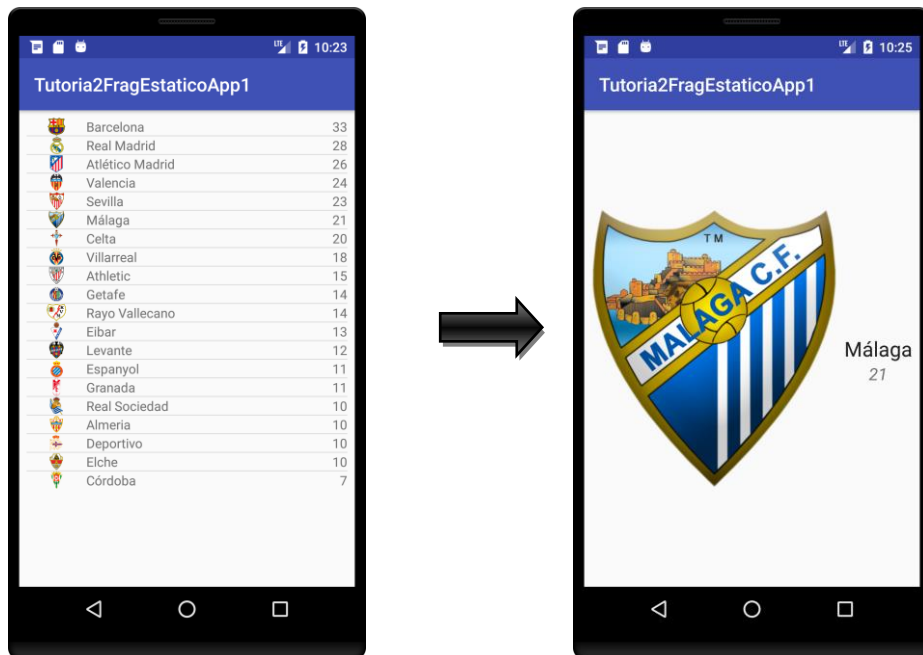


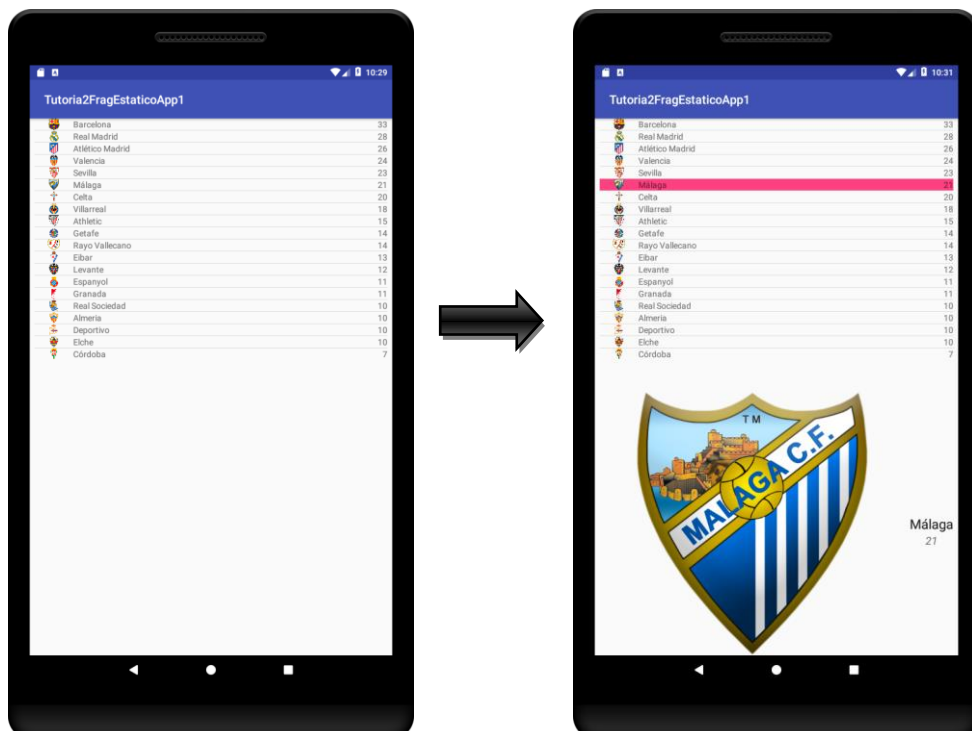
Ejercicio 1: Aplicación Tutoria2Frag-APP1-FragmentEstático

Crear una APP como la de **Tutoria2APP5**, el listado de fútbol, pero ahora usando **fragments estáticos**, de forma que se muestre así:

