



Tarea 9. La Clase `Fragment`.

CONCEPTOS.

Un fragmento forma parte de una interfaz de usuario o comportamiento de una aplicación, que se puede colocar en una `Activity` para permitir un mejor diseño modular de la actividad. Un fragmento es una especie de sub-`Activity`.

Las siguientes características de los fragmentos son importantes:

- Un fragmento tiene su propio diseño, comportamiento propio y sus propias devoluciones de llamada en su ciclo de vida.
- Se pueden añadir o eliminar fragmentos en una actividad, mientras se está ejecutando la actividad.
- Se pueden combinar varios fragmentos en una sola actividad para construir una interfaz de usuario multi-panel.
- Un fragmento puede utilizarse en varias actividades.
- El ciclo de vida del `Fragment` está estrechamente relacionado con el ciclo de vida de su actividad anfitrión, lo que significa que cuando la actividad se pausa, también se detendrán todos los fragmentos disponibles en la actividad.
- Un fragmento puede implantar un comportamiento que no tenga algún componente de interfaz de usuario.

Los fragmentos se crean heredando de la clase `Fragment` y se insertan en la plantilla de la actividad, declarando el fragmento en el archivo de la plantilla de la actividad, como un elemento `<fragment>`.

Con `Fragment` se tiene mayor flexibilidad y se elimina la limitación de tener una sola actividad en la pantalla a la vez. Ahora se puede tener una sola `Activity`, pero cada una se puede componer de varios fragmentos que tendrán su propio diseño, eventos y ciclo de vida completo.

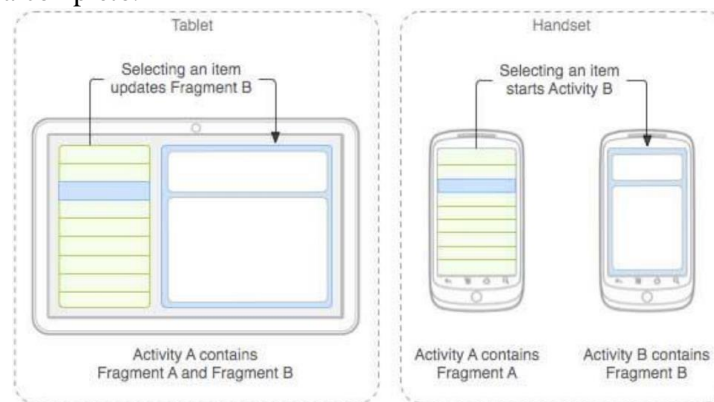


Figura 1. Una aplicación con una `Actividad A` que tiene dos fragmentos.

En la figura 1, una aplicación puede tener dos fragmentos en una `Actividad A`, cuando se ejecuta en una tableta. Sin embargo, en un celular no hay suficiente espacio para ambos fragmentos, por lo que la `Actividad A` incluye sólo al fragmento de la lista de artículos, y cuando el usuario selecciona un artículo, se inicia la `Actividad B`, que incluye el segundo fragmento para leer el artículo.

Uso de los fragmentos.

Para crear los fragmentos se siguen los siguientes pasos:

- Primero, decidir el número de fragmentos que desea utilizar en una actividad. Por ejemplo, utilizar dos fragmentos, en modo horizontal y vertical, en un dispositivo.
- Enseguida, basándose en el número de fragmentos, crear las clases que heredarán de la clase `Fragment`., la cual tiene la invocación a sus métodos. Se puede sobrescribir cualquiera de los métodos, según los requisitos.
- Por cada fragmento se tienen que crear archivos XML que tendrán el diseño para los fragmentos definidos.
- Finalmente, modificar el archivo de la actividad para definir la lógica real de la sustitución de fragmentos en base a los requisitos de la aplicación.



Los métodos más importantes se pueden sobrescribir son los siguientes:

`onCreate()` El sistema invoca a este método para crear al fragmento. Se deben inicializar los componentes esenciales del fragmento que se desea conservar cuando el fragmento se pausa, se detiene y se reanuda.

`onCreateView()` El sistema invoca a este método cuando se le solicite al fragmento dibujar su interfaz de usuario por primera vez. Para dibujar la interfaz de usuario del fragmento, se debe devolver un componente View desde este método que la raíz de la plantilla del fragmento. Se puede devolver un valor nulo si el fragmento no proporciona una interfaz de usuario.

`onPause()` El sistema llama a este método como la primera indicación de que el usuario abandona el fragmento. Esto es por lo general donde se debe realizar cualquier cambio que debe persistir más allá de la sesión actual del usuario.

DESARROLLO

EJEMPLO 1.

Diseñar una aplicación que posea dos fragmentos. Uno de los fragmentos tiene un modo horizontal y el segundo fragmento un modo vertical, de tal forma que al girar el teléfono se cambia del fragmento vertical al horizontal.

