Fragments

Mª Alejandra Tomás Fernández

Introducción

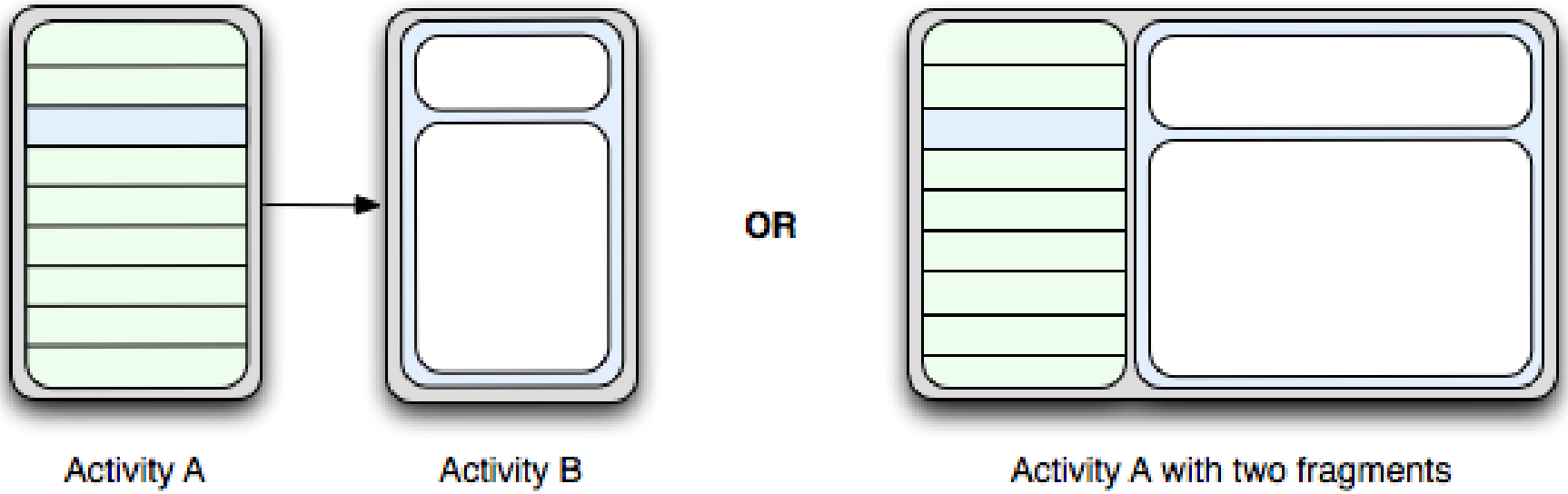
Mª Alejandra Tomás Fernández

Introducción a los Fragments (I)
(ver su API)

- **Se introdujeron en Android 3.0** (***nivel API "Honeycomb"***), principalmente para soportar diseños de IU más dinámicos y flexibles en pantallas grandes, como **las tablets**.
- Con lo **móviles**, el **diseño de una activity** consistía en **ocupar toda la pantalla**.
- Esto con **las tablets** implica **desaprovechar mucho sitio**, puesto que la **pantalla es más grande** y **se podrían mostrar varios contenidos a la vez** (***uno a la izda y otro a la derecha, por ejemplo***).
- Esa es la **idea de los FRAGMENTS**: **dividir el layout de una actividad en fragmentos** (***paneles***).

Introducción a los Fragments (II)
(ver su API)

- Por **ejemplo**: una **aplicación sobre noticias** puede utilizar **un fragmento** para **mostrar una lista de artículos a la izquierda** y **otro fragmento** para **mostrar un artículo a la derecha.**

