

Manual Android

Patrones de Diseño

Daniel mendoza arredondo

A lo largo del siguiente documento se describe y se guía la manera de crear una aplicación Android en la cual se implementan 3 patrones de diseño Null Object, Singleton y Prototype

El primero Null Object se describe como encapsular la ausencia de un objeto mediante la provisión de una alternativa sustituible que ofrezca un valor por defecto con comportamiento "No hacer nada". En resumen, un diseño en donde "nada resultará de nada".

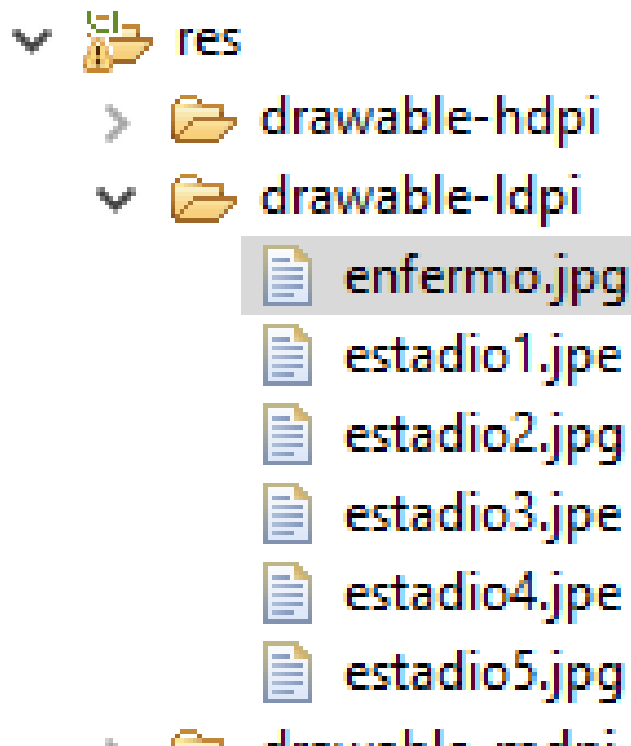
El segundo Singleton este patrón de diseño se encarga de que una clase determinada unicamente pueda tener un único objeto. Normalmente una clase puede instanciar todos los objetos que necesite.

Y por último el patrón prototype tiene como finalidad crear nuevos objetos duplicándolos, clonando una instancia creada previamente.

Este patrón especifica la clase de objetos a crear mediante la clonación de un prototipo que es una instancia ya creada. La clase de los objetos que servirán de prototipo deberá incluir en su interfaz la manera de solicitar una copia, que será desarrollada luego por las clases concretas de prototipos.

Para Comenzara vamos explicar un poco la aplicación, esta será para jugar un partido de ful-bol en la cual necesitamos una cancha sin ella no se puede jugar, un balón que solo se podrá crear una vez pero si podrá ser clonado.

Para empezar agregamos 5 imágenes de estadios y una que represente tristeza a nuestro proyecto.



A continuación se presenta la vista que tendrá nuestra aplicación siguiendo del código que lo genera.



El código para generarlo es el siguiente;

```
<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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical" >

            <ImageView
                android:id="@+id/imgEstado"
                android:layout_width="wrap_content"
                android:layout_height="260dp"
                android:src="@drawable/estadio1"
                android:paddingBottom="15dp"/>

            <Spinner
                android:id="@+id/spnEstadios"
                android:layout_width="match_parent"
                android:background="#a0a0a0"
                android:popupBackground="#d17474"
                android:layout_height="wrap_content" />

                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/btn_boton"
                    android:text="Crear Balon"
                    />

                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/btn_botonClonar"
                    android:text="Clonar Balon Para Repuesto"
                    />

            </LinearLayout>
        </ScrollView>

    </RelativeLayout>
```

Para controlar nuestra app el código en la actividad principal es el siguiente.

