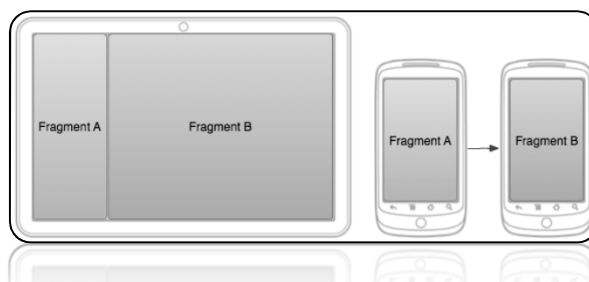


## Fragmentos (1) [obsoletos desde API 28]

Un fragmento representa una interfaz de usuario o una sección de dicha interfaz, de manera que podemos combinar varios fragmentos en una actividad o reutilizar el mismo fragmento en varias actividades.



194

## Fragmentos (2)

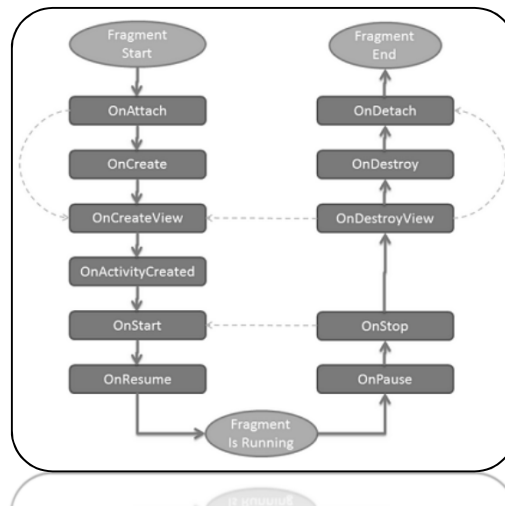
Un fragmento tiene su propia vista y su propio ciclo de vida, puede agregarse y eliminarse de una actividad de forma dinámica.

Para poder utilizar un fragmento, éste debe agregarse a una actividad y por tanto su ciclo de vida depende del ciclo de vida de la actividad.

195

## Fragmentos (3)

### ■ Ciclo de vida



196

## Fragmentos (4)

### ■ Ciclo de vida

El ciclo de vida se inicia en el instante en que el fragmento se adjunta a la actividad que lo contiene (método *onAttach*).

A continuación sigue la creación del fragmento:

- *onCreate* (obligatorio): se invoca durante la creación y sirve para iniciar las variables esenciales para el funcionamiento del fragmento.
- *onCreateView* (obligatorio): se encarga de generar la vista del fragmento.

197

## Fragmentos (5)

### ■ Ciclo de vida

- *onActivityCreated*: se invoca cuando el fragmento termina de crearse y ya dispone de una vista y sirve para implementar la interacción con el usuario.

Y siguen los métodos que preparan el fragmento para su ejecución:

- *onStart*: el fragmento pasa a primer plano.
- *onResume*: indica que ya se puede interactuar con él.

198

## Fragmentos (6)

### ■ Ciclo de vida

A partir de éste momento el fragmento estaría en ejecución y se podría interactuar con él.

Si el fragmento pasa a estar inactivo:

- *onPause*: prepara el fragmento para la inactividad deshabilitando actualizaciones, *listener*, etc.
- *onStop*: el fragmento deja de estar en primer plano.

199

## Fragmentos (7)

### ■ Ciclo de vida

Por último si se destruye un fragmento:

- *onDestroyView*: destruye la vista del fragmento.
- *onDestroy*: destruye el fragmento.
- *onDetach*: desvincula el fragmento de la actividad que lo contiene.

200

## Fragmentos (8)

### ■ Crear un fragmento

Para crear un fragmento, hemos de crear una clase que herede de la clase *Fragment*, y para vincular el fragmento a la vista implementar el método *onCreateView* que recibirá 3 parámetros:

- *LayoutInflater inflater*: permite generar la vista a partir de un layout (XML).
- *ViewGroup container*: vista del padre en que se insertará el fragmento
- *Bundle savedInstanceState*: permite obtener el estado si se había guardado previamente.

201

## Fragmentos (9)

### ■ Crear un fragmento

```
public class DetailFragment extends Fragment {  
    @Override  
    public void onCreate (Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
    @Override  
    public View onCreateView (LayoutInflater inflater,  
        ViewGroup container, Bundle savedInstanceState) {  
        View v=inflater.inflate(R.layout.details,container,false);  
        return v;  
    }  
    @Override  
    public void onActivityCreated (Bundle savedInstanceState) {  
        super.onActivityCreated(savedInstanceState);  
    }  
}
```

202

## Fragmentos (10)

### ■ Tipos de fragmentos

- **Fragmentos estáticos:** se declaran en el *layout* de la actividad mediante la etiqueta "*fragment*" y no pueden agregarse, remplazarse o eliminarse de forma dinámica.
- **Fragmentos dinámicos:** se agregan de forma dinámica a la actividad, generalmente a través de un *FragmentManager*, y pueden agregarse, eliminarse o remplazarse durante la ejecución de la actividad.

203

## Fragmentos (11)

- Fragmentos Estáticos

## Práctica Guiada

204

## Fragmentos (12)

- Fragmentos Dinámicos

## Práctica Guiada

205

## Fragmentos (13)

### ■ Comunicación fragmento-actividad

Aunque los fragmentos son objetos independientes que pueden utilizarse en diferentes actividades, puede existir comunicación entre un fragmento y la actividad que lo contiene.

Para ello desde el fragmento se puede acceder a la actividad mediante el método *getActivity*.

Ejemplo:

```
Button botFragment1 =  
(Button) getActivity().findViewById(R.id.botFragment1);
```

206

## Fragmentos (14)

### ■ Comunicación fragmento-actividad

También desde una actividad se puede acceder a los fragmentos que contiene utilizando el método *findFragmentById*, al que se puede acceder desde el gestor de fragmentos: *getFragmentManager* o *getSupportFragmentManager*.

Ejemplo:

```
Fragment fragmento =  
getSupportFragmentManager().findFragmentById(R.id.contenedor);
```

207