

Tecnologías y Desarrollo en Dispositivos Móviles

Apartado 11: Fragments

Autores:

Víctor M. Rivas Santos / Miguel Á. García Cumbreras
(Antonio Rueda Ruiz)

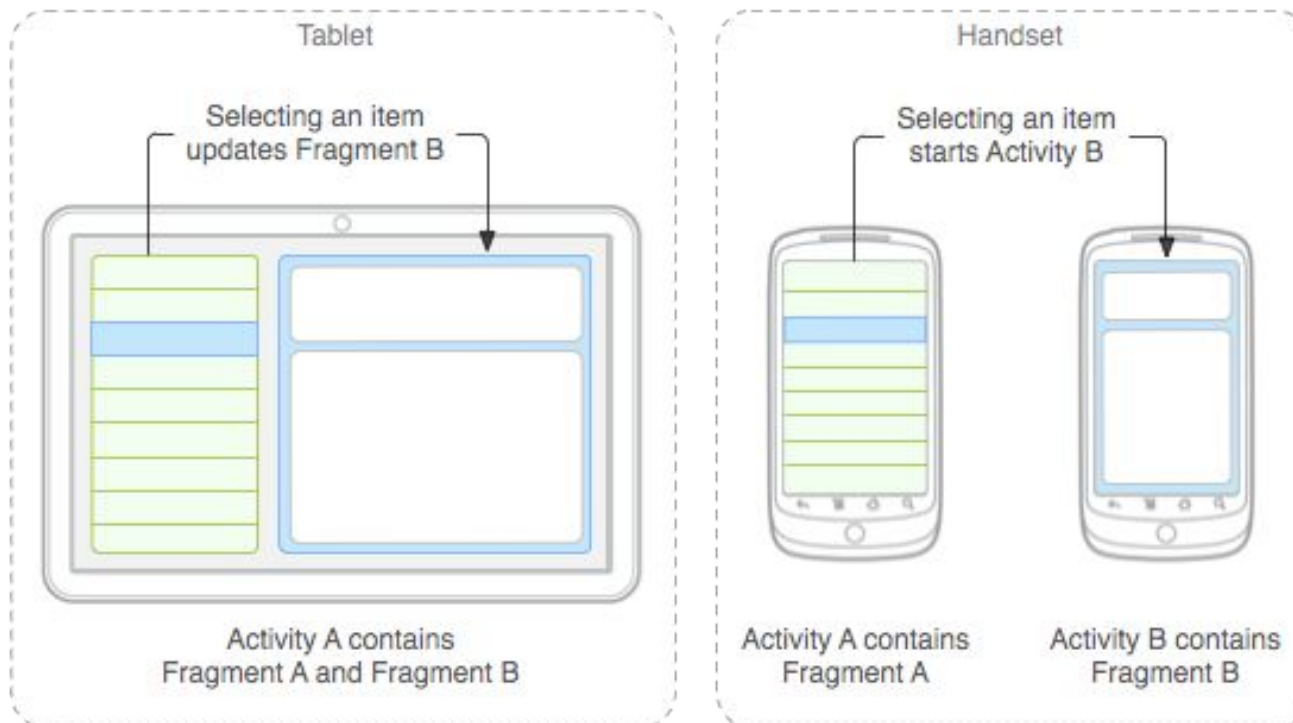
¿Qué son los fragments?

- Un fragment es una porción de IU que puede incrustarse en una actividad
- En cierta forma es una “subactividad” dentro de una actividad:
 - Tiene su ciclo de vida particular
 - Responde a sus propios eventos
 - Puede añadirse o eliminarse dinámicamente durante la ejecución de la actividad

<https://developer.android.com/guide/components/fragments.html>

¿Para qué se utilizan?