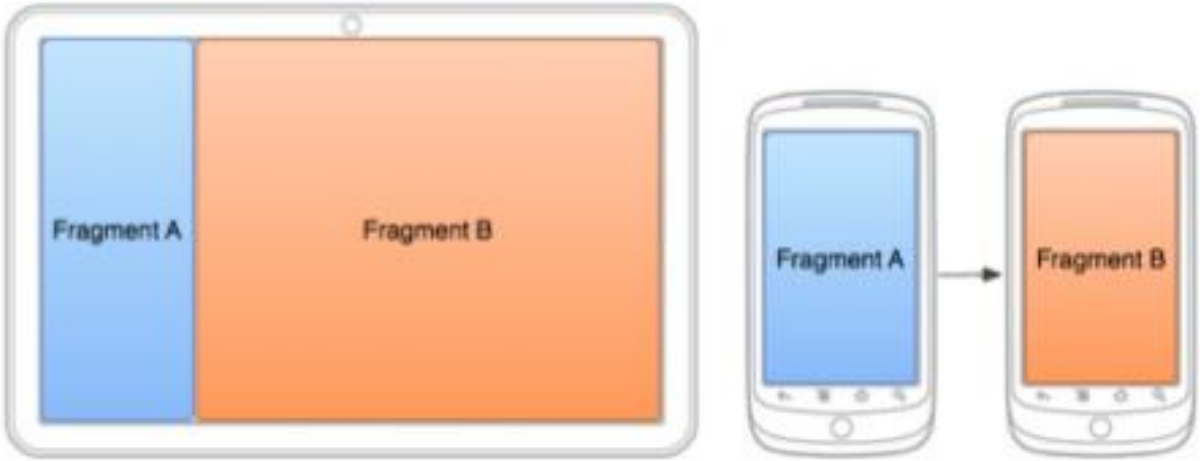


- Los **fragmentos aparecen en la actividad**, **juntos**.
- En **este ejemplo**, en el **teléfono** habrá **2 activity**, **cada una con su fragment** (***A y B***), y en la **tablet** 1 **activity** con **2 fragments**.

Introducción a los Fragments (III)
(ver su API)

- Un **fragment** se encarga de **controlar** el comportamiento de **una porción de la interfaz de usuario de una Activity.**

Introducción a los Fragments (IV)
(ver su API)



- Los **Fragments** tienen su **propio ciclo de vida**.
- **Su ciclo de vida** está totalmente **ligado al ciclo de vida de una activity** (***si ésta muere**, todos sus **fragments se destruyen**).*
- **Reciben eventos de entrada** y tiene sus **propios callback**.
- **No tienen contexto propio**. Su **contexto** es el de la **Activity** a la que están ligados.
- Hay varios **tipos de Fragments**, según su función: **DialogFragment**, **ListFragment**, **PreferenceFragment**, y **WebViewFragment**

Usar Fragments antes de Android 3.0 (V)
(ver su API)

- Google apuesta fuerte por ***fragments***.
- ¿Qué pasa con dispositivos anteriores a la 3.0?
  - Se ha creado una librería de compatibilidad que será lincada automáticamente si indicamos: minVersion<3.0

```
<uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
```

- Tenemos 2 posible paquetes a usar para programar:
  - android.app.Fragment: Si minVersion >=3.0
  - android.support.v4.app.Fragment: Si minVersion<3.0
- Nunca se pueden mezclar estos 2 paquetes en un mismo proyecto.

¿Qué son?

Mª Alejandra Tomás Fernández

¿Qué es un Fragment? (I)
(ver su API)

- Representa un **comportamiento/porción de un interface de usuario (UI)** de una activity.
- Los fragments **son almacenados en activities**:
  - **MÚLTIPLES FRAGMENTS, UNA ACTIVITY:**

    Usar una activity sin tener en cuenta el tamaño del dispositivo, pero decidiendo en tiempo de ejecución:

    1. Si se incluyen ambos fragments en el layout de la activity, para crear un *interface de usuario MULTI-PANEL,*
    2. Si se intercambian los fragments, de forma que sólo se ve uno de cada vez, para crear un *interface de usuario SIMPLE-PANEL.*

  - **MÚLTIPLES FRAGMENTS, MÚLTIPLES ACTIVITIES**:

    ❑ En una **tablet** situamos **varios fragments en la misma activity**, pero

    ❑ En **dispositivos más pequeños**, usamos **activities separadas para almacenar cada fragment.**

De esta forma, podemos, por ejemplo: si el diseño de la tablet lleva 2 fragments en una activity, podemos aprovechar la misma activity para dispositivos pequeños, pero sustituyendo el layout por uno alternativo que incluya sólo el primer fragment. Cuando se esté ejecutando en un dispositivo pequeño y debamos cambiar al otro fragment, arrancaremos la otra activity que incluye el otro fragment.