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;

import com.mendoza.arredondo.daniel.estadosdeanimo.NullObject;

public class MainActivity extends Activity {
    //Se declaran los controles
    private ImageView imgEstado;
    private Spinner spnEstadio;
    private Button botonCrear;
    private Button botonClonar;
    private Balon balon;
    private Balon balonClonado;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Se relacionan los controles
        imgEstado = (ImageView) findViewById(R.id.imgEstado);
        spnEstadio = (Spinner) findViewById(R.id.spnEstadios);
        botonCrear = (Button) findViewById(R.id.btn_boton);
        botonClonar = (Button) findViewById(R.id.btn_botonClonar);

        botonCrear.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                if(balon == null){
                    balon = Balon.getBalon();
                    Toast.makeText(getApplicationContext(), "Se ha creado un balón
de color: "+balon.getColor()+"\n"+
                                " Con la mediada: "+balon.getMedida(),
Toast.LENGTH_SHORT).show();
                }else{
                    Toast.makeText(MainActivity.this, "El balon ya se seleccionó" +
                                ", no puedes elejir otro. ",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

```

    botonClonar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            if(balon == null){
                Toast.makeText(getApplicationContext(), "No puedes Clonar sin
antes crear un Balon.", Toast.LENGTH_SHORT).show();
            }else{
                balonClonado = balon;
                Toast.makeText(getApplicationContext(), "Se ha clonado un balón
de color: "+balonClonado.getColor()+"\n"+
                    " Con la mediada: "+balonClonado.getMedida(),
Toast.LENGTH_SHORT).show();
            }
        }
    });

    //Se crea un arreglo con las operaciones de los spinner
    String operacionSpinner[] = new String[]{"Estadio 1", "Estadio 2", "Estadio
3", "Estadio 4", "No jugar"};
    //Se asigna los adaptadores para pasar el arreglo como opciones al spinner
    ArrayAdapter<String> adaptador = new ArrayAdapter<String>(getApplicationContext(),
        android.R.layout.select_dialog_singlechoice, operacionSpinner);
    spnEstadio.setAdapter(adaptador);
    //FIN DE LA INICIALIZACION DEL SPINNER
    //Se asigna un escuchador Al Spinner
    spnEstadio.setOnItemClickListener(new OnItemSelectedListener() {

        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1,
            int arg2, long arg3) {
            // TODO Auto-generated method stub
            if (spnEstadio.getSelectedItem().equals("Estadio 1")) {
                imgEstado.setImageResource(R.drawable.estadiol1);
            }else if (spnEstadio.getSelectedItem().equals("Estadio 2")) {
                imgEstado.setImageResource(R.drawable.estadio2);
            }else if (spnEstadio.getSelectedItem().equals("Estadio 3")) {
                imgEstado.setImageResource(R.drawable.estadio3);
            }else if (spnEstadio.getSelectedItem().equals("Estadio 4")) {
                imgEstado.setImageResource(R.drawable.estadio4);
            }else if (spnEstadio.getSelectedItem().equals("No jugar")) {
                //Plamamos una imagen en la que se refleje que no se
realiza nada

                imgEstado.setImageResource(R.drawable.enfermo);
                NullObject nula = new NullObject();
                nula.imagenNula();
                Toast.makeText(getApplicationContext(), "Null Object No se
jugará", Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
            // TODO Auto-generated method stub
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}

```

Para representar el patrón null object creamos una clase llamada NullObject con el siguiente código.

```
public class NullObject {  
  
    public void imagenNula()  
    {  
        //No se especifica nada  
        //No se hace nada  
    }  
}
```

Para representar el patrón Singleton necesitamos una clase con el nombre Balon y el siguiente código:

```
public class Balon {

    private static Balon balon;
    private String color;
    private int medida;

    public Balon() {
        // TODO Auto-generated constructor stub
        this.color = "Blanco";
        this.medida = 5;
    }

    public static Balon getBalon(){
        //Nos aseguramos que se crea un solo objeto de tipo Balon
        if(balon == null){
            balon = new Balon();
        }
        return balon;
    }

    public static void setBalon(Balon balon) {
        Balon.balon = balon;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public int getMedida() {
        return medida;
    }

    public void setMedida(int medida) {
        this.medida = medida;
    }

}
```


Para representar el patrón Prototype se necesitan dos clases la primera llamada PrototipoBalon y la segunda Clonable.

```
public class PrototipoBalon extends Balon implements Clonable{

    @Override
    public Object clonar() {
        // TODO Auto-generated method stub
        Balon bal = new Balon();

        bal.setColor(this.getColor());
        bal.setMedida(this.getMedida());
        return bal;
    }
}
```

```
public interface Clonable {

    public Object clonar();
}
```