- Se utiliza fundamentalmente para:
 - Reutilizar porciones de UI en varias actividades
 - Mejorar la experiencia de usuario en pantallas grandes apaisadas (tablets)



Ciclo de vida de un fragmento

- Es similar (y en cierta forma dependiente) del ciclo de vida de una actividad.
 - Si la actividad contenedora es la activa, sus fragmentos pueden pasar cada uno por sus propios estados
 - Si la actividad contenedora entra en pausa o se destruye, los fragmentos también lo hacen

Algunos callback adicionales

- *onAttach*: se asocia el fragmento a una actividad
- *onActivityCreated*: cuando la actividad finaliza su método *onCreated*
- *onCreateView*: invocado al representar por primera vez el fragmento.
 - Permite devolver a la actividad contenedora el objeto View que representa el fragmento
- *onStart*: se ejecutará cuando termine el método homónimo de la actividad (es decir, cuando pase a visible)
- *OnDetach*: ejecutado al desagregar el fragmento de la actividad

Acceso a los elementos del fragmento desde el propio fragmento

- *findViewById* NO es un método del fragmento, sino del View que se crea cuando se “infla” o de la actividad que lo incrusta

FragmentoTest.java

```
public class FragmentoTest extends Fragment {
    View fragmento;


    @Override
    public View onCreateView(LayoutInflater inflater
        , ViewGroup container
        , Bundle savedInstanceState) {
        fragmento=inflater.inflate(R.layout.fragment_fragmento_test, container, false);
        ImageView imagen=(ImageView) fragmento.findViewById( R.id.idImagen);
        imagen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i=new Intent(Intent.ACTION_VIEW, Uri.parse( "http://duckduckgo.com?q="
                    + ((TextView) getActivity().findViewById(R.id.idTitulo)).getText()
                    + "+"
                    + ((TextView) fragmento.findViewById(R.id.idAutor)).getText()));
                startActivity(i);
            }
        });

        return fragmento;
    }
}
```


Creación de un fragmento (a)

- Android Studio nos permite crear fácilmente todo el código de un fragmento
 - Tanto el código Java con los *callback*
 - Como el recurso del *layout*
- (Botón derecho ratón) *New > Fragment > FragmentBlank*
 - O cualquier otro de los tipos deseados

Creación de un fragmento (b)

 **Configure Component**
Android Studio

Creates a blank fragment that is compatible back to API level 4.



Fragment Name:

Fragment Layout Name:

Target Source Set:

☒ Create layout XML?

☐ Include fragment factory methods?

☒ Include interface callbacks?

Creación de un fragmento (c)

res/layout/fragment_fragmento_test.xml

```
<FrameLayout 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"
    tools:context="com.example.vrivas.fragment_reutilizacion.FragmentoTest">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/hello_blank_fragment" />
</FrameLayout>
```

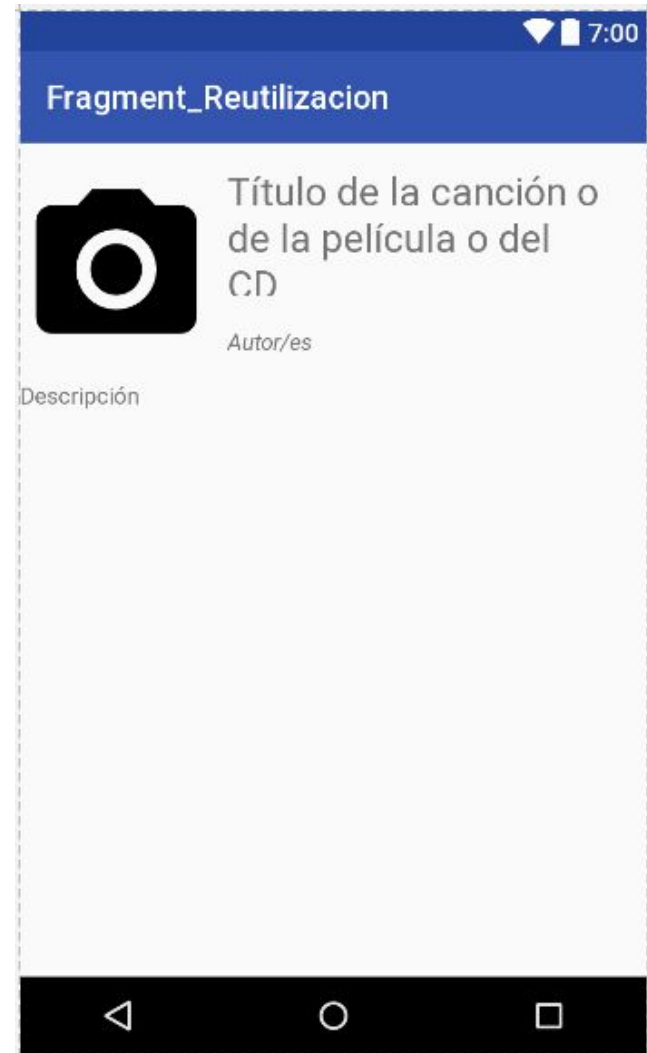
java/FragmentoTest.java

```
public class FragmentoTest extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater
                               , ViewGroup container
                               , Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_fragmento_test, container, false);
    } // Devuelve el View del fragmento para que pueda ser usado por la actividad
```

