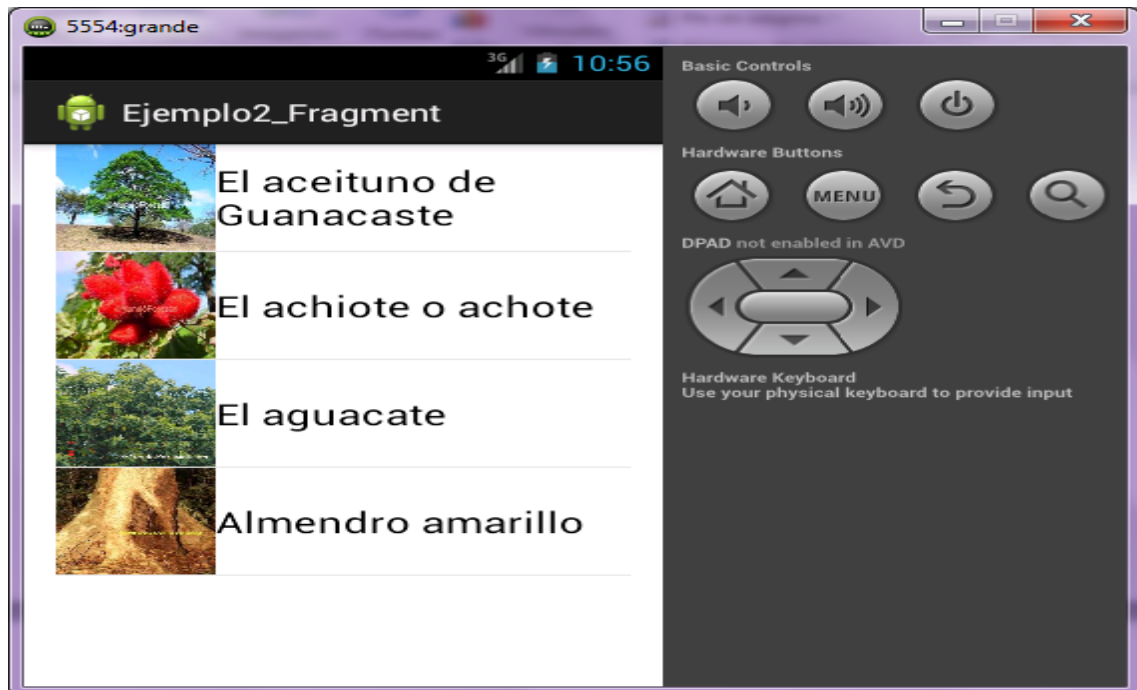


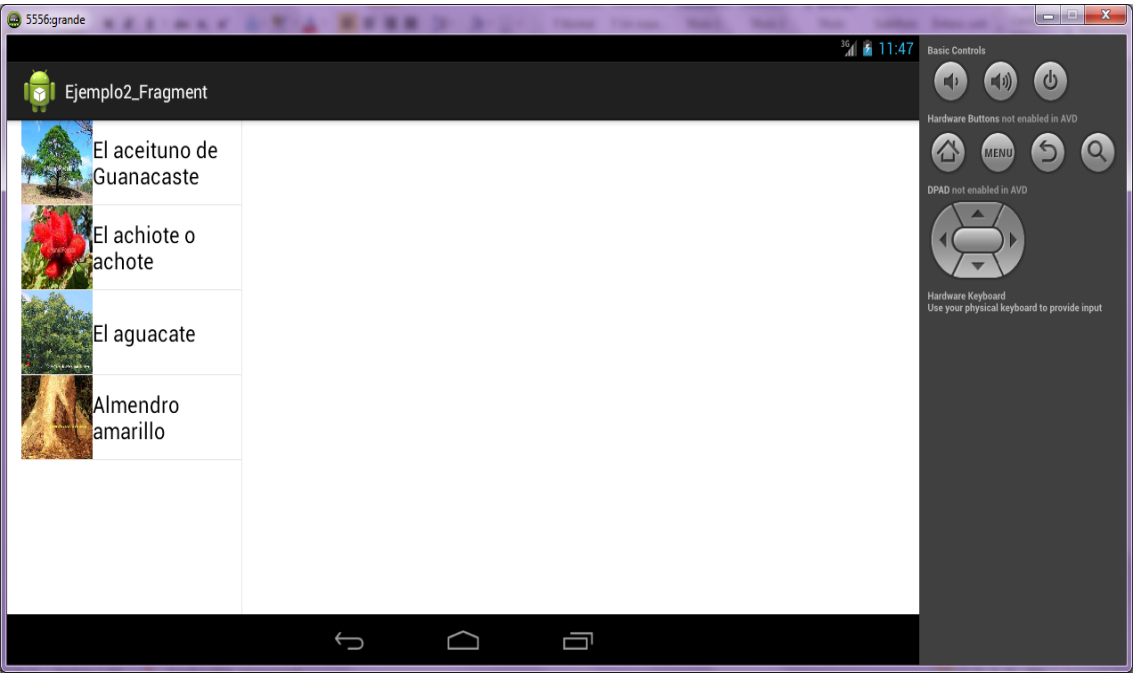
Ejemplo 2 Fragments

- Pasos para la creación del segundo ejemplo del taller.