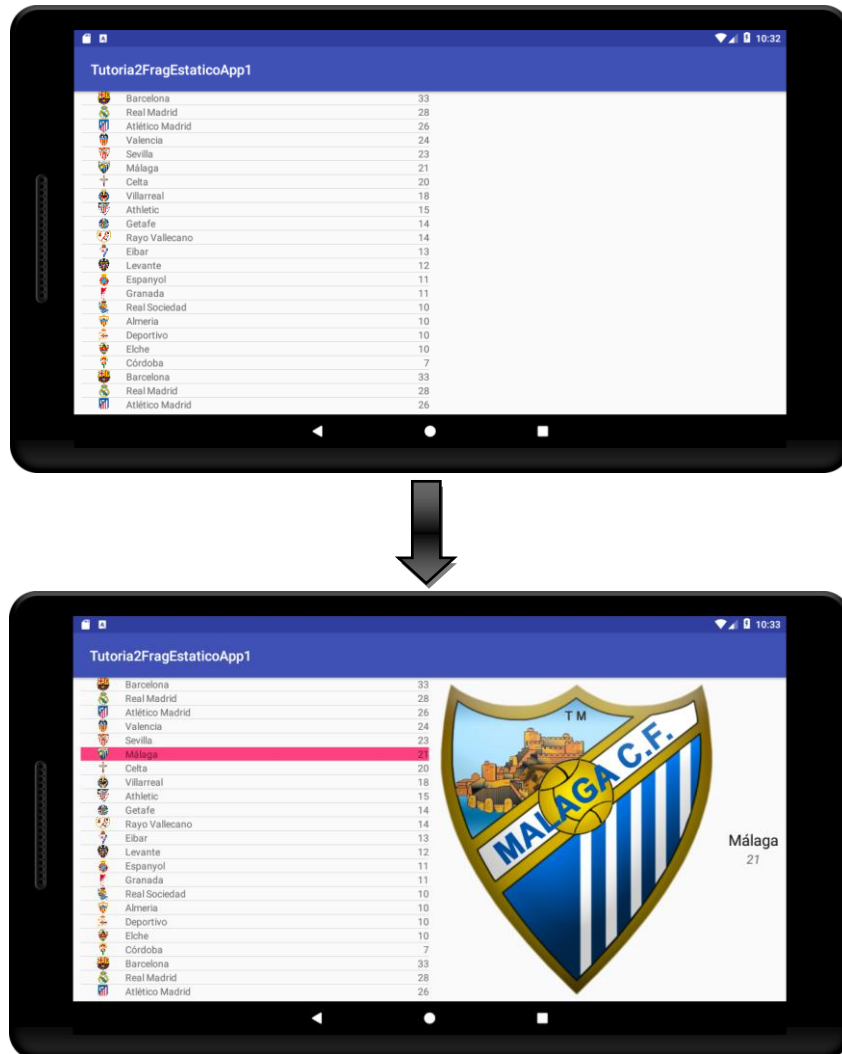
- Si estamos en un **teléfono**, veremos el listado de fútbol sólo, en un fragment y al pulsar sobre un equipo, se nos lanza otra Activity que contiene los detalles del equipo (*en otro fragment*).



- Si estamos en una **tablet vertical**, veremos en la parte superior el listado de equipos (fragment Listado) y en la parte inferior el detalle del equipo elegido (fragment detalle).



- Si estamos en una **tablet horizontal**, veremos en la parte izquierda el listado de equipos (fragment listado) y en la parte derecha el detalle del equipo elegido (fragment detalle).



PASOS DETALLADOS:

PASO 1: Crear el proyecto y copiar las imágenes de los escudos de los equipos en una carpeta drawable (que si no existe debemos crear en recursos).

NOTA: es importante que los nombres de los ficheros de imágenes y/o recursos no lleven mayúsculas.

PASO 2: Como vamos a manejar equipos de fútbol, crear la clase modelo **Equipo**, con cada uno de los atributos genéricos que van a formar el ítem. Crear el constructor, sus getter y setter.

```

public class Equipo {
    private String nombreEquipo;
    private Drawable imagenEscudo;
    private int puntos;

    //CONSTRUCTOR
    public Equipo(String nombreEquipo, Drawable imagenEscudo, int puntos)
    throws IllegalArgumentException {
        if (nombreEquipo==null || nombreEquipo.isEmpty()) {
            throw new IllegalArgumentException("ERROR: Nombre del equipo
vacío. Obligatorio un nombre");
        } else if (imagenEscudo == null) {
            throw new IllegalArgumentException("ERROR: Equipo sin foto en
el escudo. Marque la casilla 'Sin Foto' si no desea foto");
        } else {
            this.nombreEquipo = nombreEquipo;
            this.imagenEscudo = imagenEscudo;
            this.puntos = puntos;
        }
    }

    //GETTERS AND SETTERS
    public String getNombreEquipo() {
        return nombreEquipo;
    }

    public void setNombreEquipo(String nombreEquipo) {
        this.nombreEquipo = nombreEquipo;
    }

    public Drawable getImagenEscudo() {
        return imagenEscudo;
    }

    public void setImagenEscudo(Drawable imagenEscudo) {
        this.imagenEscudo = imagenEscudo;
    }

    public int getPuntos() {
        return puntos;
    }

    public void setPuntos(int puntos) {
        this.puntos = puntos;
    }
}