Creación de un fragmento (d)

- Una vez creada la estructura básica
 - Hemos de crear el diseño del *layout* del fragmento
 - Es conveniente PROPORCIONAR un identificador a dicho *layout*

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



Asociación del fragmento a la actividad

- Un fragmento puede ser incrustado en tantas actividades como deseemos
- Existen dos formas:
 - Programáticamente (lo veremos más tarde)
 - Mediante el diseñador gráfico
 - 1) **Abrir la actividad** en modo diseño
 - 2) Seleccionar en los widgets: ***Layout > Fragment***
 - 3) **Incluir** un *fragment*
 - 4) Se abre una ventana en la que podemos **seleccionar** nuestro fragmento diseñado
 - 5) **Asociarle un *android:id*** al fragmento dentro del XML de la actividad
 - 6) **Eliminar el lanzamiento de la excepción** en el método *onAttach* (dentro del código Java del fragmento)

Acceso a los elementos del fragmento desde la actividad

- Al usar un identificador dentro de la actividad para referenciar el fragmento, podemos acceder a sus elementos

```
public class LibroActivity extends FragmentActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_libro);  
  
        View fragment=(View) findViewById(R.id.frTest);  
  
        ((TextView) fragment.findViewById(R.id.idTitulo)).setText(getIntent().getStringExtra("TITULO"));  
  
        ((TextView) fragment.findViewById(R.id.idAutor)).setText(getIntent().getStringExtra("AUTOR"));  
  
        ((TextView) fragment.findViewById(R.id.idDescripcion))  
            .setText(getIntent().getStringExtra("DESCRIPCION"));  
  
        ((ImageView) fragment.findViewById(R.id.idImagen))  
            .setImageResource(getIntent().getIntExtra("IMAGEN", R.drawable.ic_menu_camera));  
    }  
}
```

Acceso a los elementos de la actividad desde el fragmento

- En cualquier momento, utilizando el método *getActivity()*

```
FragmentManager act=getActivity();  
TextView titulo=(TextView) act.findViewById(R.id.textMain);
```

- En el callback *onAttach(Context context)* del fragmento

```
public void onAttach(Context context) {  
    super.onAttach(context);  
  
    TextView texto=((Activity) context).findViewById(R.id.unTextoCualquiera);  
  
    if (texto!=null){  
        texto.setText("Se adjuntó el fragmento");  
    }  
  
}
```

Inserción de un fragmento programáticamente (a)

- Podemos insertar dinámicamente un fragmento dentro de cualquier layout que tengamos disponible
- En primer lugar, dejamos preparado el “contenedor” del fragmento

res/layout/contenedor_pelicula.xml (que a su vez se incluye en activity_pelicula.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.vrivas.fragment_reutilizacion.PeliculaActivity"
    tools:showIn="@layout/activity_pelicula"
    android:id="@+id/contenedor_pelicula">
    <!-- ESTE CONTENEDOR ESTÁ VACÍO -->
</android.support.constraint.ConstraintLayout>
```

Inserción de un fragmento programáticamente (b)

- Posteriormente, hacemos que se muestre insertándolo en dicho contenedor

java/PeliculaActivity.java

```
public class PeliculaActivity extends AppCompatActivity {
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_pelicula);
```

```
    FragmentManager fragmentManager=this.getSupportFragmentManager();  
    FragmentTransaction fragmentTransaction=fragmentManager.beginTransaction();  
    fragmentoTest nuevoFragmento=new FragmentoTest();  
    fragmentTransaction.add( R.id.contenedor_pelicula, nuevoFragmento );  
    fragmentTransaction.commit();  
}
```

