Introducción a Android

Sesión 4

UNIVERSIDAD RICARDO PALMA
LIMA • MCMLXI X • PERU

¿Qué vamos a ver hoy día?

- Ciclo de vida de una aplicación Android
- Activities
- Fragments

Ciclo de vida

- Los pasos que una aplicación sigue desde su inicio a su fin
- Diferente al ciclo de vida de un aplicativo java
- Recursos limitados de hardware

Aplicación Android

- Cada aplicación restringe el acceso a sus recursos por parte de otras aplicaciones.

- Todos los componentes de una aplicación corren en un mismo proceso (si es que no se especifica lo contrario).

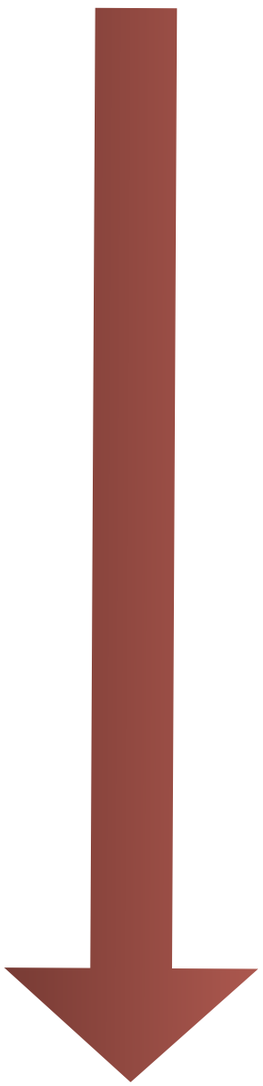
- Cada proceso se situa en una pila.

¿Qué sucede cuando se agota la memoria o el CPU se hace más lento?

- Procesos pueden ser terminados para reclamar la necesidad de recursos.
  - ¿Cómo se decide qué proceso terminar?
- Ciertos componentes terminados pueden ser restauradas a su último estado cuando sea requerido por el usuario.

¿Cómo decidir que proceso terminar?

Menos probable



Más probable

1. El proceso corriendo con un componente en primer plano (foreground).

2. Cualquier proceso con un componente que sea visible, pero no corriendo en primer plano (foreground).

3. Cualquier proceso corriendo un componente que no este en primer plano (background).

4. Cualquier proceso no corriendo un componente. Este es conocido como un proceso vacio.

Activity

- Uno de los componentes principales de las app Android
- Provee la interfaz para que el usuario interactúe con la aplicación
- Posee un ciclo de vida, controlado por el sistema
    - Callbacks para insertar nuestro código en los puntos vitales del ciclo de vida