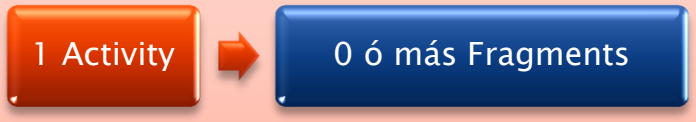
¿Qué es un Fragment? (II)
(ver su API)

- Los fragments **son almacenados en activities,** por tanto, tienen que ser:
  - **Cargados**
  - **Mostrados**
  - **Borrados,**
  - **…**

  según las activities vayan cambiando su estado.

IMPORTANTE:

- Varios FRAGMENTS pueden ser incrustados en 1 ACTIVITY (UI MULTIPANEL)



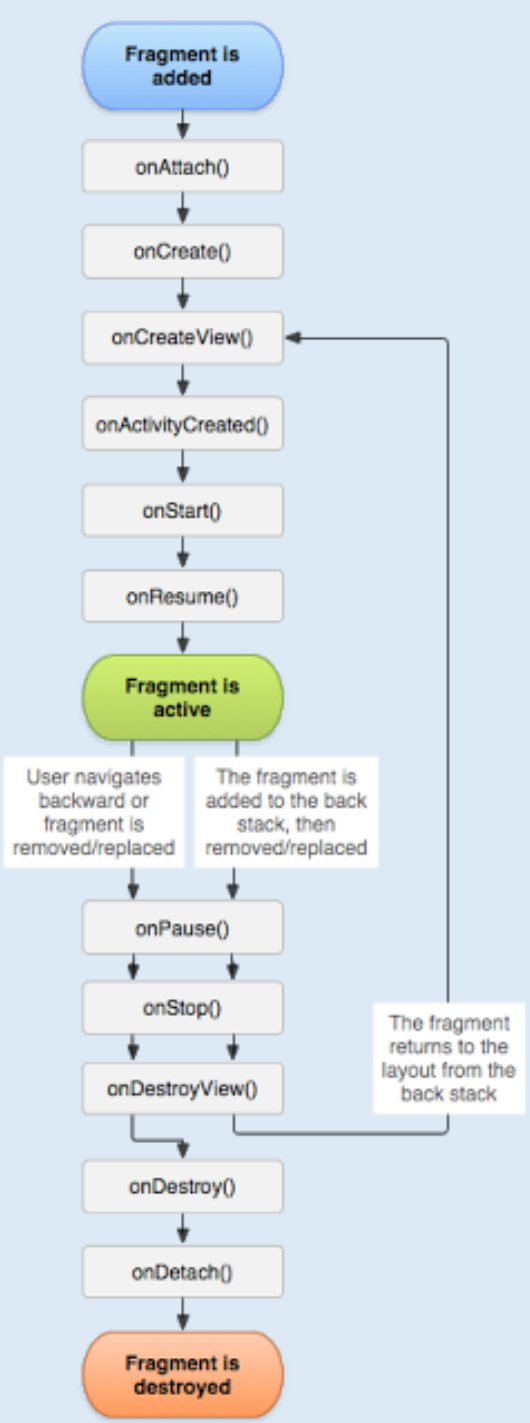
- Un FRAGMENT puede ser reusado en varias ACTIVITIES