```
@Override
```

```
protected void onResume() {  
    super.onResume();  
    View fragment=(View) this.findViewById(R.id.fragmento_test); //Nombre del id en XML  
    ((TextView) fragment.findViewById(R.id.idTitulo)).  
        .setText(getIntent().getStringExtra("TITULO"));  
    // Resto de vistas del fragmento
```

```
}
```

Lo explicamos en la próxima diapositiva ;)

Transacciones

- Una actividad puede agregar, quitar, reemplazar y modificar los fragmentos que contiene
- Los cambios pueden hacerse de uno en uno o en grupo
- Una transacción es un conjunto de cambios que la actividad confirma que desea hacer

java/activity_película.java

```
// 1. Recuperamos el FragmentManager
// a) Usando getSupportFragmentManager si es desde una actividad que hereda de
//     AppCompatActivity
// b) Usando getFragmentManager si es desde otro fragmento
FragmentManager fragmentManager=this.getSupportFragmentManager();

// 2. Creamos una FragmentTransaction a partir del FragmentManager
FragmentTransaction fragmentTransaction=fragmentManager.beginTransaction();

// 3. Añadimos, borramos, etc. el o los fragmentos deseados
FragmentoTest nuevoFragmento=new FragmentoTest();
fragmentTransaction.add( R.id.contenedor_película, nuevoFragmento );