```

PASO 3: Creamos en la carpeta de recursos RES dentro de values un fichero XML llamado **arrays.xml** que contendrá la fuente de datos que usaremos para cargar el array en el adaptador. En este caso al hacer referencia a la imagen directamente, tendremos que indicar en el <ítem> del array de recurso @drawable/Barcelona, y así sucesivamente:

```

<resources>
    <array name="escudo_equipo">
        <item>@drawable/barcelona</item>
        <item>@drawable/madrid</item>
        <item>@drawable/atletico_madrid</item>
    </array>
</resources>

```

```
<item>@drawable/valencia</item>
<item>@drawable/sevilla</item>
<item>@drawable/malaga</item>
<item>@drawable/celta</item>
<item>@drawable/villa_real</item>
<item>@drawable/athletic</item>
<item>@drawable/getafe</item>
<item>@drawable/rayo_vallecano</item>
<item>@drawable/eibar</item>
<item>@drawable/levante</item>
<item>@drawable/espanyol</item>
<item>@drawable/granada</item>
<item>@drawable/real_sociedad</item>
<item>@drawable/almeria</item>
<item>@drawable/deportivo</item>
<item>@drawable/elche</item>
<item>@drawable/cordoba</item>
</array>

<string-array name="nombre_equipo">
    <item>Barcelona</item>
    <item>Real Madrid</item>
    <item>Atlético Madrid</item>
    <item>Valencia</item>
    <item>Sevilla</item>
    <item>Málaga</item>
    <item>Celta</item>
    <item>Villa Real</item>
    <item>Athletic</item>
    <item>Getafe</item>
    <item>Rayo Vallecano</item>
    <item>Eibar</item>
    <item>Levante</item>
    <item>Espanyol</item>
    <item>Granada</item>
    <item>Real Sociedad</item>
    <item>Almeria</item>
    <item>Deportivo</item>
    <item>Elche</item>
    <item>Córdoba</item>
</string-array>

<integer-array name="puntos_equipo">
    <item>33</item>
    <item>28</item>
    <item>26</item>
    <item>24</item>
    <item>23</item>
    <item>21</item>
    <item>20</item>
    <item>18</item>
    <item>15</item>
    <item>14</item>
    <item>14</item>
    <item>13</item>
    <item>12</item>
    <item>11</item>
    <item>11</item>
    <item>10</item>
    <item>10</item>
    <item>10</item>
```

```

        <item>10</item>
        <item>7</item>
    </integer-array>
</resources>

```

PASO 4: Como tenemos una ListView, creamos el **layout que es el que se repite en cada fila de la listview (da igual si hay fragments o no, al haber una ListView lo vamos a necesitar)** y que estará formado por los distintos elementos que forman cada fila de la listview **listaEquipos**. Se llamará **“ítem_lista_equipos.xml”**. En nuestro caso estará compuesto por una imagen (escudo), un textview para el nombre y otro textview para los puntos.



PASO 5: Como tenemos una ListView con un diseño personalizado, necesitamos un **Adaptador para enlazar nuestros datos (equipos) con la ListView**. Lo he llamado **EquipoAdaptador.java**:

```

public class EquipoAdaptador extends ArrayAdapter<Equipo> {
    private Context context;
    private ArrayList<Equipo> datos;

    public EquipoAdaptador(Context context, ArrayList<Equipo> datos) {
        //al super pasamos el contexto, el layout de cada item de la lista y
        // el array con los datos
        super(context, R.layout.item_lista_equipos, datos);

        //guardamos los parámetros en las variables de la clase
        this.context = context;
        this.datos = datos;
    }

    @NonNull
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // declaro variable ViewHolder
        EquipoHolder holder = null;
        // recojo el valor de convertView
        View item = convertView;

        if (item == null) {
            // Nunca se infló la vista aún.
            // En primer lugar "inflamos" una nueva vista
            LayoutInflater inflater = LayoutInflater.from(context);
            item = inflater.inflate(R.layout.item_lista_equipos, null);
            // Instanciamos el ViewHolder y almacenamos las referencias a sus
            // hijos
            holder = new EquipoHolder();
            holder.escudo = (ImageView) item.findViewById(R.id.ivEscudoItem);