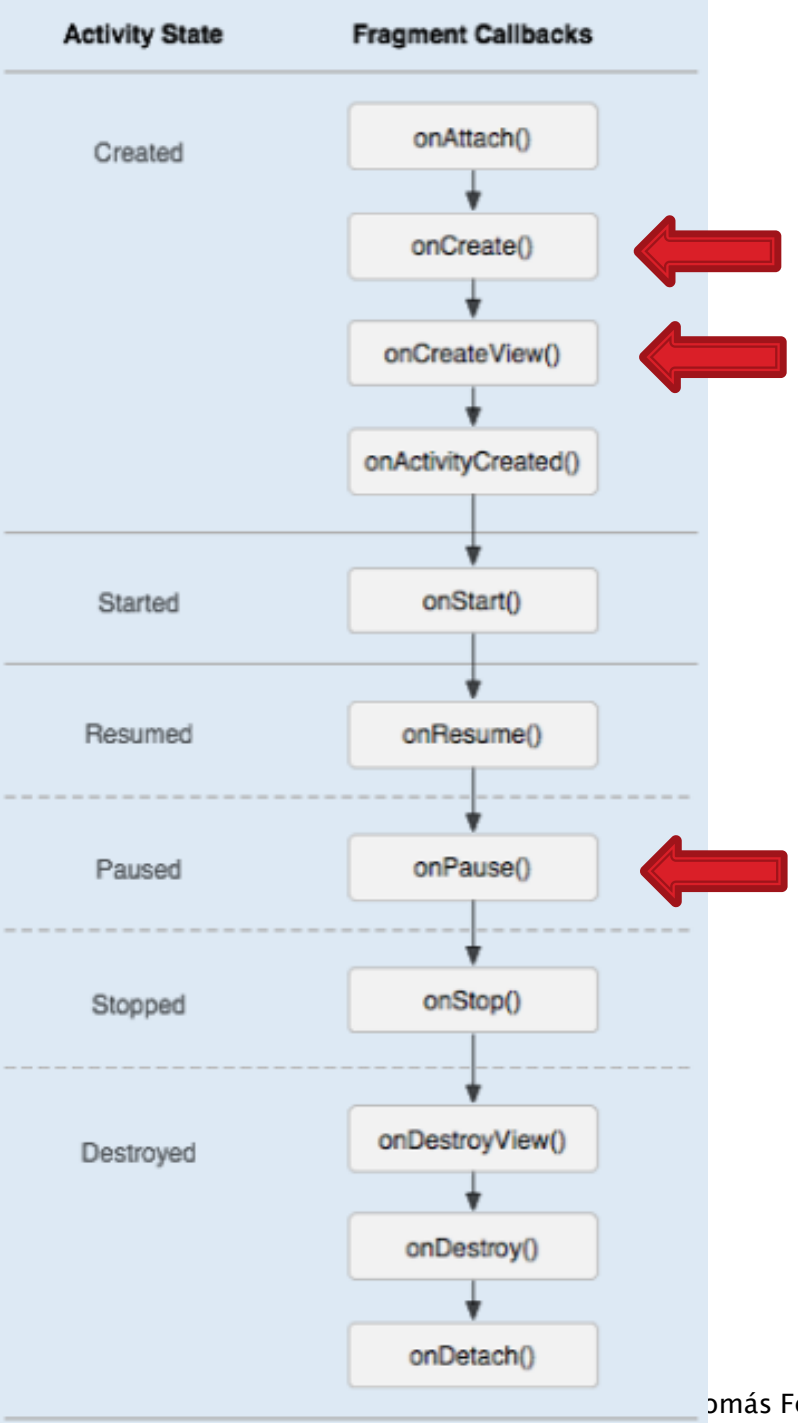
¿Qué es un Fragment? (III)
(ver su API)

- Un **fragment tiene** su propio:
  - **LAYOUT**
  - **CICLO DE VIDA** (asociado al de la Activity que lo contiene)
    - *Incluye sus propios* ***métodos CallBacks***
  - **EVENTOS DE USUARIO**