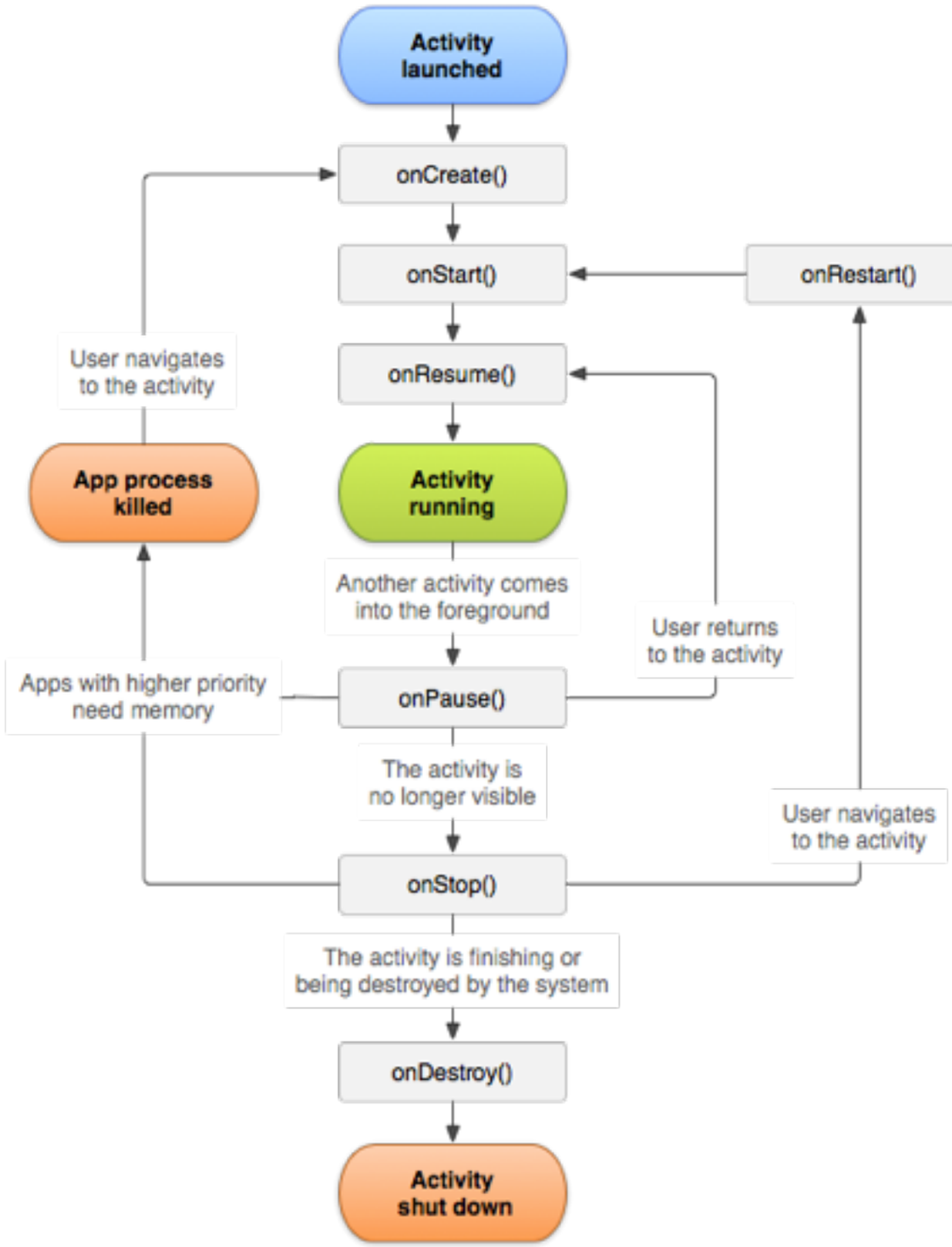
Creación de un Activity

- Diseño de interfaz de usuario (en XML)
- Código Java:
  - Extender la clase `Activity` (o una de sus subclases)
    - Implementar el método `onCreate`
      - Establecer la interfaz de usuario con `setContentView`
- Declaración en el Manifest

Ciclo de vida de un Activity

- Controlado por el sistema Android
- Podemos ejecutar nuestro código en determinados puntos sobreescribiendo los métodos de la clase `Activity`
- Siempre llamar al método de la clase base (`super.on...()`)