```

```

        holder.nombreEquipo = (TextView)
item.findViewById(R.id.tvNombreEquipoItem);
        holder.puntosEquipo = (TextView)
item.findViewById(R.id.tvPuntosEquipoItem);
        // guardamos el holder en la vista item
        item.setTag(holder);
    } else {
        // obtenemos el ViewHolder guardado previamente, para tener
acceso
        // rápido a los TextView
        holder = (EquipoHolder) item.getTag();
    }

    //Asignamos los datos correspondientes la equipo de la posición
position

    // asignamos el escudo.
holder.escudo.setImageDrawable(getItem(position).getImagenEscudo());
    // asignamos el nombre
holder.nombreEquipo.setText(datos.get(position).getNombreEquipo());
    // asignamos los puntos

holder.puntosEquipo.setText(String.valueOf(datos.get(position).getPuntos()));

    // Devolvemos la vista para que se muestre en el ListView.
    return item;
}

static class EquipoHolder {
    ImageView escudo;
    TextView nombreEquipo;
    TextView puntosEquipo;
}
}

```

PASO 6: Creamos los FRAGMENTS:

Tendremos 2 fragments:

- Uno para la *Lista de Equipos*
- Otro para el *detalle del equipo elegido*

Al igual que con las activities, para **cada fragment** debemos diseñar:

- Su *layout XML*
- Su *código JAVA* para la lógica

Además, tendremos luego que diseñar para cada Activity que tengamos, sus **layouts, que serán:**

- **Para teléfonos:** tendremos 2 Activities:
 - La principal, cuyo layout contendrá el fragment listado
 - La que se lanza al elegir un equipo, que contendrá el fragment detalle.

- **Para tablets en horizontal:** tendremos una Activity principal cuyo layout incluye el fragment listado a la izquierda y el detalle a la derecha.
- **Para tablets en vertical:** tendremos una Activity principal cuyo layout incluye el fragment listado arriba y el detalle abajo.

Vamos con los fragments:

PASO 6.1: Creamos el PRIMER FRAGMENT: el **listado**

Contendrá un **ListView** para mostrar los equipos.

El layout lo llamaremos: **fragment_teams_list.xml**

```
<?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="vertical"
    tools:context=".TeamFragmentList"
    tools:listitem="@layout/item_lista_equipos" >
    <ListView
        android:id="@+id/lvListadoEquipos"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>
</LinearLayout>
```

Su código java se llamará **"TeamFragmentList.java"** y será:

```
/**
 * A fragment representing a list of Items.
 * <p/>
 * Activities containing this fragment MUST implement the {@link
 * OnListFragmentSelectionListener}
 * interface.
 */

public class TeamFragmentList extends Fragment {

    //listener
    private OnListFragmentSelectionListener mListener;
    //listview
    private ListView lstTeamsList;

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

```

    public TeamFragmentList() {
    }

    //PATRÓN FACTORÍA USADO PARA INSTANCIAR FRAGMENTS QUE LLEVAN
    PARÁMETROS O PUEDEN LLEVARLOS
    public static TeamFragmentList newInstance(int valor) {
        TeamFragmentList fragment = new TeamFragmentList();
        /* PARAMETROS A PASARLE
        Bundle args = new Bundle();
        args.putInt("CLAVE", VALOR);
        //le paso los parámetros al fragment con setArguments
        (equivale a putExtras en un Intent)
        fragment.setArguments(args);
        */
        return fragment;
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnListFragmentSelectionListener) {
            mListener = (OnListFragmentSelectionListener) context;
        } else {
            throw new RuntimeException(context.toString()
                + " must implement
OnListFragmentSelectionListener");
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
        container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_teams_list,
            container, false);
    }

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

        lstTeamsList = (ListView)
        getView().findViewById(R.id.lvListadoEquipos);
        // Set the list choice mode to allow only one selection at a
        time
        lstTeamsList.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
        lstTeamsList.setSelector(R.color.colorAccent); //resalta
        opción elegida en la lista
        //listaTeamArrayList contiene los datos de todos los equipos y
        se encuentra
        //en forma de Lista estática en la Activity
        lstTeamsList.setAdapter(new
        EquipoAdaptador(this.getActivity(),
            MainActivity.teamArrayList));
        //asigno el oyente a la listview
        lstTeamsList.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {

```



```

        @Override
        public void onItemClick(AdapterView<?> list, View view,
int pos, long id) {
            if (mListener != null) {
                // Inform the MainActivity that the item in
                position pos has been selected
                mListener.onListFragmentSelection(pos);
            }
        }
    });