Ciclo de vida de un Fragment

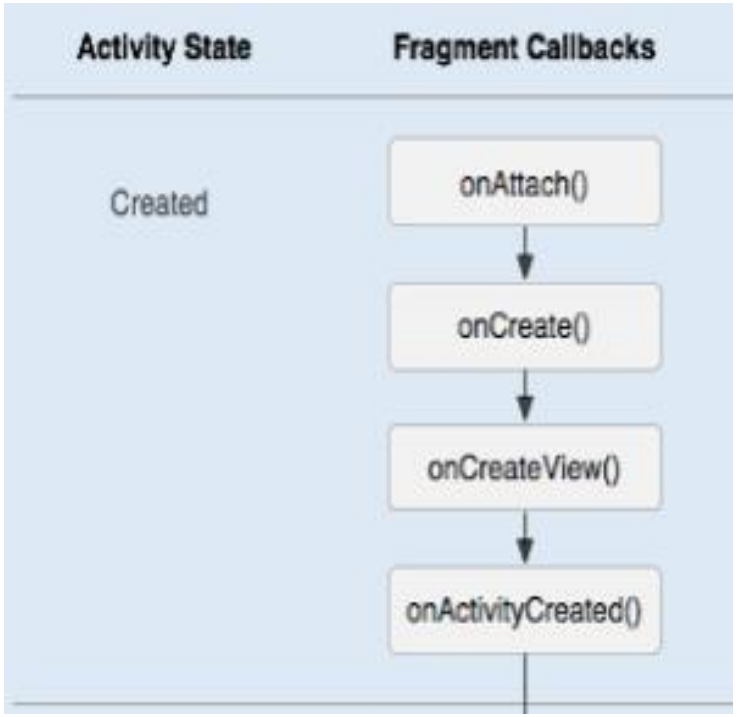
Mª Alejandra Tomás Fernández


Fragment is added
onAttach()
onCreate()
onCreateView()
onActivityCreated()
onStart()
onResume()
Fragment is active
User navigates backward or fragment is removed/replaced
The fragment is added to the back stack, then removed/replaced
onPause()
onStop()
onDestroyView()
The fragment returns to the layout from the back stack
onDestroy()
onDetach()
Fragment is destroyed


Activity State
Fragment Callbacks
Created
onAttach()
onCreate()
onCreateView()
onActivityCreated()
Started
onStart()
Resumed
onResume()
Paused
onPause()
Stopped
onStop()
Destroyed
onDestroyView()
onDestroy()
onDetach()

Ciclo vida Fragment (II)

Cuando la **Activity que contiene el fragment está siendo creada**, varios ***callbacks*** son llamados



onAttach() → Llamado cuando el fragment ha sido asociado con la Activity (Activity is passed in here).

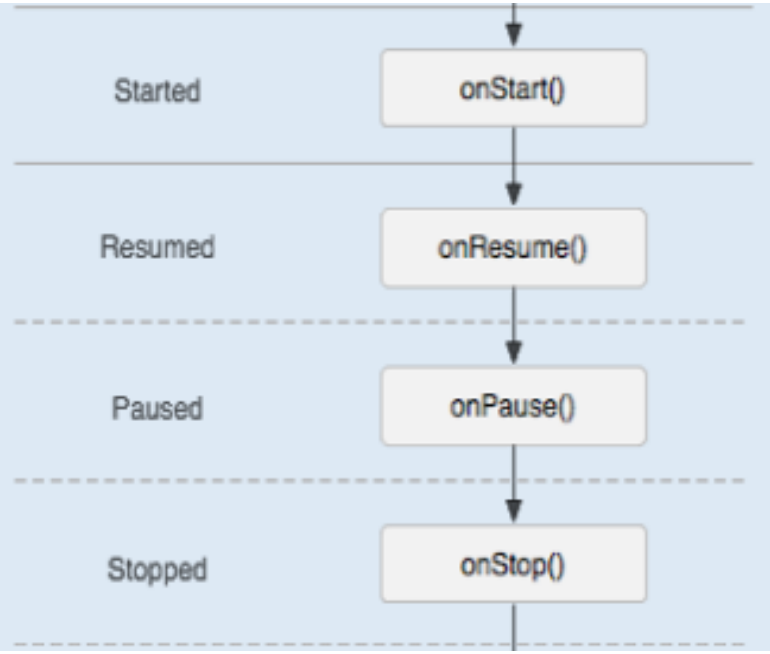
onCreate() →Llamado cuando se crea el fragment. Debería ***inicializarse componentes no de UI*** esenciales que queramos conservar cuando el fragment pause, pare o reanude.