Ciclo de vida de un Activity



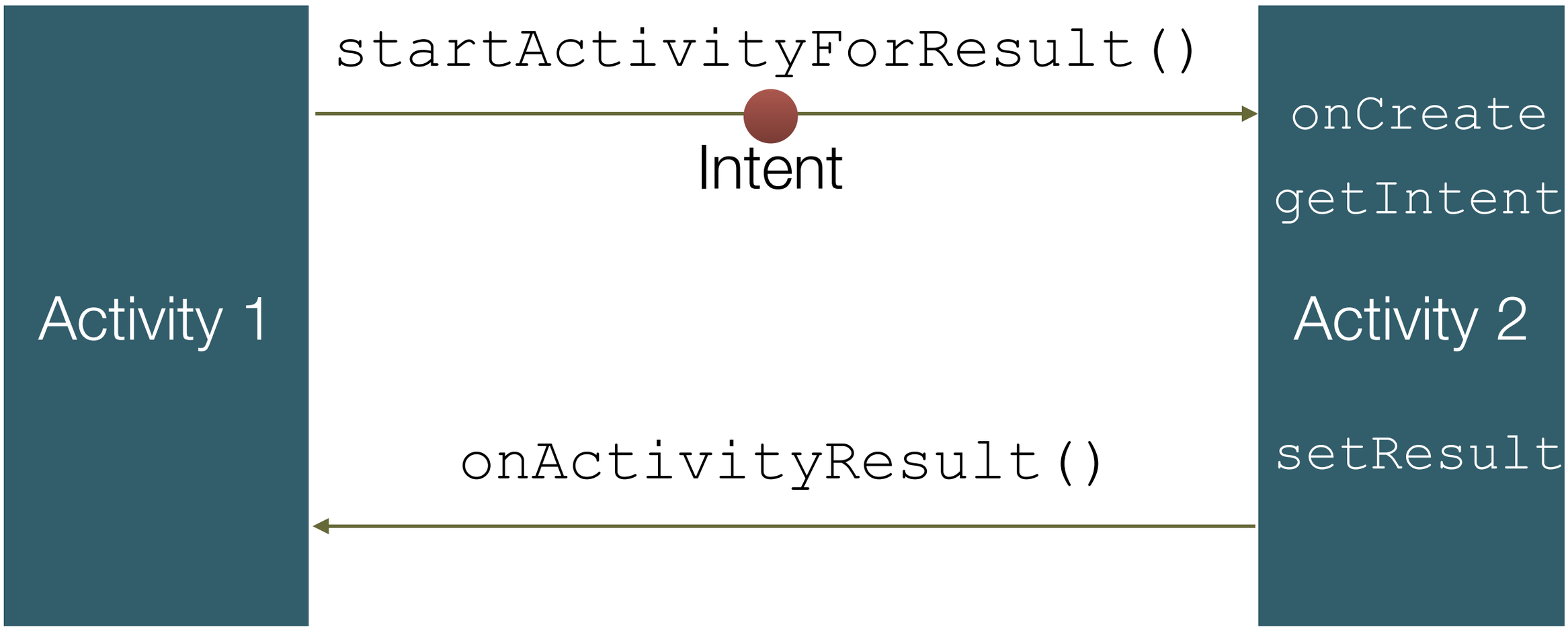
Ciclo de vida de un Activity

- onCreate(): Llamado cuando un activity es creado. Normalmente aca se hace la inicialización de la actividad.
- onStart(): Llamado cuando un activity se hace visible en la pantalla del usuario.
- onResume(): Llamado cuando un activity comienza a interactuar con el usuario. Este callback se llama siempre que se inicia (start) o se reinicia (restart) un activity.
- onPause(): Llamado cuando un activity es pauseado o reclama CPU o otros recursos. Acá se debería guardar el estado de información, para que cuando la actividad sea reiniciada, pueda volver al mismo estado.
- onStop(): Llamado cuando se requiere parar un activity. La lleva a un estado background.
- onRestart(): Llamado cuando una actividad es reiniciada, si aún se encontraba en la pila (stack).
- onDestroy(): Llamado cuando una actividad es removida completamente de la memoria del sistema. Es llamado por el método onFinish() o invocado directamente por Android para librerar recursos.