Vista en smartphone

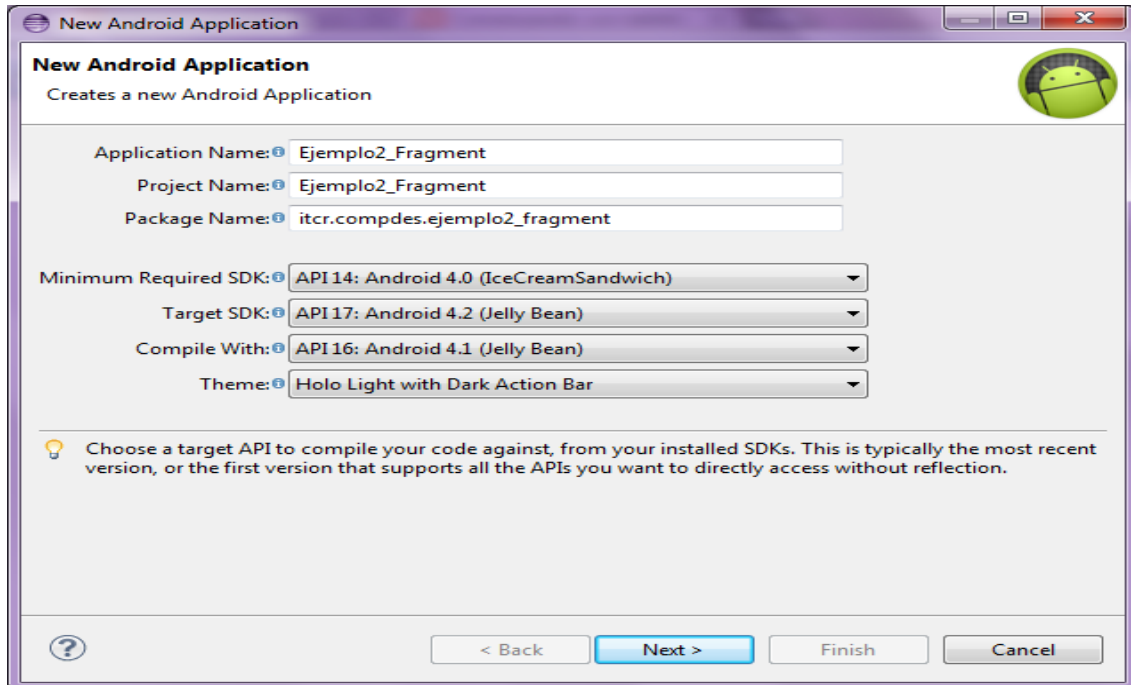


Vista en Tablet

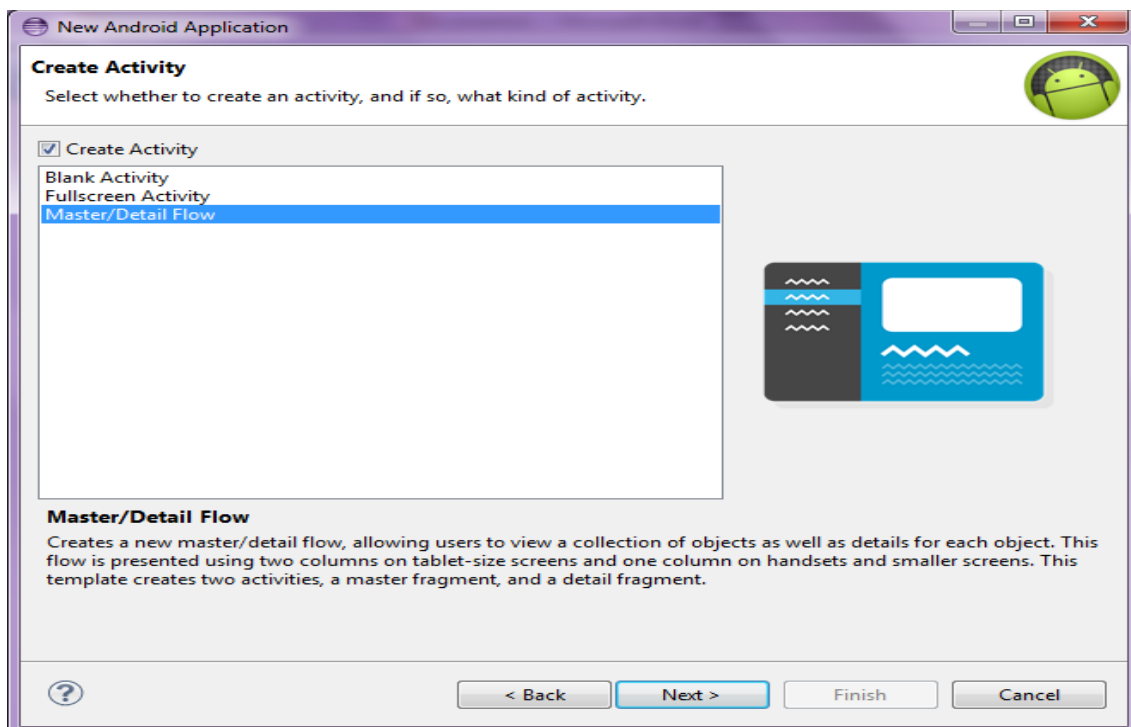


1. Creación del proyecto.

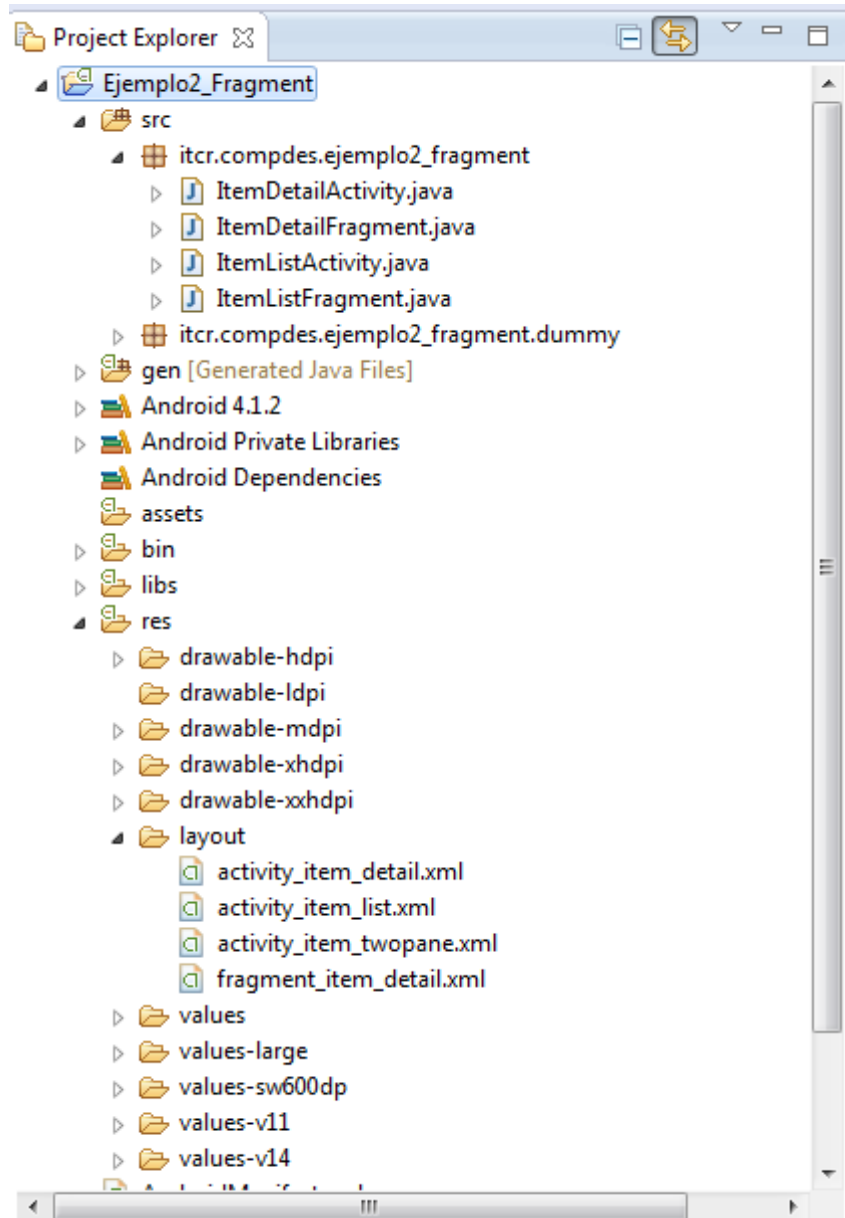
- a. Definimos el nombre del proyecto a realizar
- b. Definir el mínimo de SDK deberá ser el 4.0 Ice_CreamSandwich.
- c. Definir el target y Compilar con 4.2 Jelly Bean.



