



## Anexo C

# Contenido del CD-ROM

---

**T**odos los ficheros generados a lo largo del desarrollo de este proyecto han sido incluidos en el CD-ROM, en la carpeta “Información adicional”. A continuación, se muestra un pequeño índice con todos los recursos incluidos.



## CÓDIGOS DE PROGRAMACIÓN

### Android

#### Ficheros.java

ControlesFragment.java .....	3
DrawerItem.java .....	53
DrawerListAdapter.java .....	54
FileOperations.java .....	55
FragmentSplashActivity.java .....	57
MainActivity.java .....	58
SensoresFragment.java.....	64
Tutorial.java .....	77

#### Layouts

activity_main.xml .....	80
drawer_list_item.xml .....	81
fragment_ajustes.xml .....	82
fragment_controles.xml.....	89
fragment_sensores.xml.....	97
fragment_splash_activity.xml.....	101
header.xml.....	102
tutorial.xml.....	103

#### Recursos

main.xml .....	104
colors.xml.....	105
dimens.xml .....	108
strings.xml .....	109
AndroidManifest.xml.....	111

### Arduino

Bridge.ino.....	112
-----------------	-----



## ControlesFragment.java

```

package com.example.android.navigationdrawerexample;

import java.io.ByteArrayOutputStream;
import java.io.DataOutputStream;
import java.net.Socket;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.app.Fragment;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

public class ControlesFragment extends Fragment implements
SeekBar.OnSeekBarChangeListener, View.OnClickListener{

    /*****
    /*
    /* Declaracion de los atributos que se formarán en el fragmet*/
    /* y de las direcciones http que estableceran las conexiones */
    /*
    /*****

        TextView textlog;//Log for outputs
        public static final String TAG = "Inicio";

        /*
        * DIGITALES
        */
        private String web_servicel3i =
"http://192.168.240.1/arduino/mode/13/input/";
        private String web_servicel3o =
"http://192.168.240.1/arduino/mode/13/output/";
        private String web_servicel2i =
"http://192.168.240.1/arduino/mode/12/input/";
        private String web_servicel2o =
"http://192.168.240.1/arduino/mode/12/output/";
        private String web_servicel1i =
"http://192.168.240.1/arduino/mode/11/input/";
        private String web_servicel1o =
"http://192.168.240.1/arduino/mode/11/output/";
        private String web_servicel0i =
"http://192.168.240.1/arduino/mode/10/input/";
        private String web_servicel0o =
"http://192.168.240.1/arduino/mode/10/output/";
        private String web_service9i =
"http://192.168.240.1/arduino/mode/9/input/";