onCreateView() →Llamado para crear la jerarquía de vistas de la UI. **Aquí configuraremos la UI y devuelve una View con el UI**.

onActivityCreated() →Llamado cuando el método onCreate() de la Activity ha terminado. Es decir, **Activity creada**.

Ciclo vida Fragment (III)

Cuando la **Activity que contiene el fragment está siendo started, resumed, paused, stopped**, varios ***callbacks*** son llamados



onStart() → Llamado cuando el fragment es **visible** al usuario.

onResume() →Llamado cuando el fragment es visible al usuario e **interactuando**.

onPause() →Llamado cuando el fragment **ya no interactúa** con el usuario:

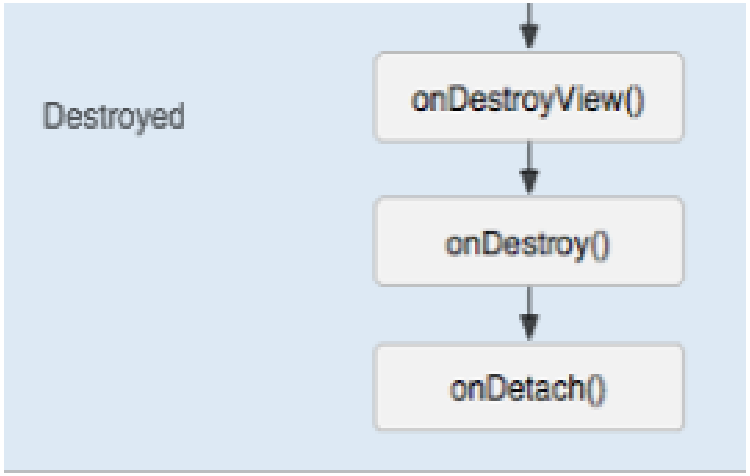
- ✓ ya sea porque la Activity esté siendo pausada,
- ✓ O el fragment esté siendo modificado en la Activity (sustituido,…).

onStop() →Llamado cuando el fragment **ya no es visible** al usuario:

- ✓ Ya sea porque la Activity esté siendo parada,
- ✓ O el fragment esté siendo modificado en la Activity (sustituido, eliminado,…)

Ciclo vida Fragment (IV)

Cuando la **Activity está siendo destruida** varios *callbacks* son llamados



onDestroyView() → Llamado cuando la jerarquía de **vistas asociada con el fragment y creada en onCreateView() ha sido desligada del fragment. La próxima vez que el fragment necesite ser mostrado, deberá crear de nuevo su vista.**

Deberíamos *liberar recursos asociados con la UI*.

onDestroy() →Llamado cuando el fragment ya no está en uso.

Deberíamos *liberar recursos*.

onDetach() →Llamado cuando el fragment está siendo desasociado de su activity.

Deberíamos *poner a NULL las referencias a la hosting activity*.

¿Qué ocurre al pulsar la tecla BACK del tfno, tablet,…?

Mª Alejandra Tomás Fernández

¿Qué ocurre al pulsar la tecla BACK del dispositivo cuando hay fragments?