- d. Se debe de seleccionar Master/Detail Flow.
- e. Seleccionar siguiente y finalizar.



- f. Resultado del Proyecto y clases generadas automáticamente.
- g. Las clases generadas las modificaremos y agregaremos unas mas.



2. Creación de layout y modificación de las que fueron creadas automáticamente.

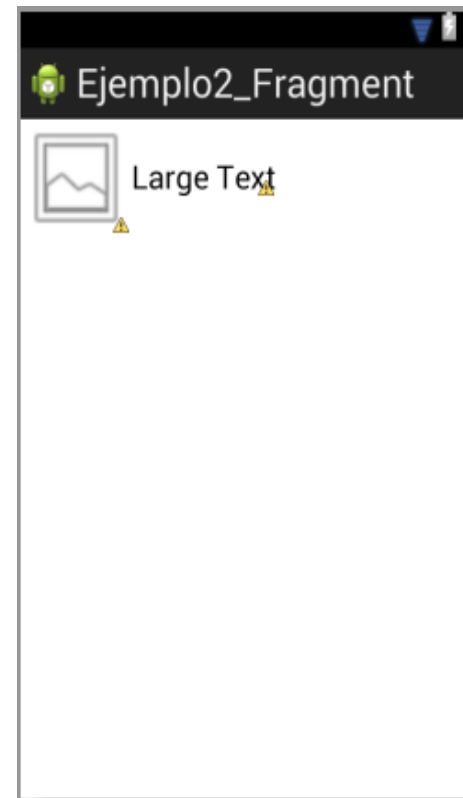
a. Crear el layout_elemento_listado.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center/left"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/imageView_imagen_miniatura"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:adjustViewBounds="true"
        android:scaleType="fitXY"
        android:contentDescription="Descripción del
contenido de la imagen"
        android:src="@android:drawable/ic_menu_gallery"
    />

    <TextView
        android:id="@+id/textView_titulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearance
Large" />

</LinearLayout>
```



b. Modificar el layout fragment_item_detail.xml

Original

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/item_detail"
    style="?android:attr/textAppearanceLarge"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:textIsSelectable="true"
    tools:context=".ItemDetailFragment" />
```

Cambiar el código por

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/an
    droid"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center/top"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView_superior"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Large Text"