```



```
private String web_service9o =
"http://192.168.240.1/arduino/mode/9/output/";
private String web_service8i =
"http://192.168.240.1/arduino/mode/8/input/";
private String web_service8o =
"http://192.168.240.1/arduino/mode/8/output/";
private String web_service7i =
"http://192.168.240.1/arduino/mode/7/input/";
private String web_service7o =
"http://192.168.240.1/arduino/mode/7/output/";
private String web_service6i =
"http://192.168.240.1/arduino/mode/6/input/";
private String web_service6o =
"http://192.168.240.1/arduino/mode/6/output/";
private String web_service5i =
"http://192.168.240.1/arduino/mode/5/input/";
private String web_service5o =
"http://192.168.240.1/arduino/mode/5/output/";
private String web_service4i =
"http://192.168.240.1/arduino/mode/4/input/";
private String web_service4o =
"http://192.168.240.1/arduino/mode/4/output/";
private String web_service3i =
"http://192.168.240.1/arduino/mode/3/input/";
private String web_service3o =
"http://192.168.240.1/arduino/mode/3/output/";
private String web_service2i =
"http://192.168.240.1/arduino/mode/2/input/";
private String web_service2o =
"http://192.168.240.1/arduino/mode/2/output/";

private String web_serviced13 =
"http://192.168.240.1/arduino/digital/13";
private String web_serviced12 =
"http://192.168.240.1/arduino/digital/12";
private String web_serviced11 =
"http://192.168.240.1/arduino/digital/11";
private String web_serviced10 =
"http://192.168.240.1/arduino/digital/10";
private String web_serviced9 =
"http://192.168.240.1/arduino/digital/9";
private String web_serviced8 =
"http://192.168.240.1/arduino/digital/8";
private String web_serviced7 =
"http://192.168.240.1/arduino/digital/7";
private String web_serviced6 =
"http://192.168.240.1/arduino/digital/6";
private String web_serviced5 =
"http://192.168.240.1/arduino/digital/5";
private String web_serviced4 =
"http://192.168.240.1/arduino/digital/4";
private String web_serviced3 =
"http://192.168.240.1/arduino/digital/3";
private String web_serviced2 =
"http://192.168.240.1/arduino/digital/2";

/*
 * ANALOGICAS
 */

private String web_servicepwm11 =
"http://192.168.240.1/arduino/analog/11";
private String web_servicepwm10 =
"http://192.168.240.1/arduino/analog/10";
```



```
private String web_servicepwm9 =
"http://192.168.240.1/arduino/analog/9";
private String web_servicepwm6 =
"http://192.168.240.1/arduino/analog/6";
private String web_servicepwm5 =
"http://192.168.240.1/arduino/analog/5";
private String web_servicepwm3 =
"http://192.168.240.1/arduino/analog/3";

private String web_serviceact =
"http://192.168.240.1/arduino/actualizar";

Socket socket = null;//the socket for the connection
DataOutputStream dataOutputStream = null;//outputstream to send
commands
Boolean connected=false;//stores the connection status

private Switch btn13o, btn12o, btn11o, btn10o, btn9o, btn8o, btn7o,
btn6o;
private Switch btn5o, btn4o, btn3o, btn2o;
private CheckBox btn13i, btn12i, btn11i, btn10i, btn9i, btn8i, btn7i;
private CheckBox btn6i, btn5i, btn4i, btn3i, btn2i;
private SeekBar sb11, sb10, sb9, sb6, sb5, sb3;
private TextView txt11, txt10, txt9, txt6, txt5,txt3, txtprueba1,
txtprueba2,
txtprueba3, txtprueba4, txtprueba5, txtprueba6, txtprueba7,
txtprueba8,
txtprueba9, txtprueba10, txtprueball, txtprueba12;
private boolean i13, i12, i11, i10, i9, i8, i7, i6, i5, i4, i3, i2 =
false;
private int pwm11, pwm10, pwm9, pwm6, pwm5, pwm3;
private String ao11, ao10, ao9, ao6, ao5, ao3;
private Button act;