- La **principal diferencia** entre **el ciclo de vida de una activity y un fragment** es **CÓMO SON ALMACENADOS CADA UNO EN SUS RESPECTIVAS PILAS "BACK STACK"**:

  - Una **activiy** es situada en una **pila de Activities** que **gestiona el sistema automáticamente**, cuando esa activity es parada *(por ejemplo, al pulsar el botón BACK).*

  - Sin embargo, un **fragment**:

    a) Es **almacenado en una pila** (BACK STACK) **diferente** y **gestionada por la activity que lo contiene** y,

    b) **Sólo lo almacena en esa pila si NOSOTROS EXPLÍCITAMENTE le decimos** que lo guarde en la pila mediante la llamada al método "**addToBackStack()**" de la transacción (***fragmentTransaction***). Lo que realmente memoriza este método es la transacción.

¿Cómo crear un Fragment?

Mª Alejandra Tomás Fernández

Creando un Fragment (I)

- Para cada fragment *(al igual que ocurre con las Activities*), necesitamos crear:
  - **LAYOUT** que representa su interfaz de usuario.
    - Puede haber fragments que no tengan parte gráfica, sólo representen un comportamiento de fondo (gestionar logins desde diferentes sitios: facebook, twitter,…
  - **Subclase de Fragment** que representa la lógica del Fragment (JAVA) y donde sobreescribamos el método **onCreateView()** ***en el que devolveremos la vista de dicho Fragment (su LAYOUT)***

```
public class FragmentUNO extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {

        return inflater.inflate(R.layout.fragment_uno, container, false);
    }

}
```

Vista padre donde se insertará el layout

Indica si el inflado del layout debe ser insertado en el ViewGroup (En este caso es false porque directamente se esta insertando el layout en el ViewGroup).

Layout del fragment

View padre

Creando un Fragment (II)

- Android incorpora también unas subclases de Fragment que podemos extender según el tipo de Fragment que queramos usar:

  - **DialogFragment:** muestra un **diálogo flotante.**
  - **ListFragment**: muestra una **lista de items** gestionada por un adaptador (similar a **ListActivity**). Provee varios métodos para gestionar la lista, como *onListItemClick().*
    - *Si usamos este tipo de Fragment, no hace falta crear un LAYOUT, pues lo crea por defecto y será una LISTVIEW.*
  - **PreferenceFragment:** Muestra una **lista de preferencias** (similar a **PreferenceActivity**). Se suele usar cuando se crean una Activity de "**configuración**" para un juego, aplicación,…

Tipos de Fragments

Mª Alejandra Tomás Fernández

Tipos de Fragments (I)

1. **FRAGMENTS ESTÁTICOS**
   - ✓ Son aquellos que **se declaran directamente en el fichero XML del layout de la Activity** *(en el fichero layout que es llamado en **setContentView()**)*
   - ✓ Estos fragments **no pueden ser eliminados ni sustituidos** por nada (obtendremos errores en caso contrario).
   - ✓ Se pueden **especificar propiedades de layout para el fragment** como si fuese una vista normal: ancho, alto, peso,…

2. **FRAGMENTS DINÁMICOS**
   - ✓ Son aquellos que **se añaden a la activity en tiempo de ejecución desde código JAVA** y se asocian a un ViewGroup (*se recomienda el uso de FrameLayout)*.
   - ✓ Se añaden usando **FragmentManager** y **FragmentTransaction**.
   - ✓ **Pueden ser eliminados o sustituidos por otro fragment** u otro contenido.

¿Cómo obtener el Context desde un Fragment?

Mª Alejandra Tomás Fernández

¿cómo obtener el Context desde un Fragment?

- Ya sabemos que **un Fragment NO TIENE CONTEXTO PROPIO, SU CONTEXTO ES EL DE LA PROPIA ACTIVITY** que lo contiene.

  - **Si necesitamos el Context desde nuestro Fragment, podemos llamar a " getActivity()".**

  - **CUIDADO:**

    - *Sólo podemos llamar a "getActivity()" cuando el fragment está ligado (attached) a la activity*.
    - Si el **fragment aún no ha sido ligado** a la Activity o **ha sido** *detached*, ese método **devuelve NULL.**