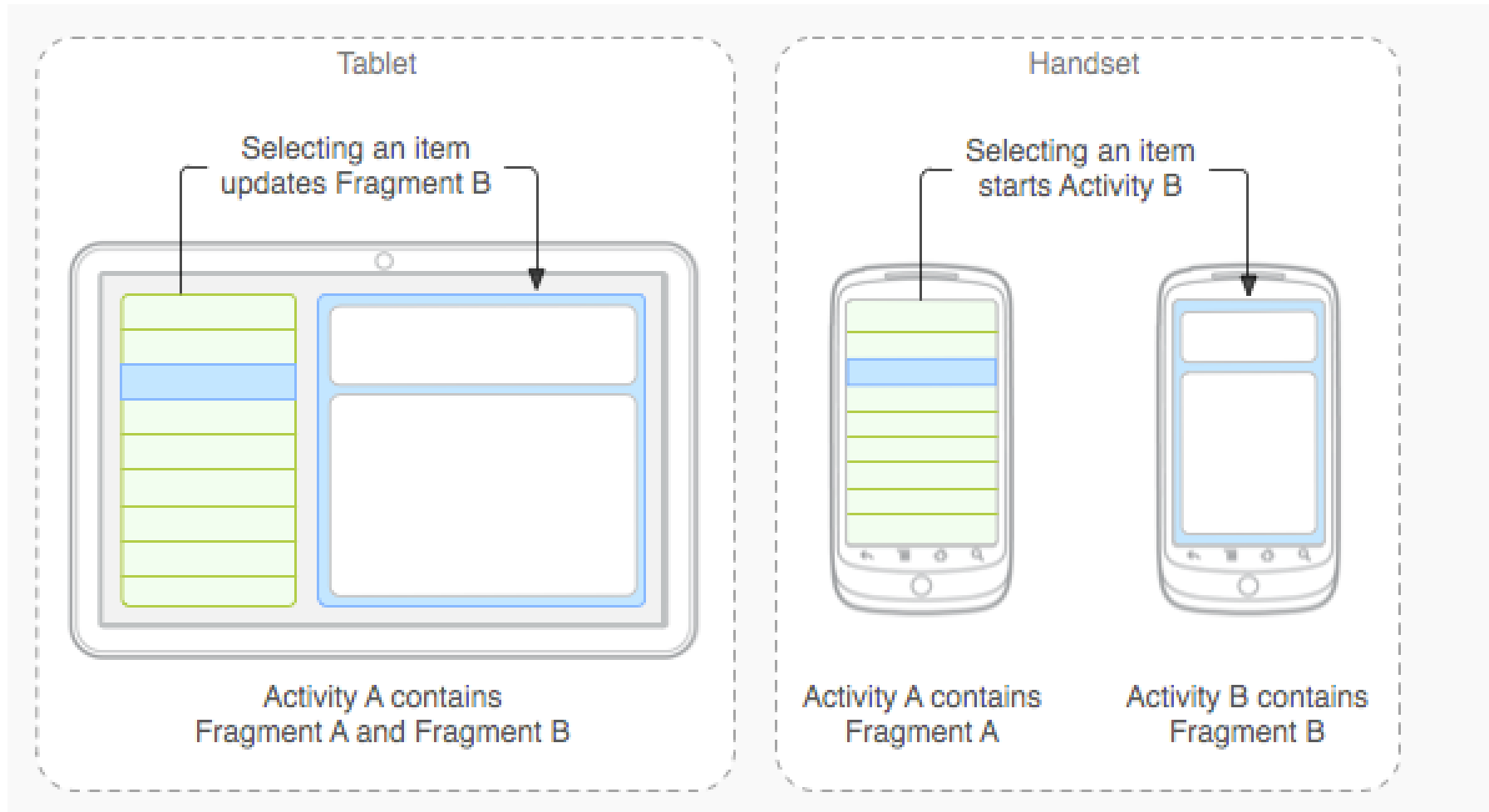


# Interfaces dinámicos (Fragments)

Phd. MPuerto Paule Ruiz

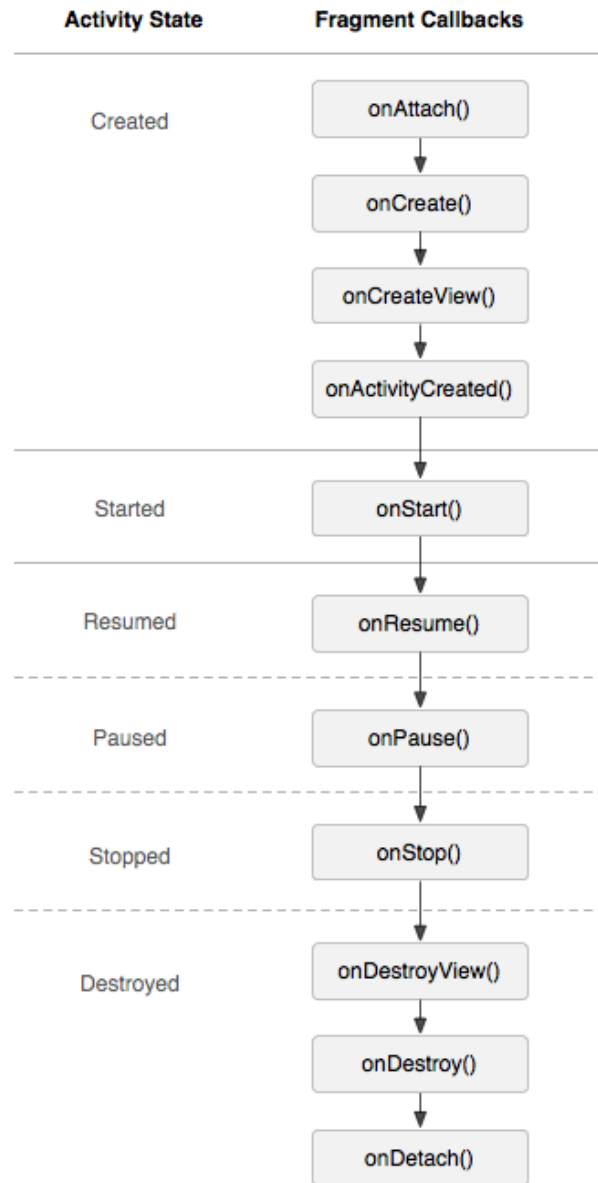
# Concepto



# Fragments

- Adaptación a distintas pantallas (diferentes dispositivos)
- Inyección dinámica de código
- API 11 o superior

# Ciclo de Vida Fragment



Fuente: <http://developer.android.com>

# Tipos Fragments

- DialogFragment
- ListFragment
- PreferenceFragment
- FragmentActivity

# Librerías

- Import  
*android.support.v4.app.Fragment[Transaction  
/FragmentManager/ListFragment/..]*
- Gradle  
Compile *`com.android.support:appcompat-v7:22.0.0`*

# Añadir Fragments a la Activity

- Declaración fragments en el layout Activity (activity\_main.xml)

`<fragment`

`android:id="@+id/map"`

`android:name="com.google.android.gms.maps.SupportMapFragment" />`

- Añadir fragments a una Vista (View)
  - Determinar el ViewGroup
  - Iniciar una transacción

# Añadir fragments a una View

layout/activity\_main.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment_container"
```



```
//creamos el fragmento
```

```
TituloFragment primerfragmento=new TituloFragment();
```

```
//Le pasamos los datos de la actividad, en caso de que ésta los tenga
primerfragmento.setArguments (getIntent().getExtras());
```

```
//Añadimos al fragment_container el fragmento de los títulos
```

```
getSupportFragmentManager().beginTransaction()
```

```
    .add (R.id.fragment_container, primerfragmento).commit();
```



# Transacciones de Fragments

- Una activity puede borrar, reemplazar o añadir un fragment ante acciones del usuario
- Esta capacidad se realiza con las transacciones

```
FragmentManager transaction =  