}

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

// Callback interface that allows this Fragment to notify the
// MainActivity when
// user clicks on a List Item
public interface OnListFragmentSelectionListener {
    void onListFragmentSelection(int posEquipo);
}
}

```

PASO 6.2: Creamos el SEGUNDO FRAGMENT: el detalle del equipo

Contendrá un **ImageView** para mostrar el escudo del equipo, y dos **TextView** para mostrar el nombre del equipo y los puntos.

El layout lo llamaremos: **fragment_team_detail.xml**

```

<RelativeLayout
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=".TeamDetailFragment">

    <ImageView
        android:id="@+id/ivEscudoDetailFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_toLeftOf="@+id/tvNombreEquipoDetailFragment"
        android:adjustViewBounds="true"
        android:src="@mipmap/ic_launcher" />

    <TextView
        android:id="@+id/tvNombreEquipoDetailFragment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"

```

```

        android:layout_marginLeft="8dp"
        android:text="Nombre equipo"

    android:textAppearance="@style/Base.TextAppearance.AppCompat.Large" />

    <TextView
        android:id="@+id/tvPuntosEquipoDetailFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/tvNombreEquipoDetailFragment"
        android:layout_alignStart="@+id/tvNombreEquipoDetailFragment"
        android:layout_below="@+id/tvNombreEquipoDetailFragment"
        android:gravity="center_horizontal"
        android:text="Puntos"

    android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"
        android:textStyle="italic" />
</RelativeLayout>

```

Su código java se llamará: **TeamDetailFragment.java**

```

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link TeamDetailFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class TeamDetailFragment extends Fragment {

    public TeamDetailFragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @return A new instance of fragment TeamDetailFragment.
     */
    public static TeamDetailFragment newInstance(String param1) {
        TeamDetailFragment fragment = new TeamDetailFragment();
        /*
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        fragment.setArguments(args);
        */
        return fragment;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_team_detail, container,
false);
    }

```

```

/**
 * Método que se usará desde la activity para mostrar los datos del
 * equipo que
 * se haya elegido
 *
 * @param equipo el equipo elegido
 */
public void showTeam(Equipo equipo) {
    ImageView ivEscudo=(ImageView)
    getView().findViewById(R.id.ivEscudoDetailFragment);
    TextView tvEquipo =
        (TextView)
    getView().findViewById(R.id.tvNombreEquipoDetailFragment);
    TextView
    tvPuntos=(TextView) getView().findViewById(R.id.tvPuntosEquipoDetailFragment);

    ivEscudo.setImageDrawable(equipo.getImagenEscudo());
    tvEquipo.setText(equipo.getNombreEquipo());
    tvPuntos.setText(equipo.getPuntos());
}
}

```

PASO 7: Creamos ahora la Activity principal. Sólo sus layouts.

Teniendo en cuenta que tendrá **3 layouts: 1 por defecto (en teléfono vertical) y dos alternativos: para tablet vertical y para tablet horizontal.**

El código java será único, pues es una activity: MainActivity.

PASO 7.1: Creamos el layout por defecto (para teléfono): será sólo el fragment listado de equipos. Se guarda en **/res/layout/activity_main.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"

tools:context="com.android.alejandra.tutoria2fragestaticoapp1.MainActivity">

    <fragment
        android:id="@+id/fragment"
        android:name=".TeamFragmentList"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"

```

```

        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>

```

PASO 7.2: Creamos el layout para tablet grandes y horizontales, que tendrá un listado a la izquierda y a la derecha el detalle del equipo. Se guarda en `/res/layout-sw600dp-land/activity_main.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"

tools:context="com.android.alejandra.tutoria2fragestaticoapp1.MainActi
vity">

    <fragment
        android:id="@+id/fragmentListado"

android:name="com.android.alejandra.tutoria2fragestaticoapp1.TeamFragm
entList"

        android:layout_width="500dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="16dp"
        android:layout_marginStart="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toLeftOf="@+id/fragmentDetail"
        app:layout_constraintTop_toTopOf="parent" />

    <fragment
        android:id="@+id/fragmentDetail"

android:name="com.android.alejandra.tutoria2fragestaticoapp1.TeamDetai
lFragment"

        android:layout_width="500dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="0dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toRightOf="@+id/fragmentListado"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />
</android.support.constraint.ConstraintLayout>

```

PASO 7.3: Creamos el layout para tablet grandes y en vertical, que tendrá un listado arriba y abajo el detalle del equipo. Se guarda en **/res/layout-sw600dp-port/activity_main.xml**:

```
<?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="vertical"

    tools:context="com.android.alejandra.tutoria2fragestaticoapp1.MainActi
vity">

    <fragment
        android:id="@+id/fragmentListado"

        android:name="com.android.alejandra.tutoria2fragestaticoapp1.TeamFragm
entList"