public String NC="N/C";
public String PWM="PWM";
public String DI="DI";
public String DO="DO";

String mCurrentPosition = "N/C";
public String p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13;

final String CONFIG_PIN = "config_pin";
final String CONFIG_PIN3 = "config_pin3";
final String CONFIG_PIN4 = "config_pin4";
final String CONFIG_PIN5 = "config_pin5";
final String CONFIG_PIN6 = "config_pin6";
final String CONFIG_PIN7 = "config_pin7";
final String CONFIG_PIN8 = "config_pin8";
final String CONFIG_PIN9 = "config_pin9";
final String CONFIG_PIN10 = "config_pin10";
final String CONFIG_PIN11 = "config_pin11";
final String CONFIG_PIN12 = "config_pin12";
final String CONFIG_PIN13 = "config_pin13";

public String pin2 = "Pin2";
public String pin3 = "Pin3";
public String pin4 = "Pin4";
public String pin5 = "Pin5";
public String pin6 = "Pin6";
public String pin7 = "Pin7";
public String pin8 = "Pin8";
public String pin9 = "Pin9";
public String pin10 = "Pin10";
public String pin11 = "Pin11";
```



```
public String pin12 = "Pin12";
public String pin13 = "Pin13";

/*****
/*
/* Constructor vacío necesario para que nuestra aplicación
/* no tenga fallos a la hora de recrear el Fragment.
/*
/*
/*****

    public ControlesFragment() {
    }

/*****
/*
/*      --OnCreateView--
/* El sistema lo llama cuando es el momento de que el
/* fragmento dibuje la IU por primera vez.
/* Para dibujar una IU para el fragmento, hay que devolver
/* un View desde este método que es la raíz del layout del
/* fragmento.
/* Se puede devolver null si el fragmento no suministra
/* una IU.
/*
/*
/*****
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.fragment_controles,
container, false);

/*****
/*
/* Declaración de las vistas que forman el layout
/*
/*
/*****
    btn13o = (Switch) v.findViewById(R.id.switch12);
    btn13i = (CheckBox) v.findViewById(R.id.CheckBox12);
    btn12o = (Switch) v.findViewById(R.id.switch11);
    btn12i = (CheckBox) v.findViewById(R.id.CheckBox11);
    btn11o = (Switch) v.findViewById(R.id.switch10);
    btn11i = (CheckBox) v.findViewById(R.id.CheckBox10);
    btn10o = (Switch) v.findViewById(R.id.switch9);
    btn10i = (CheckBox) v.findViewById(R.id.CheckBox9);
    btn9o = (Switch) v.findViewById(R.id.switch8);
    btn9i = (CheckBox) v.findViewById(R.id.CheckBox8);
    btn8o = (Switch) v.findViewById(R.id.switch7);
    btn8i = (CheckBox) v.findViewById(R.id.CheckBox7);
    btn7o = (Switch) v.findViewById(R.id.switch6);
    btn7i = (CheckBox) v.findViewById(R.id.CheckBox6);
    btn6o = (Switch) v.findViewById(R.id.switch5);
    btn6i = (CheckBox) v.findViewById(R.id.CheckBox5);
    btn5o = (Switch) v.findViewById(R.id.switch4);
    btn5i = (CheckBox) v.findViewById(R.id.CheckBox4);
    btn4o = (Switch) v.findViewById(R.id.switch3);
    btn4i = (CheckBox) v.findViewById(R.id.CheckBox3);
    btn3o = (Switch) v.findViewById(R.id.switch2);
    btn3i = (CheckBox) v.findViewById(R.id.CheckBox2);
    btn2o = (Switch) v.findViewById(R.id.switch1);

    //Inhabilitamos todos los botones por defecto.
```



```
btn2o.setEnabled(false);
btn3o.setEnabled(false);
btn4o.setEnabled(false);
btn5o.setEnabled(false);
btn6o.setEnabled(false);
btn7o.setEnabled(false);
btn8o.setEnabled(false);
btn9o.setEnabled(false);
btn10o.setEnabled(false);
btn11o.setEnabled(false);
btn12o.setEnabled(false);
btn13o.setEnabled(false);

btn2i = (CheckBox) v.findViewById(R.id.CheckBox1);
sb11 = (SeekBar) v.findViewById(R.id.seekBar6);
