragment {  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.example_fragment, container, false);  
    }  
}
```

Como agregar un Fragmento en el archivo Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
        android:id="@+id/list"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
        android:id="@+id/viewer"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```

- El atributo **android:name** de `<fragment>` especifica la clase kotlin/java del Fragment

Como agregar un Fragmento desde java

```
FragmentManager fragmentManager = getSupportFragmentManager();  
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
```

```
ExampleFragment fragment = new ExampleFragment();  
fragmentTransaction.add(R.id.fragment_container, fragment);  
fragmentTransaction.commit();
```

- Mientras la actividad se está ejecutando, se puede agregar fragmentos al diseño .
- Se necesita especificar un **ViewGroup** en el que se colocará el fragmento.

Como comunicarse con la Actividad

- Un fragmento está directamente vinculada a la actividad que la contiene usando el método **getActivity()**

```
View listView = getActivity().findViewById(R.id.list);
```

- Una actividad puede llamar a métodos del fragmento mediante la adquisición de una referencia a Fragment desde **FragmentManager** usando **findFragmentById()** o **findFragmentByTag()**.

```
ExampleFragment fragment = (ExampleFragment) getSupportFragmentManager().findFragmentById(R.id.example_fragment);
```


Compartir datos entre fragmentos

```
package com.example.lab2ejemplo

Teofilo Copa
interface ContadorListener {
    Teofilo Copa
    fun incrementar() //metodo d
    Teofilo Copa
    fun getValorActual():Int //
    Teofilo Copa
    fun resetear() //metodo que
    Teofilo Copa
    fun reducir() //meotod que
}
```

```
Teofilo Copa
11 </> class MainActivity : AppCompatActivity(), ContadorListener {
12     var fra1: FragmentUno? = null
13     var fra2: FragmentDos? = null
14     var btnMain1: Button? = null
15     var btnMain2: Button? = null
16
17     var cont = 0
18     Teofilo Copa
19     override fun incrementar() {
20         cont++
21     }
22
23     Teofilo Copa
24     override fun getValorActual(): Int {
25         return cont
26     }
27
28     Teofilo Copa
29     override fun resetear() {
30         cont = 0
31     }
32
33     Teofilo Copa
34     override fun reducir() {
35         if (cont > 0) cont--
36         else cont = 0
37     }
38 }
```

- Mediante el uso de una interfaz

Compartir datos entre fragmentos

```
fra2?.addListener( object : ContadorListener {  
    override fun incrementar() {  
        TODO( reason: "Not yet implemented")  
    }  
  
    override fun getValorActual(): Int {  
        TODO( reason: "Not yet implemented")  
    }  
  
    override fun resetear() {  
        TODO( reason: "Not yet implemented")  
    }  
  
    override fun reducir() {  
        TODO( reason: "Not yet implemented")  
    }  
})
```

- Mediante el la instancia de una clase anónima

Compartir datos entre fragmentos

- Mediante el uso de una interfaz

```
Teofilo Copa *
/> class FragmentDos : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

    var contadorListener: ContadorListener? = null
    Teofilo Copa
    fun addListener(c: ContadorListener) {
        this.contadorListener = c
    }
}
```

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    val txt2 = view.findViewById<TextView>(R.id.txt2)
    val btn = view.findViewById<Button>(R.id.btn2)
    btn.setOnClickListener { it: View!
        contadorListener?.reducir();
        txt2.setText(contadorListener!!.getValorActual().toString())
    }
}
```

Demostración