        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_weight="1" />

    <fragment
        android:id="@+id/fragmentDetail"

        android:name="com.android.alejandra.tutoria2fragestaticoapp1.TeamDetai
lFragment"

        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_weight="1" />
</LinearLayout>
```

PASO 8: Creamos ahora la Activity que se mostrará cuando estemos en un teléfono y se elija un equipo de la lista de equipos. Será una activity que muestra el detalle de ese equipo.

Diseñaremos ahora sólo su layout, que contendrá el fragmen detalle de equipo.

Está en /res/layout y se llamará: **activity_detail_team.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"

tools:context="com.android.alejandra.tutoria2fragestaticoapp1.MainActivity">

    <fragment
        android:id="@+id/fragmentDetail"

android:name="com.android.alejandra.tutoria2fragestaticoapp1.TeamFragmentList"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>

```

PASO 9: Escribimos la lógica de la MainActivity.java:

- Como el fragment TeamFragmentList incorpora un Listener, la Activity debe implementarlo para poder usarlo:

```

public class MainActivity extends AppCompatActivity implements
TeamFragmentList.OnListFragmentSelectionListener {

```

- Tendremos que crear un array de equipos.
Crearemos un ArrayList<Equipo>.

```

static ArrayList<Equipo> teamArrayList=new ArrayList<Equipo>();

```

- Crear un método que rellene el array con la información de cada equipo

En el onCreate:

```

// cargamos los valores en el arrayList
rellenarArrayList();

```

Método que añadiremos para rellenar el ArrayList arrayEquipos:

```

private void rellenarArrayList() {
    //recupero el array de los escudos de los recursos
    TypedArray arrayTypedEscudos =
        getResources().obtainTypedArray(R.array.escudo_equipo);

    //recupero el array de los nombres de equipos
    String[] arrayNombreEquipos =
        getResources().getStringArray(R.array.nombre_equipo);

    //recupero el array de los puntos de equipos
    int[] arrayPuntosEquipos =
        getResources().getIntArray(R.array.puntos_equipo);

    //relleno el arraylist
    for (int i = 0; i < arrayNombreEquipos.length; i++) {
        arrayEquipos.add(new Equipo(arrayNombreEquipos[i],
                                     arrayTypedEscudos.getDrawable(i),
                                     arrayPuntosEquipos[i]));
    }
}

```

- Implementamos el método del Listener `TeamFragmentManager.OnListFragmentSelectionListener` que será llamado cuando el usuario pulse en la lista de equipos, sobre uno:

```

@Override
public void onListFragmentSelection(int posEquipo) {

    //intentamos obtener el segundo fragment, el de detalle
    TeamDetailFragment teamDetailFragment = (TeamDetailFragment)
    getFragmentManager().findFragmentById(R.id.fragmentDetail);

    if (teamDetailFragment != null) {
        //hay dos paneles
        //muestro el equipo en el detailFragment
        teamDetailFragment.showTeam(teamArrayList.get(posEquipo));
    } else {
        //hay un solo panel
        //debo lanzar la actividad DetailTeamActivity
        Intent i = new Intent(this, DetailTeamActivity.class);
        i.putExtra(DetailTeamActivity.EXTRA_POS_EQUIPO,
posEquipo);
        startActivity(i);
    }
}

```

El código completo del MainActivity será:

```

public class MainActivity extends AppCompatActivity implements
TeamFragmentManager.OnListFragmentSelectionListener {

    static ArrayList<Equipo> teamArrayList = new ArrayList<Equipo>();

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

        // cargamos los valores en el arrayList
        rellenarArrayList();

    }

    private void rellenarArrayList() {
        //recupero el array de los escudos de los recursos
        TypedArray arrayTypedEscudos =
getResources().obtainTypedArray(R.array.escudo_equipo);

        //recupero el array de los nombres de equipos
        String[] arrayNombreEquipos =
getResources().getStringArray(R.array.nombre_equipo);

        //recupero el array de los puntos de equipos
        int[] arrayPuntosEquipos =
getResources().getIntArray(R.array.puntos_equipo);

        //relleno el arrayList
        for (int i = 0; i < arrayNombreEquipos.length; i++) {
            teamArrayList.add(new Equipo(arrayNombreEquipos[i],
arrayTypedEscudos.getDrawable(i),
arrayPuntosEquipos[i]));
        }

    }

    @Override
    public void onListFragmentSelection(int posEquipo) {

        //intentamos obtener el segundo fragment, el de detalle
        TeamDetailFragment teamDetailFragment = (TeamDetailFragment)
getFragmentManager().findFragmentById(R.id.fragmentDetail);

        if (teamDetailFragment != null) {
            //hay dos paneles
            //muestro el equipo en el detailFragment
            teamDetailFragment.showTeam(teamArrayList.get(posEquipo));
        } else {
            //hay un solo panel
            //debo lanzar la actividad DetailTeamActivity
            Intent i = new Intent(this, DetailTeamActivity.class);
            i.putExtra(DetailTeamActivity.EXTRA_POS_EQUIPO,
posEquipo);
            startActivity(i);
        }
    }
}