sb10 = (SeekBar) v.findViewById(R.id.seekBar5);
sb9 = (SeekBar) v.findViewById(R.id.seekBar4);
sb6 = (SeekBar) v.findViewById(R.id.seekBar3);
sb5 = (SeekBar) v.findViewById(R.id.seekBar2);
sb3 = (SeekBar) v.findViewById(R.id.seekBar1);
txt11 = (TextView) v.findViewById(R.id.textView6);
txt10 = (TextView) v.findViewById(R.id.textView5);
txt9 = (TextView) v.findViewById(R.id.textView4);
txt6 = (TextView) v.findViewById(R.id.textView3);
txt5 = (TextView) v.findViewById(R.id.textView2);
txt3 = (TextView) v.findViewById(R.id.textView1);

//Establecemos un valor maximo de 255 en el seekBar (PWM)
sb11.setMax(255);
sb10.setMax(255);
sb9.setMax(255);
sb6.setMax(255);
sb5.setMax(255);
sb3.setMax(255);

//Inhabilitamos todos los botones por defecto.
sb3.setEnabled(false);
sb5.setEnabled(false);
sb6.setEnabled(false);
sb9.setEnabled(false);
sb10.setEnabled(false);
sb11.setEnabled(false);

//Invocacion de llamadas cuando el usuario pulsa un boton.
//Tanto para la seeksbar como para el switch
sb11.setOnSeekBarChangeListener(this);
sb10.setOnSeekBarChangeListener(this);
sb9.setOnSeekBarChangeListener(this);
sb6.setOnSeekBarChangeListener(this);
sb5.setOnSeekBarChangeListener(this);
sb3.setOnSeekBarChangeListener(this);
btn13o.setOnClickListener(this);
btn12o.setOnClickListener(this);
btn11o.setOnClickListener(this);
btn10o.setOnClickListener(this);
btn9o.setOnClickListener(this);
btn8o.setOnClickListener(this);
btn7o.setOnClickListener(this);
btn6o.setOnClickListener(this);
btn5o.setOnClickListener(this);
btn4o.setOnClickListener(this);
btn3o.setOnClickListener(this);
btn2o.setOnClickListener(this);

act = (Button) v.findViewById(R.id.button1);
```



```

txtprueba1 = (TextView) v.findViewById(R.id.textViewPrueba1);
txtprueba2 = (TextView) v.findViewById(R.id.textViewPrueba2);
txtprueba3 = (TextView) v.findViewById(R.id.textViewPrueba3);
txtprueba4 = (TextView) v.findViewById(R.id.textViewPrueba4);
txtprueba5 = (TextView) v.findViewById(R.id.textViewPrueba5);
txtprueba6 = (TextView) v.findViewById(R.id.textViewPrueba6);
txtprueba7 = (TextView) v.findViewById(R.id.textViewPrueba7);
txtprueba8 = (TextView) v.findViewById(R.id.textViewPrueba8);
txtprueba9 = (TextView) v.findViewById(R.id.textViewPrueba9);
txtprueba10 = (TextView) v.findViewById(R.id.textViewPrueba10);
txtprueball = (TextView) v.findViewById(R.id.textViewPrueball);
txtprueball2 = (TextView) v.findViewById(R.id.textViewPrueball2);

return v;
}

/*****
/*
/* Metodo de llamada cuando el fragmento esta siendo visible */
/* ante el usuario.
/*
/*
*****/
@Override
public void onStart() {
    super.onStart();

/*****
/*
/* Carga los datos guardados en la memoria interna
/*
/*
*****/

//Crea una nueva clase para realizar la invocacion posterior.
FileOperations fop = new FileOperations();
//Envio de parametros a la nueva clase
String carga2 = fop.read(pin2);

//Condicion necesaria para comprobar si existen los ficheron que
queremos leer.
if(carga2 != null){
    //Si el fichero existe, leemos de la memoria interna y le
asociamos ese valor
//al texview que se creo a traves del layout
txtprueba1.setText(carga2);
//Asignamos ese valor a una variable
p2=txtprueba1.getText().toString();

//CompareTo compara los String recibidos, devolviendo el
valor cero si son iguales,
// -1 s es menor o 1 si es mayor.
if(p2.compareTo(NC) != 0 && p2.compareTo(DI) != 0){
    btn2o.setEnabled(true);
}
}
else {
    Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
}

String carga3 = fop.read(pin3);