getSupportFragmentManager().beginTransaction();
```

```
transaction.replace(R.id.fragment_container, newFragment);  
/*añado la transacción a la pila de transacciones. Esta pila la controla la activity,  
permitiendo volver al fragment anterior si el usuario pulsa el botón de atrás*/  
transaction.addToBackStack(null);
```

```
// Commit de transaction  
transaction.commit();
```

# Comunicación con la Activity

- Un mismo fragmento puede ser usado en diferentes activities.
- Es preciso indicar en qué activity concreta está y poder trabajar con la vista precisa

```
TextView t1 = (TextView) getActivity().findViewById(R.id.text1);
```

# Comunicación entre fragments (I)

- Eventos en un fragment A que provocan acciones en otro fragment B.
- Pasos a realizar:
  - Definir una interfaz en fragment A
  - Implementar evento concreto

```
public class fragmentA extends ListFragment {  
    OnItemSelectedListener Item;  
  
    public interface OnItemSelectedListener {  
        public void onItemSelected (int value);  
    }  
}
```

# Comunicación entre fragments (II)

- Implementar la interfaz en la activity principal

```
public class MainActivity extends ActionBarActivity  
    implements fragmentA.OnItemSelectedListener {...}
```

```
public void onItemSelected(int valor) {  
    FragmentoB nuevoFragment = new FragmentoB();  
    Bundle args = new Bundle();  
    args.putInt(FragmentoB.key, valor);  
    nuevoFragment.setArguments(args);}
```

- FragmentB recoge valor key en su método onStart

(por ejemplo)

```
public void onStart() {  
    super.onStart();  
    //Empieza el fragmento  
    Bundle args = getArguments();....}
```