// 4. Confirmamos la transacción
fragmentTransaction.commit();
```

Free as a Beer

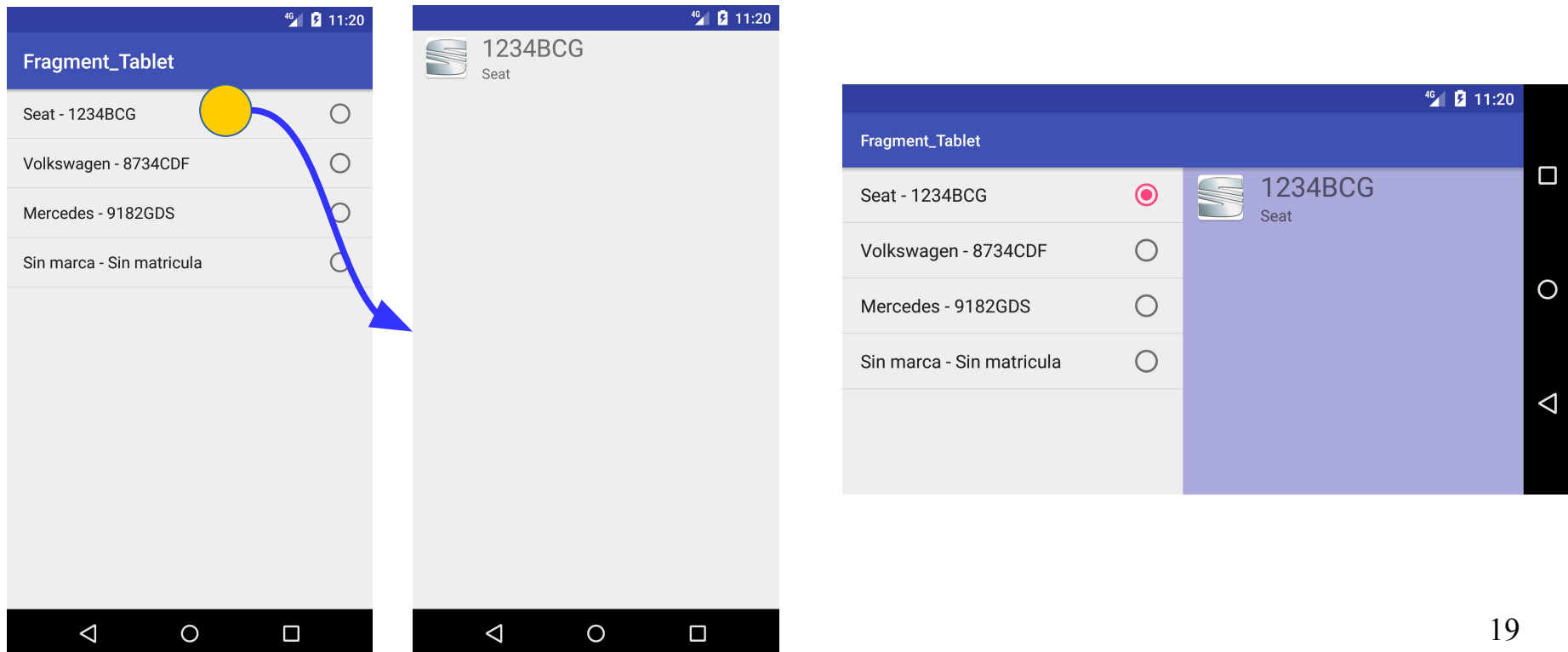
MTV: Practicar la reutilización de fragmentos

- Realiza una aplicación en la que mediante un GridLayout o TableLayout muestres distintas variedades de cerveza (o cualquier otra cosa que se te ocurra) usando una imagen o un texto.
- Al pinchar en cada una de ellas debe mostrarse información sobre la variedad en la que se ha pinchado en otra actividad que por defecto estará vacía.
 - Debe usarse por tanto un fragmento que se insertará y mostrará programáticamente
- Sube a la plataforma un documento PDF en el que se muestre:
 - Una captura de la pantalla con las imágenes de las variedades de cerveza y otra con la información de una de ellas
 - El código que permite que se inserte y muestre el fragmento
- *Bonus track:* puedes inspirarte escuchando la canción ["Free as a bird"](#) de los Beatles

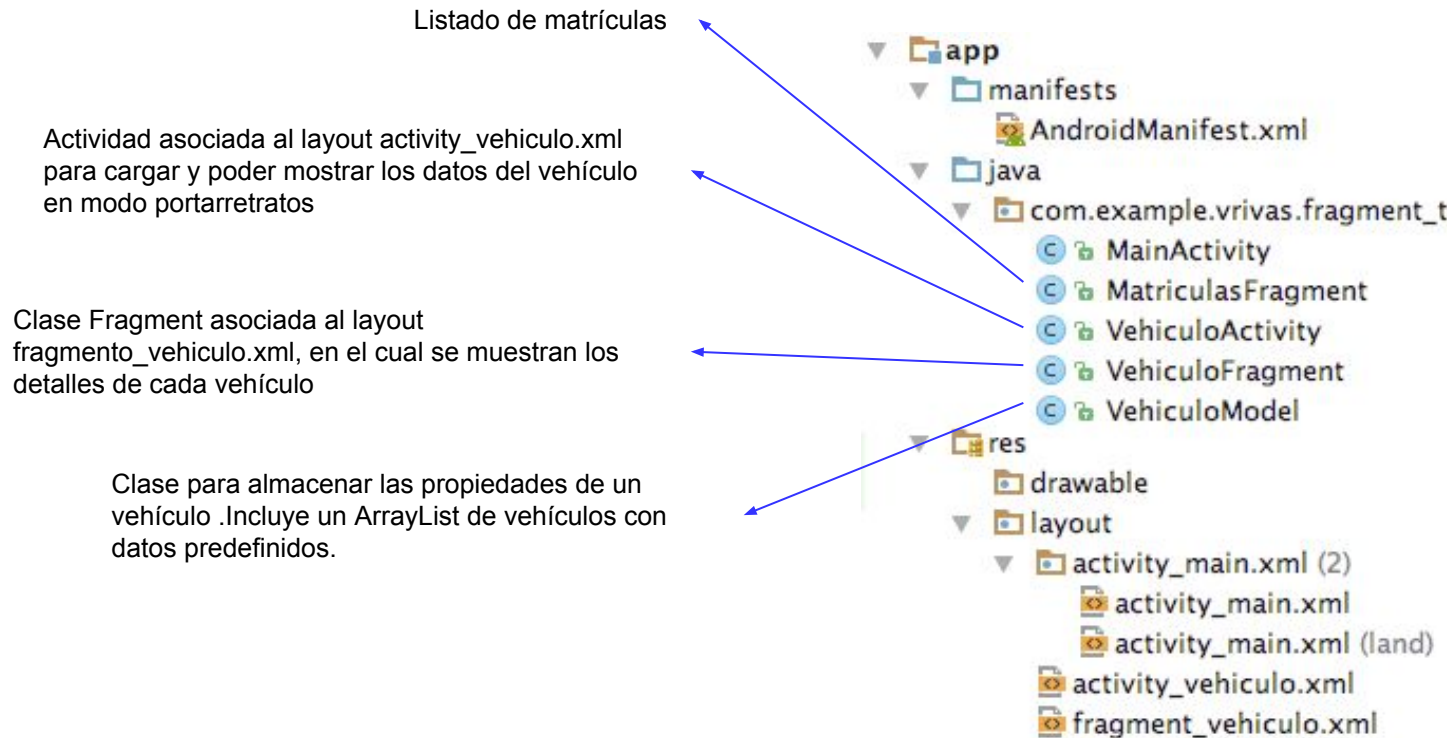
Vistas maestro/detalle

Ejemplo: Fragment_Tablet

- La reutilización de fragmentos es especialmente útil cuando queremos aprovechar el espacio ofrecido en orientaciones apaisadas



Organización del proyecto



Organización del proyecto

res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:orientation="horizontal">
    <fragment class="com.example.vrivas.fragment_tablet.MatriculasFragment"
        android:id="@+id/frMatriculas"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent"/>
    <!-- el id debe existir, aunque no se use -->
</LinearLayout>
```