Paso 1. Crear un proyecto Fragmentos. En la carpeta `java/mipaquete` del repositorio, modificar el archivo `MainActivity.java` predeterminado, con el código siguiente.

```
import android.os.Bundle;
import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.res.Configuration;
import android.view.WindowManager;
import android.widget.HorizontalScrollView;
public class MainActivity extends Activity{
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        Configuration c = getResources().getConfiguration();
        FragmentManager fm = getFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();
        if(c.orientation == Configuration.ORIENTATION_LANDSCAPE) {
            HorizontalFragmento hf = new HorizontalFragmento();
            ft.replace(android.R.id.content, hf);
        }else{
            VerticalFragmento vf = new VerticalFragmento();
            ft.replace(android.R.id.content, vf);
        }
        ft.commit();
    }
}
```

Paso 2. Crear los archivos Java para cada fragmento.

En la misma carpeta anterior, crear el archivo Java para el primer fragmento `HorizontalFragmento.java`:

```
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class HorizontalFragmento extends Fragment{
    @Override
    public View onCreateView(LayoutInflater li, ViewGroup vg, Bundle b){
        return li.inflate(R.layout.horizontal_fragmento, vg, false);
    }
}
```



```
}
}
```

Paso 3. En la misma carpeta anterior, crear el archivo Java para el segundo fragmento VerticalFragmento.java:

```
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class VerticalFragmento extends Fragment{
    @Override
    public View onCreateView(LayoutInflater li, ViewGroup vg, Bundle b){
        return li.inflate(R.layout.vertical_fragmento, vg, false);
    }
}
```

Paso 4. Crear los archivos XML para cada fragmento.

En la carpeta res/layout, crear el XML para el primer fragmento horizontal_fragmento.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#8bae15" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/horizontal"
        android:textColor="#000000"
        android:textSize="30px" />
    <!-- Aquí se incluyen otros componentes GUI -->
</LinearLayout>
```

Paso 5. En la misma carpeta anterior, crear el XML para el segundo fragmento vertical_fragmento.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"
    android:background="#777777">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/vertical"
        android:textColor="#000000"
        android:textSize="20px" />
    <!-- Aquí se incluyen otros componentes GUI -->
</LinearLayout>
```

Paso 6. En la misma carpeta anterior, abrir y modificar el archivo activity_main.xml con el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

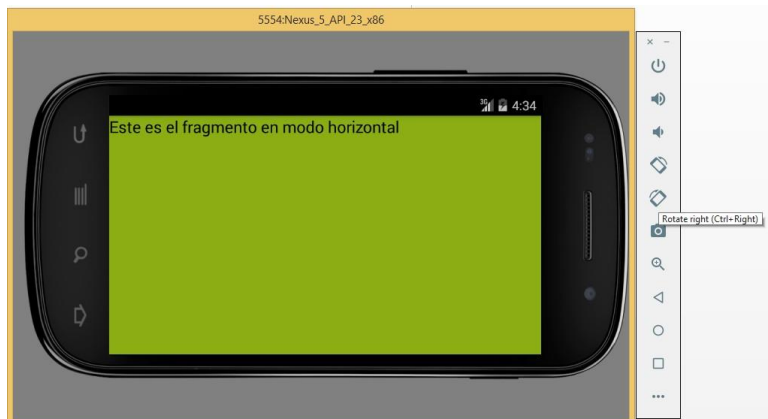


```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">
<fragment
    android:name="com.example.escom.fragmentos2.HorizontalFragmento"
    android:id="@+id/xhvf1"
    android:layout_weight="1"
    android:layout_width="0dp"
    android:layout_height="match_parent" />
<fragment
    android:name="com.example.escom.fragmentos2.VerticalFragmento"
    android:id="@+id/xvfv1"
    android:layout_weight="2"
    android:layout_width="0dp"
    android:layout_height="match_parent" />
</LinearLayout>
```

Paso 7. En la carpeta `res/values`, abrir y modificar el archivo `strings.xml` con el siguiente código:

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
    <string name="app_name">Fragmentos2</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Fragmentos2</string>
    <string name="horizontal">Este es el fragmento en modo horizontal</string>
    <string name="vertical">Este es el fragmento en modo vertical</string>
</resources>
```

Paso 8. Por último, ejecutar la aplicación. Observar que se puede utilizar una misma actividad, pero con diferentes interfaces gráficas de usuario utilizando diferentes fragmentos. Se pueden utilizar diferentes tipos de componentes GUI para diferentes interfaces gráficas de usuario en base a sus necesidades.. Observar en el menú derecho del emulador virtual, la opción para cambiar la orientación:



Nota. Generar un reporte con la imagen de la ejecución. Utilizar la sintaxis `AlumnoTarea9Grupo.pdf` y enviarla al sitio indicado por el profesor.