```





```
        if(carga3 != null) {
            txtprueba2.setText(carga3);
            p3=txtprueba2.getText().toString();

            if(p3.compareTo(NC) != 0 && p3.compareTo(DO) == 0 &&
p3.compareTo(DI) != 0){
                btn3o.setEnabled(true);
            }
            if(p3.compareTo(PWM) == 0){
                sb3.setEnabled(true);
            }
        }
        else {
            Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
        }

String carga4 = fop.read(pin4);

if(carga4 != null) {
    txtprueba3.setText(carga4);
    p4=txtprueba3.getText().toString();

    if(p4.compareTo(NC) != 0 && p4.compareTo(DI) != 0){
        btn4o.setEnabled(true);
    }
}
else {
    Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
}

String carga5 = fop.read(pin5);

if(carga5 != null) {
    txtprueba4.setText(carga5);
    p5=txtprueba4.getText().toString();

    if(p5.compareTo(NC) != 0 && p5.compareTo(DO) == 0 &&
p5.compareTo(DI) != 0){
        btn5o.setEnabled(true);
    }
    if(p5.compareTo(PWM) == 0){
        sb5.setEnabled(true);
    }
}
else {
    Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
}

String carga6 = fop.read(pin6);

if(carga6 != null) {
    txtprueba5.setText(carga6);
    p6=txtprueba5.getText().toString();

    if(p6.compareTo(NC) != 0 && p6.compareTo(DO) == 0 &&
p6.compareTo(DI) != 0){
        btn6o.setEnabled(true);
    }
}
```



```
        if(p6.compareTo(PWM) == 0){
            sb6.setEnabled(true);
        }
    }
    else {
        Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
    }

String carga7 = fop.read(pin7);

if(carga7 != null){
    txtprueba6.setText(carga7);
    p7=txtprueba6.getText().toString();

    if(p7.compareTo(NC) != 0 && p7.compareTo(DI) != 0){
        btn7o.setEnabled(true);
    }
}
else {
    Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
}

String carga8 = fop.read(pin8);

if(carga8 != null){
    txtprueba7.setText(carga8);
    p8=txtprueba7.getText().toString();

    if(p8.compareTo(NC) != 0 && p8.compareTo(DI) != 0){
        btn8o.setEnabled(true);
    }
}
else {
    Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
}

String carga9 = fop.read(pin9);

if(carga9 != null){
    txtprueba8.setText(carga9);
    p9=txtprueba8.getText().toString();

    if(p9.compareTo(NC) != 0 && p9.compareTo(DO) == 0 &&
p9.compareTo(DI) != 0){
        btn9o.setEnabled(true);
    }
    if(p9.compareTo(PWM) == 0){
        sb9.setEnabled(true);
    }
}
else {
    Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
}

String carga10 = fop.read(pin10);

if(carga10 != null){
```



```
txtprueba9.setText(carga10);
p10=txtprueba9.getText().toString();

    if(p10.compareTo(NC) != 0 && p10.compareTo(DO) == 0 &&
p10.compareTo(DI) != 0){
    btn10o.setEnabled(true);
    }
    if(p10.compareTo(PWM) == 0){
    sb10.setEnabled(true);
    }
    }
    else {
        Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
    }

String carga11 = fop.read(pin11);

    if(carga11 != null){
        txtprueba10.setText(carga11);
        p11=txtprueba10.getText().toString();

        if(p11.compareTo(NC) != 0 && p11.compareTo(DO) == 0 &&
p11.compareTo(DI) != 0){
        btn11o.setEnabled(true);
        }
        if(p11.compareTo(PWM) == 0){
        sb11.setEnabled(true);
        }
        }
        else {
            Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
        }

String carga12 = fop.read(pin12);

    if(carga12 != null){
        txtprueba11.setText(carga12);
        p12=txtprueba11.getText().toString();

        if(p12.compareTo(NC) != 0 && p12.compareTo(DI) != 0){
        btn12o.setEnabled(true);
        }
        }
        else {
            Toast.makeText(getActivity(), "File not Found",
Toast.LENGTH_SHORT).show();
        }

String carga13 = fop.read(pin13);

    if(carga13 != null){
        txtprueba12.setText(carga13);
        p13=txtprueba12.getText().toString();

        if(p13.compareTo(NC) != 0 && p13.compareTo(DI) != 0){
        btn13o.setEnabled(true);
        }
        }
        else {
```



```

        Toast.makeText(getActivity(), "File not Found",
        Toast.LENGTH_SHORT).show();
    }

    /**
     * Lectura de los datos recibidos a traves de la interfaz
     * enviados por el otro fragment.
     */
    /**
     *
     */
    /**
     *
     */

    //getArguments devuelve los argumentos asociados cuando el
    fragment fue instanciado.
    Bundle args = getArguments();

    //Si no hay argumentos no hace nada.
    if (args != null) {
        //Una vez recibidos los argumentos, devuelve el valor
        asociado con un
        //nombre clave y lo asocia al textview correspondiente con
        setText

        txtprueba1.setText(args.getString(CONFIG_PIN));
        p2=txtprueba1.getText().toString();

        //Si el argumento recibido es DO (Digital Output)
        establecemos el boton
        //para que puede ser manejado por el usuario
        if(p2.compareTo(NC) != 0 && p2.compareTo(DI) != 0){
            btn2o.setEnabled(true);
        }
    }

    if (args != null) {
        txtprueba2.setText(args.getString(CONFIG_PIN3));
        p3=txtprueba2.getText().toString();

        if(p3.compareTo(NC) != 0 && p3.compareTo(DO) == 0 &&
        p3.compareTo(DI) != 0){
            btn3o.setEnabled(true);
        }
        if (p3.compareTo(PWM) == 0){
            sb3.setEnabled(true);
        }
    }

    if (args != null) {
        txtprueba3.setText(args.getString(CONFIG_PIN4));
        p4=txtprueba3.getText().toString();

        if(p4.compareTo(NC) != 0 && p4.compareTo(DI) != 0){
            btn4o.setEnabled(true);
        }
    }

    if (args != null) {
        txtprueba4.setText(args.getString(CONFIG_PIN5));
        p5=txtprueba4.getText().toString();

        if(p5.compareTo(NC) != 0 && p5.compareTo(DO) == 0 &&
        p5.compareTo(DI) != 0){
            btn5o.setEnabled(true);
        }
        if (p5.compareTo(PWM) == 0){
            sb5.setEnabled(true);
        }
    }