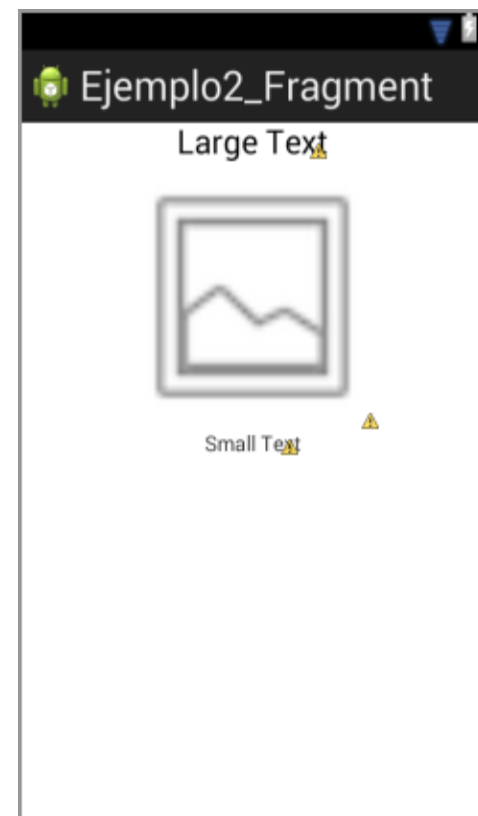
        android:textAppearance="?android:attr/textAppe
        aranceLarge" />

    <ImageView
        android:id="@+id/imageView_imagen"
        android:layout_width="178dp"
        android:layout_height="178dp"
        android:adjustViewBounds="true"
        android:contentDescription="Descripción del
        contenido de la imagen"
        android:scaleType="fitXY"

        android:src="@android:drawable/ic_menu_gallery" />

    <TextView
        android:id="@+id/textView_inferior"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Small Text"

        android:textAppearance="?android:attr/textAppearance
        Small" />
```



3. Creación y modificación de las clases creadas automáticamente.

a. Crear la clase lista adaptador

```
import java.util.ArrayList;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;

public abstract class Lista_adaptador extends BaseAdapter {

    private ArrayList<?> entradas;
    private int R_layout_IdView;
    private Context contexto;

    public Lista_adaptador(Context contexto, int R_layout_IdView,
        ArrayList<?> entradas) {
        super();
        this.contexto = contexto;
        this.entradas = entradas;
        this.R_layout_IdView = R_layout_IdView;
    }

    @Override
    public View getView(int posicion, View view, ViewGroup pariente) {
        if (view == null) {
            LayoutInflater vi = (LayoutInflater)
                contexto.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            view = vi.inflate(R_layout_IdView, null);
        }
        onEntrada (entradas.get(posicion), view);
        return view;
    }

    @Override
    public int getCount() {
        return entradas.size();
    }

    @Override
    public Object getItem(int posicion) {
        return entradas.get(posicion);
    }

    @Override
    public long getItemId(int posicion) {
        return posicion;
    }