```



```
    }
}
```

PASO 10: Escribimos la lógica de la **DetailTeamActivity.java**, que será la que se llame cuando estemos en un teléfono y se pulse sobre un equipo:

```
public class DetailTeamActivity extends AppCompatActivity {
    public static final String EXTRA_POS_EQUIPO =
        "com.android.alejandra.tutoria2fragestaticoappl.EXTRA_POS_EQUIPO";

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

        //obtengo la pos del equipo a mostrar
        int pos = getIntent().getIntExtra(EXTRA_POS_EQUIPO, 0);

        //obtengo el fragment del layout
        TeamDetailFragment teamDetailFragment = (TeamDetailFragment)
            getSupportFragmentManager().findFragmentById(R.id.fragmentDetail);
        //muestro el equipo

        teamDetailFragment.showTeam(MainActivity.teamArrayList.get(pos));
    }
}
```

Ejercicio 2: Aplicación Tutoria2Frag-APP2-FragmentDinámico

Modificar la APP anterior para realizarla ahora usando **fragments dinámicos**:

PASOS A SEGUIR:

PASO 1: Modificar los layouts de las Activities cambiando las etiquetas **<fragment>** por **<FrameLayout>**, puesto que ahora, los fragments no se incrustan en el layout, sino que se añaden a **CONTENEDORES en tiempo de ejecución**.

Nos quedan los layouts de las Activities así:

LAYOUT ACTIVITY DETALLE:

```
<?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"

    tools:context="com.android.alejandra.tutoria2fragdinamicoapp2.
MainActivity">

    <FrameLayout
        android:id="@+id/containerDetail"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

LAYOUT MAINACTIVITY POR DEFECTO (TFNO):

```
<?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"

    tools:context="com.android.alejandra.tutoria2fragdinamicoapp2.M
ainActivity">

    <FrameLayout
        android:id="@+id/containerListFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

LAYOUT MAINACTIVITY TABLET VERTICAL:

```

<?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="vertical"

    tools:context="com.android.alejandra.tutoria2fragdinamicoapp2.
MainActivity">

    <FrameLayout
        android:id="@+id/containerListFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_weight="1" />

    <FrameLayout
        android:id="@+id/containerDetail"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_weight="1" />
</LinearLayout>

```

LAYOUT MAINACTIVITY TABLET HORIZONTAL:

```

<?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"

    tools:context="com.android.alejandra.tutoria2fragdinamicoapp2.Ma
inActivity">

    <FrameLayout
        android:id="@+id/containerListFragment"
        android:layout_width="500dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"

```

```

        android:layout_marginRight="16dp"
        android:layout_marginStart="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"

        app:layout_constraintRight_toLeftOf="@+id/fragmentDetail"
        app:layout_constraintTop_toTopOf="parent" />

<FrameLayout
    android:id="@+id/containerDetail"
    android:layout_width="500dp"
    android:layout_height="0dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.0"

    app:layout_constraintLeft_toRightOf="@+id/fragmentListado"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
</android.support.constraint.ConstraintLayout>

```

PASO 2: Modificar el fragment **TeamDetailFragment** para evitar cometer el error típico al llamar a `showTeam` y que no esté creada la vista. Para ello, en el método factoría `newInstance` le pasaré la `pos` como parámetro y la recogeré en el `onCreate` del fragment:

```

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link TeamDetailFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class TeamDetailFragment extends Fragment {
    private static final String ARG_POS_EQUIPO = "pos_equipo";
    private int posEquipoMostrar;
    //views xml
    private ImageView ivEscudo;
    private TextView tvEquipo;
    private TextView tvPuntos;

    public TeamDetailFragment() {
        // Required empty public constructor
    }