res/layout-land/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:baselineAligned="false">
    <fragment class="com.example.vrivas.fragment_tablet.MatriculasFragment"
        android:id="@+id/frMatriculas"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent"/>
    <FrameLayout
        android:id="@+id/flVehiculo"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:background="@color/colorFragmentVehiculo">
    </FrameLayout>
</LinearLayout>
```

El fragmento con el ListView es el que “manda” (a)

```
public class MatriculasFragment extends ListFragment {
    boolean modoApaisado;
    int posicionActual=0;
    @Override
    public void onActivityCreated( Bundle savedInstanceState ){
        super.onActivityCreated(savedInstanceState);
        if( savedInstanceState!=null) {
            posicionActual=savedInstanceState.getInt("posicionActual",0);
        }
        View panelVehiculo=getActivity().findViewById(R.id.flVehiculo);
        modoApaisado =panelVehiculo!=null && panelVehiculo.getVisibility()==View.VISIBLE;
    }
    @Override
    public void onResume() {
        super.onResume();
        getListView().setChoiceMode(ListView.CHOICE_MODE_SINGLE);
        setListAdapter(new ArrayAdapter<VehiculoModel>(getActivity()
            , android.R.layout.simple_list_item_single_choice
            , vehiculos));
        if (modoApaisado) {
            muestraVehiculo(posicionActual);
        }
    }

    @Override
    public void onSaveInstanceState(Bundle estado) {
        super.onSaveInstanceState(estado);
        estado.putInt("posicionActual", posicionActual);
    }
}
```

El fragmento con el ListView es el que “manda” (b)

```
// MatriculasFragment.java
```

```
@Override
```

```
public void onItemClick( ListView l, View v, int position, long id){  
    muestraVehiculo(position);  
}
```

```
void muestraVehiculo(int posicion) {  
    posicionActual=posicion;  
    if ( modoApaisado ) {  
        getListView().setItemChecked(posicion, true);  
        VehiculoFragment detallesVehiculo=(VehiculoFragment) getFragmentManager()  
            .findFragmentById(R.id.frDetallesVehiculo);  
        if( detallesVehiculo==null || detallesVehiculo.getPosicionMostrada()!=posicion ) {  
            detallesVehiculo=VehiculoFragment.nuevaInstancia( posicion );  
            FragmentTransaction transaction=getFragmentManager().beginTransaction();  
            transaction.replace(R.id.flVehiculo, detallesVehiculo);  
            transaction.commit();  
        }  
    } else {  
        Intent i=new Intent(getActivity(), VehiculoActivity.class);  
        i.putExtra( "posicion", posicion);  
        startActivity(i);  
    }  
}
```

El fragmento con el ListView es el que “manda” (c)

```
// MatriculasFragment.java
```

```
// Este método será invocado cada vez que queramos eliminar un vehículo
```

```
void eliminarVehiculo( int posicion ) {  
    vehiculos.remove(posicion);  
    setListAdapter(new ArrayAdapter<VehiculoModel>(getActivity()  
        , android.R.layout.simple_list_item_single_choice  
        , vehiculos));  
}
```


El fragmento con los datos del vehículo los visualiza (a)