    /** Devuelve cada una de las entradas con cada una de las vistas a la
     que debe de ser asociada
    public abstract void onEntrada (Object entrada, View view);
    {
        }
    }
}
```

- b. Crear la clase Lista_contenido, esta clase se encargara de tener los datos que se mostraran en el ejemplo.

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class Lista_contenido {
    // Donde se guardan las entradas de la lista.
    public static ArrayList<Lista_entrada> ENTRADAS_LISTA = new
    ArrayList<Lista_entrada>();
    // Donde se asigna el identificador a cada entrada de la lista
    public static Map<String, Lista_entrada> ENTRADAS_LISTA_HASHMAP = new
    HashMap<String, Lista_entrada>();
    // Creamos estáticamente las entradas
    static {
        aniadirEntrada(new Lista_entrada("0", R.drawable.im_aceituno,
        "El aceituno de Guanacaste", "Atractivo y noble árbol natural nativo desde
        Florida hasta Centroamérica y el Caribe."
        aniadirEntrada(new Lista_entrada("1", R.drawable.achiote, "El
        achiote o achote", "Es un arbusto grande de rápido crecimiento que a veces
        puede alcanzar el tamaño de un arbolito de unos 5 metros o un poco más."));
        aniadirEntrada(new Lista_entrada("2", R.drawable.im_aguacate,
        "El aguacate", "Se sabe que el aguacate es un árbol originario desde Yucatán
        hasta Costa Rica, aunque no se sabe exactamente de dónde porque no existen
        individuos silvestres. Aunque su reproducción es muy fácil por medio de
        semillas."));
        aniadirEntrada(new Lista_entrada("3",
        R.drawable.im_almendrodelapa, "Almendro amarillo", "Uno de los árboles en gran
        peligro extinción mejor conocidos en Costra Rica, gracias a un ave tropical
        que también se encuentra en gran peligro de extinción: la lapa verde (Ara
        ambigua ). El árbol proporciona alimento y refugio a la lapa verde."));
    }
    /** Añade una entrada a la lista
     * @param entrada Elemento que añadimos a la lista
     */
    private static void aniadirEntrada(Lista_entrada entrada) {
        ENTRADAS_LISTA.add(entrada);
        ENTRADAS_LISTA_HASHMAP.put(entrada.id, entrada);
    }
    /**
     * Representa una entrada del contenido de la lista
     */
    public static class Lista_entrada {
        public String id;
        public int idImagen;
        public String textoEncima;
        public String textoDebajo;

        public Lista_entrada (String id, int idImagen, String
        textoEncima, String textoDebajo) {
            this.id = id;
            this.idImagen = idImagen;
            this.textoEncima = textoEncima;
            this.textoDebajo = textoDebajo;
        }
    }
}
```


- c. Se debe de eliminar el paquete que se genera.

itcr.compdes.ejemplo2_fragment.dummy

- d. Se debe modificar la clase ItemListFragment, el código que está en amarillo se debe de modificar.

```
import android.app.Activity;
import android.os.Bundle;
import android.support.v4.app.ListFragment;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;

public class ItemListFragment extends ListFragment {

    private static final String STATE_ACTIVATED_POSITION =
"activated_position";
    private Callbacks mCallbacks = sDummyCallbacks;
    private int mActivatedPosition = ListView.INVALID_POSITION;

    public interface Callbacks {

        public void onItemSelected(String id);

    }

    private static Callbacks sDummyCallbacks = new Callbacks() {
        @Override
        public void onItemSelected(String id) {
        }
    };

    public ItemListFragment() {
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setListAdapter(new Lista_adaptador(getActivity(),
R.layout.layout_elemento_listado, Lista_contenido.ENTRADAS_LISTA){
            @Override
            public void onEntrada(Object entrada, View view) {
                if (entrada != null) {
                    TextView texto_superior_entrada = (TextView)
view.findViewById(R.id.textview_titulo);
                    if (texto_superior_entrada != null)

                        texto_superior_entrada.setText(((Lista_contenido.Lista_entrada)
entrada).textoEncima);
                    ImageView imagen_entrada = (ImageView)
view.findViewById(R.id.imageView_imagen_miniatura);
                    if (imagen_entrada != null)
                        imagen_entrada.setImageResource(((Lista_contenido.Lista_entr
ada) entrada).idImagen);
                }
            }
        });
    }
}
```

```

@Override
    public void onViewCreated(View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        // Restore the previously serialized activated item position.
        if (savedInstanceState != null
            && savedInstanceState.containsKey(STATE_ACTIVATED_POSITION))
        {
            setActivatedPosition(savedInstanceState
                .getInt(STATE_ACTIVATED_POSITION));
        }

@Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        if (!(activity instanceof Callbacks)) {
            throw new IllegalStateException(
                "Activity must implement fragment's callbacks.");
        }

        mCallbacks = (Callbacks) activity;
    }
@Override
    public void onDetach() {
        super.onDetach();

        // Reset the active callbacks interface to the dummy implementation.
        mCallbacks = sDummyCallbacks;
    }
@Override
    public void onItemClick(ListView listView, View view, int position,
        long id) {
        super.onItemClick(listView, view, position, id);

        // Notify the active callbacks interface (the activity, if the
        // fragment is attached to one) that an item has been selected.

        mCallbacks.onItemSelected(Lista_contenido.ENTRADAS_LISTA.get(position).id)
;

    }
@Override
    public void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        if (mActivatedPosition != ListView.INVALID_POSITION) {
            // Serialize and persist the activated item position.
            outState.putInt(STATE_ACTIVATED_POSITION, mActivatedPosition);
        }
    }
    public void setActivateOnItemClick(boolean activateOnItemClick) {
        // When setting CHOICE_MODE_SINGLE, ListView will automatically
        // give items the 'activated' state when touched.
        getListView().setChoiceMode(
            activateOnItemClick ? ListView.CHOICE_MODE_SINGLE
                : ListView.CHOICE_MODE_NONE);
    }
    private void setActivatedPosition(int position) {
        if (position == ListView.INVALID_POSITION) {
            getListView().setItemChecked(mActivatedPosition, false);
        } else {
            getListView().setItemChecked(position, true);
        }

        mActivatedPosition = position;
    }
}