Trabajando con varios Activities

- Intents para iniciar Activities
  - También permiten enviar datos de un Activity a otro
- `startActivity` y `startActivityForResult`
  - `getIntent`
  - `setResult`
  - `onActivityResult`
- Back Stack

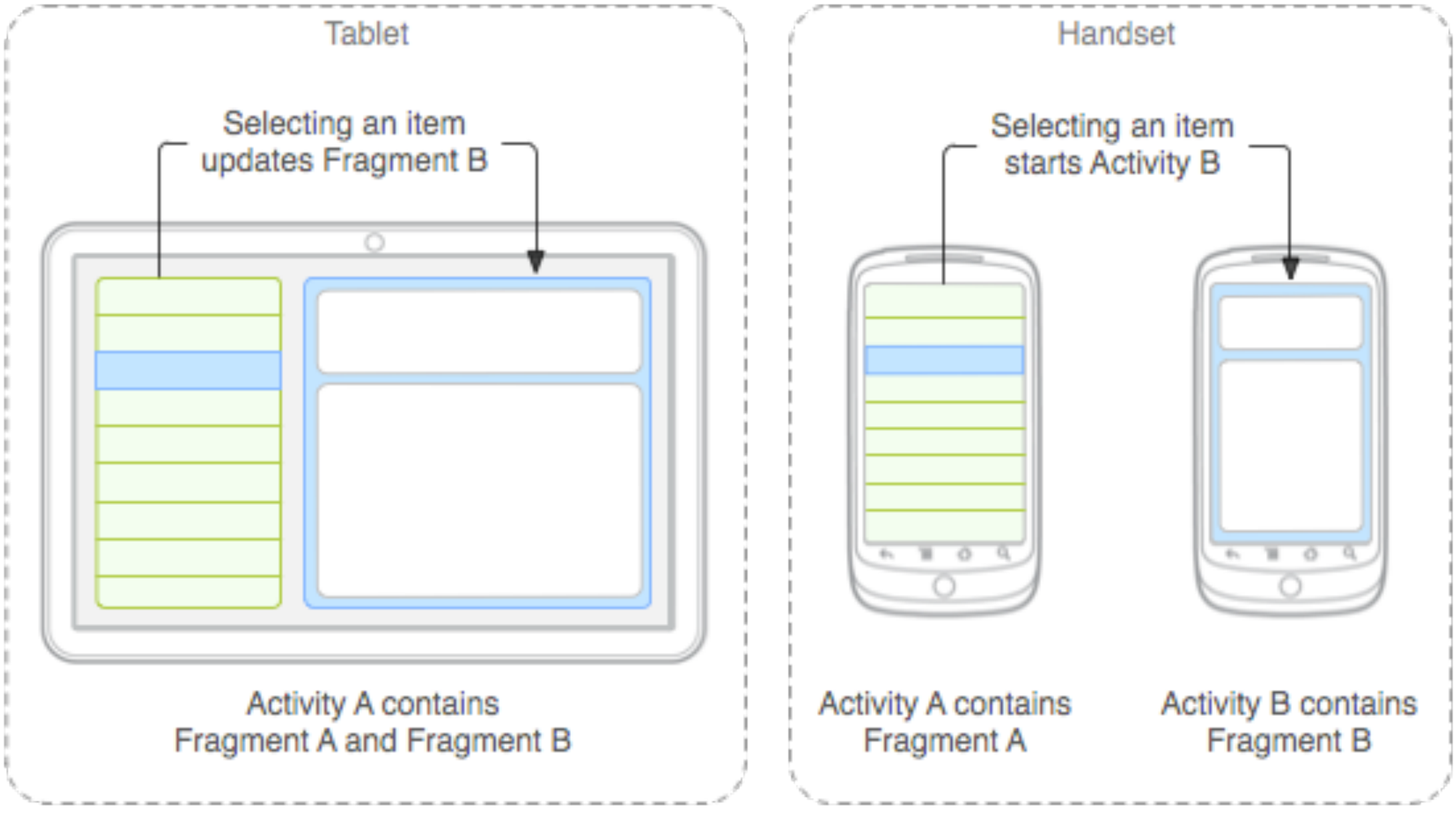
Trabajando con varios Activities



Fragments

- Sección reutilizable de la interfaz de usuario, con su propio ciclo de vida
    - Un Activity puede tener muchos Fragments
- Siempre van insertados en un Activity, y su ciclo de vida está asociado al del Activity en que están insertados
    - Se pueden colocar o quitar mientras el Activity está en primer plano