```
public class VehiculoFragment  
    extends Fragment {
```

```
    private OnClickListener actividadContenedora; // Imprescindible para comunicarse  
                                                    // con la actividad contenedora
```

```
    public VehiculoFragment() {} // Required empty public constructor
```

```
    public static VehiculoFragment nuevaInstancia( int posicion ) {  
        VehiculoFragment toRet=new VehiculoFragment();  
        Bundle args=new Bundle();  
        args.putInt( "posicion", posicion);  
        toRet.setArguments(args);  
        return toRet;  
    }
```

El fragmento con los datos del vehículo los visualiza (b)

@Override

```
public void onAttach(Context context) {  
    super.onAttach(context);  
    if (context instanceof OnClickDeleteButton) {  
        actividadContenedora = (OnClickDeleteButton) context;  
    } else {  
        throw new RuntimeException(context.toString()  
            + " must implement OnClickDeleteButton");  
    }  
}
```

@Override

```
public void onDetach() {  
    super.onDetach();  
    actividadContenedora = null;  
}
```

/**

** This interface **must be implemented by activities that contain this fragment** to allow an interaction in this fragment to be communicated to the activity and potentially other fragments contained in that activity.*

*/

```
public interface OnClickDeleteButton {  
    void onClickDeleteButton(int posicion);  
}
```

```
public int getPosicionMostrada() {  
    return getArguments().getInt("posicion", 0);  
}
```

La actividad principal es bastante simple

```
package com.example.vrivas.fragment_tablet;
```

```
import android.net.Uri;
```

```
import android.os.Bundle;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.widget.Toast;
```

```
public class MainActivity
```

```
    extends AppCompatActivity
```

```
    implements VehiculoFragment.OnClickDeleteButton {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public void onClickDeleteButton(int posicion){
```

```
        Toast.makeText(this, "Eliminando posicion" + posicion, Toast.LENGTH_SHORT).show();
```

```
        MatriculasFragment matriculasFragment=(MatriculasFragment) getSupportFragmentManager()  
            .findFragmentById(R.id.frMatriculas);
```

```
        matriculasFragment.eliminarVehiculo( posicion );
```

```
    }
```

```
}
```

Comunicación de eventos a la actividad contenedora

- El fragmento que muestra los detalles INCLUYE:
 - Una interfaz con sus respectivos métodos
 - Un atributo que la instancia al cual se asigna la actividad contenedora
 - Una llamada a los métodos de dicha interfaz a través del atributo
- Por tanto, la actividad contenedora ESTÁ OBLIGADA A implementar el método que responde al evento ocurrido

```
public class MainActivity
extends AppCompatActivity
implements VehiculoFragment.OnClickDeleteButton
{
    @Override
    public void onClickDeleteButton(int posicion){
        matriculasFragment
            .eliminarVehiculo( posicion );
    }
}
```

```
public class VehiculoFragment
extends Fragment {
    public interface OnClickDeleteButton {
        void onClickDeleteButton(int posicion);
    }

    private OnClickDeleteButton actividadContenedora;

    @Override
    public View onCreateView(LayoutInflater inflater
        , ViewGroup container
        , Bundle savedInstanceState) {
        myFab.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                actividadContenedora
                    .onClickDeleteButton(getPosicionMostrada());
            }
        });
        return toRet;
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnClickDeleteButton) {
            actividadContenedora = (OnClickDeleteButton) context;
        } else {
            //...
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        actividadContenedora = null;
    }
}
```

(Even more) Free as a Beer

MTV: Practicar la vista maestro/detalle

- Genera una copia del ejercicio anterior y conviértelo en una vista maestro/detalle como la que acabamos de explicar
- Intenta poner un botón flotante en el ListView para añadir una nueva variedad de cerveza, y mantén el botón para eliminar una variedad de cerveza en la vista de detalle de la misma.
- Sube a la plataforma un documento PDF en el que se recojan capturas de pantalla que muestren la funcionalidad de la aplicación tanto en formato portarretrato como en formato panorámico.