```

e. Modificar la clase ItemDetailFragment

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

/**
 * A fragment representing a single Item detail screen. This fragment is either
 * contained in a {@link ItemListActivity} in two-pane mode (on tablets) or a
 * {@link ItemDetailActivity} on handsets.
 */
public class ItemDetailFragment extends Fragment {
    /**
     * The fragment argument representing the item ID that this fragment
     * represents.
     */
    public static final String ARG_ITEM_ID = "item_id";

    /**
     * The dummy content this fragment is presenting.
     */
    private Lista_contenido.Lista_entrada mItem;

    /**
     * Mandatory empty constructor for the fragment manager to instantiate the
     * fragment (e.g. upon screen orientation changes).
     */
    public ItemDetailFragment() {
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (getArguments().containsKey(ARG_ITEM_ID)) {
            mItem = Lista_contenido.ENTRADAS_LISTA_HASHMAP.get(
                getArguments().getString(ARG_ITEM_ID));
        }

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_item_detail,
            container, false);

        // Show the dummy content as text in a TextView.
        if (mItem != null) {
            ((TextView)
                rootView.findViewById(R.id.textView_superior)).setText(mItem.textoEncima);
            ((TextView)
                rootView.findViewById(R.id.textView_inferior)).setText(mItem.textoDebajo);
            ((ImageView)
                rootView.findViewById(R.id.imageView_imagen)).setImageResource(mItem.idImagen);
        }

        return rootView;
    }
}
```

f. Modificar la activity principal ItemListActivity

```
public class ItemListActivity extends FragmentActivity implements
    ItemListFragment.Callbacks {

    /**
     * Whether or not the activity is in two-pane mode, i.e. running on a tablet
     * device.
     */
    private boolean mTwoPane;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_item_list);

        if (findViewById(R.id.item_detail_container) != null) {
            // The detail container view will be present only in the
            // large-screen layouts (res/values-large and
            // res/values-sw600dp). If this view is present, then the
            // activity should be in two-pane mode.
            mTwoPane = true;

            // In two-pane mode, list items should be given the
            // 'activated' state when touched.
            ((ItemListFragment)
                getSupportFragmentManager().findFragmentById(
                    R.id.item_list)).setActivateOnItemClick(true);
        }

        // TODO: If exposing deep links into your app, handle intents here.
    }

    /**
     * Callback method from {@link ItemListFragment.Callbacks} indicating that
     * the item with the given ID was selected.
     */
    @Override
    public void onItemSelected(String id) {
        if (mTwoPane) {
            // In two-pane mode, show the detail view in this activity by
            // adding or replacing the detail fragment using a
            // fragment transaction.
            Bundle arguments = new Bundle();
            arguments.putString(ItemDetailFragment.ARG_ITEM_ID, id);
            ItemDetailFragment fragment = new ItemDetailFragment();
            fragment.setArguments(arguments);
            getSupportFragmentManager().beginTransaction()
                .replace(R.id.item_detail_container,
                    fragment).commit();
        } else {
            // In single-pane mode, simply start the detail activity
            // for the selected item ID.
            Intent detailIntent = new Intent(this,
                ItemDetailActivity.class);
            detailIntent.putExtra(ItemDetailFragment.ARG_ITEM_ID, id);
            startActivity(detailIntent);
        }
    }
}
```