Fragments


Tablet
Selecting an item
updates Fragment B
Activity A contains
Fragment A and Fragment B
Handset
Selecting an item
starts Activity B
Activity A contains
Fragment A
Activity B contains
Fragment B

Creación de un Fragment

- Diseño de interfaz de usuario (en XML)
- Código Java:
  - Extender la clase `Fragment` (o una de sus subclases)
  - Implementar el método `onCreateView`
- Añadir el Fragment a un Activity
  - En tiempo de diseño
  - En tiempo de ejecución

Añadir el Fragment a un Activity

- En tiempo de diseño: en el XML de un Activity
  - No permite reemplazarlo dinámicamente
  - Utilizado para componentes (ejm. Google Maps)

```xml
<?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>
```

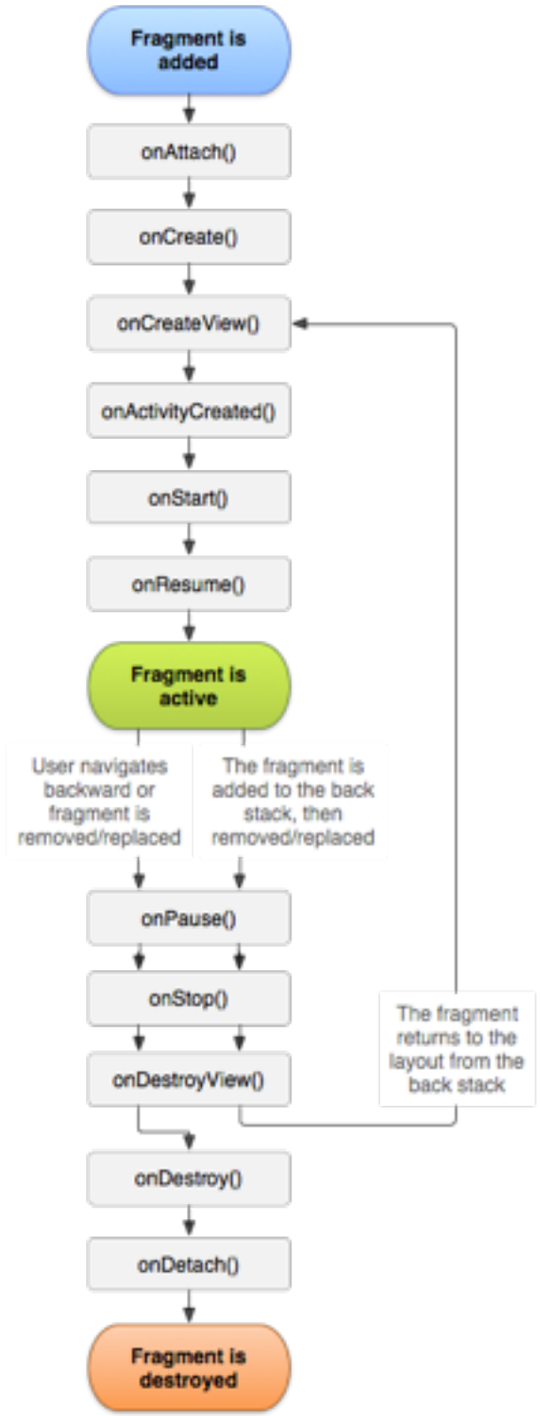
Añadir el Fragment a un Activity

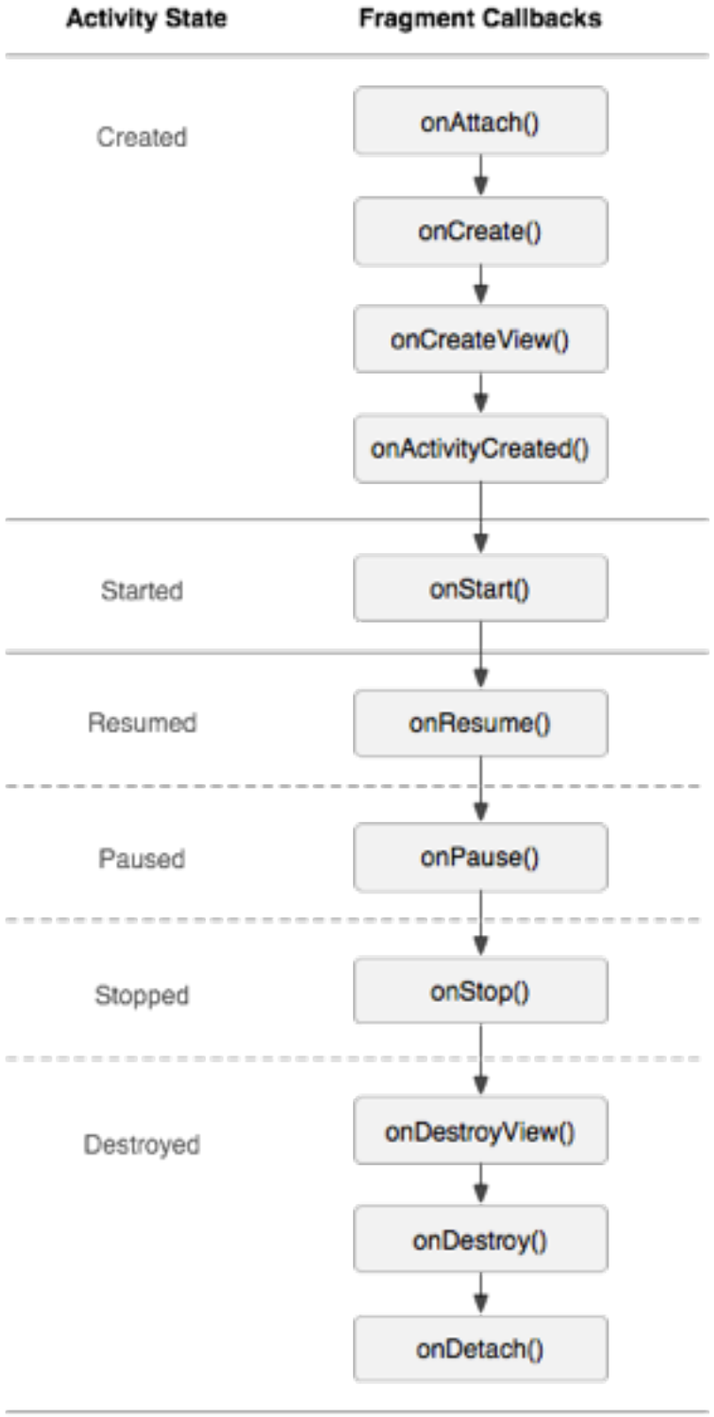
- En tiempo de ejecución: `FragmentManager` y `FragmentTransaction`
  - Permite manipular y reemplazar los Fragments
    - `add`, `remove`, `replace`, `commit`
- Los cambios incluso se pueden añadir al Back Stack

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

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

Ciclo de vida de un Fragment





Ciclo de vida de un Fragment

- onCreate(). En esta parte inicializar componentes importantes del fragment

- onCreateView(): Llamado cuando es tiempo de dibujar la interfaz gráfica por primera vez. Se debe retornar un View con la vista armada.

- onPause(): Momento en que se deben comitear cualquier cambio o persistir cierta información.

- onActivityCreated(): Buen lugar para inicializar listener de componentes View

- onAttach(): Método a ejecutarse antes del onCreate. Buen lugar para inicializar listeners que escuchan al fragment.

Comunicación con el Activity

- Fragment -> Activity: `getActivity`
- Activity -> Fragment: `findFragmentById`
- Eventos:
  - El Fragment recibe los eventos de bajo nivel (click, item seleccionado)
  - El Activity recibe eventos de alto nivel enviados por el Fragment

Comunicación con el Activity: eventos