```

```

/**
 * Use this factory method to create a new instance of
 * this fragment using the provided parameters.
 *
 * @param posEquipo pos del equipo a mostrar
 * @return A new instance of fragment TeamDetailFragment.
 */
public static TeamDetailFragment newInstance(int posEquipo) {
    TeamDetailFragment fragment = new TeamDetailFragment();
    Bundle args = new Bundle();
    args.putInt(ARG_POS_EQUIPO, posEquipo);
    fragment.setArguments(args);

    return fragment;
}

@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    posEquipoMostrar = getArguments().getInt(ARG_POS_EQUIPO);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View view = inflater.inflate(R.layout.fragment_team_detail,
container, false);
    ivEscudo = (ImageView)
view.findViewById(R.id.ivEscudoDetailFragment);
    tvEquipo =
        (TextView)
view.findViewById(R.id.tvNombreEquipoDetailFragment);
    tvPuntos = (TextView)
view.findViewById(R.id.tvPuntosEquipoDetailFragment);
    return view;
}

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

    Equipo equipo =
MainActivity.teamArrayList.get(posEquipoMostrar);

    ivEscudo.setImageDrawable(equipo.getImagenEscudo());
    tvEquipo.setText(equipo.getNombreEquipo());
    tvPuntos.setText(String.valueOf(equipo.getPuntos()));
}
}

```

PASO 3: Modificar las Activities para que añadan los fragments usando **FragmentManager**, en vez de localizándolos con **findViewById**.

MAINACTIVITY.JAVA

```
public class MainActivity extends AppCompatActivity implements
TeamFragmentManager.OnListFragmentSelectionListener {

    static ArrayList<Equipo> teamArrayList = new ArrayList<Equipo>();

    //Gestión fragments
    private FragmentManager fragmentManager;
    private TeamDetailFragment teamDetailFragment; //fragment detalle

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

        // cargamos los valores en el arrayList
        rellenarArrayList();

        // Get a reference to the FragmentManager
        fragmentManager = getSupportFragmentManager();
        // Start a new FragmentTransaction
        FragmentTransaction fragmentTransaction = fragmentManager
            .beginTransaction();
        //añado el fragment listado
        fragmentTransaction.add(R.id.containerListFragment, new
TeamFragmentManager() );
        //hago commit
        fragmentTransaction.commit();

    }

    private void rellenarArrayList() {
        //recupero el array de los escudos de los recursos
        TypedArray arrayTypedEscudos =
getResources().obtainTypedArray(R.array.escudo_equipo);

        //recupero el array de los nombres de equipos
        String[] arrayNombreEquipos =
getResources().getStringArray(R.array.nombre_equipo);

        //recupero el array de los puntos de equipos
        int[] arrayPuntosEquipos =
getResources().getIntArray(R.array.puntos_equipo);

        //relleno el arrayList
        for (int i = 0; i < arrayNombreEquipos.length; i++) {
            teamArrayList.add(new Equipo(arrayNombreEquipos[i],
arrayTypedEscudos.getDrawable(i),
arrayPuntosEquipos[i]));
        }
    }
}
```

```

    }

    @Override
    public void onListFragmentSelection(int posEquipo) {
        //averiguo si hay dos paneles, mediante la búsqueda del
        segundo contenedor
        FrameLayout containerDetail = (FrameLayout)
        findViewById(R.id.containerDetail);

        if (containerDetail != null) {
            //hay dos paneles

            //si no se instanció el fragment detalle, lo instancio
            if (teamDetailFragment == null) {
                teamDetailFragment =
                TeamDetailFragment.newInstance(posEquipo);
            }
            //compruebo si ya fue añadido ese fragment a la activity
            if (!teamDetailFragment.isAdded()) {
                //lo añado
                FragmentTransaction fragmentTransaction =
                mFragmentManager.beginTransaction();
                fragmentTransaction.add(R.id.containerDetail,
                teamDetailFragment);
                fragmentTransaction.addToBackStack(null);
                fragmentTransaction.commit();
                //obligo a que ejecute el commit ahora
                mFragmentManager.executePendingTransactions();
            }
        } else {
            //hay un solo panel
            //debo lanzar la activity DetailTeamActivity
            Intent i = new Intent(this, DetailTeamActivity.class);
            i.putExtra(DetailTeamActivity.EXTRA_POS_EQUIPO,
            posEquipo);
            startActivity(i);
        }
    }
}

```

DETAILTEAMACTIVITY.JAVA

```

public class DetailTeamActivity extends AppCompatActivity {
    public static final String EXTRA_POS_EQUIPO =
    "com.android.alejandra.tutoria2fragestaticoappl.EXTRA_POS_EQUIPO";
    private FragmentManager mFragmentManager;

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

        //obtengo la pos del equipo a mostrar
        int pos = getIntent().getIntExtra(EXTRA_POS_EQUIPO, 0);

        // Get a reference to the FragmentManager
        mFragmentManager = getFragmentManager();
        // Start a new FragmentTransaction
    }
}

```

```
        FragmentTransaction fragmentTransaction = mFragmentManager
            .beginTransaction();
        //añado el fragment detalle pasándole la pos del equipo a
        mostrar
        fragmentTransaction.add(R.id.containerDetail,
            TeamDetailFragment.newInstance(pos));
        //hago commit
        fragmentTransaction.commit();
    }

}
```