```



```
    }  
  }  
  
  if (args != null) {  
    txtprueba5.setText(args.getString(CONFIG_PIN6));  
    p6=txtprueba5.getText().toString();  
  
    if(p6.compareTo(NC) != 0 && p6.compareTo(DO) == 0 &&  
p6.compareTo(DI) != 0){  
      btn6o.setEnabled(true);  
    }  
    if(p6.compareTo(PWM) == 0){  
      sb6.setEnabled(true);  
    }  
  }  
  
  if (args != null) {  
    txtprueba6.setText(args.getString(CONFIG_PIN7));  
    p7=txtprueba6.getText().toString();  
  
    if(p7.compareTo(NC) != 0 && p7.compareTo(DI) != 0){  
      btn7o.setEnabled(true);  
    }  
  }  
  
  if (args != null) {  
    txtprueba7.setText(args.getString(CONFIG_PIN8));  
    p8=txtprueba7.getText().toString();  
  
    if(p8.compareTo(NC) != 0 && p8.compareTo(DI) != 0){  
      btn8o.setEnabled(true);  
    }  
  }  
  
  if (args != null) {  
    txtprueba8.setText(args.getString(CONFIG_PIN9));  
    p9=txtprueba8.getText().toString();  
  
    if(p9.compareTo(NC) != 0 && p9.compareTo(DO) == 0 &&  
p9.compareTo(DI) != 0){  
      btn9o.setEnabled(true);  
    }  
    if(p9.compareTo(PWM) == 0){  
      sb9.setEnabled(true);  
    }  
  }  
  
  if (args != null) {  
    txtprueba9.setText(args.getString(CONFIG_PIN10));  
    p10=txtprueba9.getText().toString();  
  
    if(p10.compareTo(NC) != 0 && p10.compareTo(DO) == 0 &&  
p10.compareTo(DI) != 0){  
      btn10o.setEnabled(true);  
    }  
    if(p10.compareTo(PWM) == 0){  
      sb10.setEnabled(true);  
    }  
  }  
  
  if (args != null) {  
    txtprueba10.setText(args.getString(CONFIG_PIN11));  
    p11=txtprueba10.getText().toString();  
  }  
}
```



```

        if(p11.compareTo(NC) != 0 && p11.compareTo(DO) == 0 &&
p11.compareTo(DI) != 0){
            btn11o.setEnabled(true);
        }
        if(p11.compareTo(PWM) == 0){
            sb11.setEnabled(true);
        }
    }

    if (args != null) {
        txtprueball1.setText(args.getString(CONFIG_PIN12));
        p12=txtprueball1.getText().toString();

        if(p12.compareTo(NC) != 0 && p12.compareTo(DI) != 0){
            btn12o.setEnabled(true);
        }
    }

    if (args != null) {
        txtpruebal2.setText(args.getString(CONFIG_PIN13));
        p13=txtpruebal2.getText().toString();

        if(p13.compareTo(NC) != 0 && p13.compareTo(DI) != 0){
            btn13o.setEnabled(true);
        }
    }

/*****
/*
/* Actualizacion de datos.
/* Envio de datos de Arduino a Android.
/*
/*
/*****

//Realizamos la lectura de datos de Android en el boton switch
si esta marcado como "DO"
    if (p13.compareTo(DO) == 0 ){

        //Creacion de una tarea asincrona. Necesaria para evitar que
una tarea bloquee
        //el hilo principal de la interfaz de usuario.
        new AsyncTask<Void, Void, Void>() {

            //Llamada despues del OnPreExecute(En este caso no
era necesario implementarlo)
            //Es aqui donde se realiza la tarea principal
            @Override
            protected Void doInBackground(Void... params) {

                //HttpClient crea una interfaz para un cliente HTTP.
                //El cliente encapsula los objetos necesarios para
ejecutar las peticiones HTTP
                //durante la autentificacion, gestion de la conexion y
otras características
                HttpClient httpclient = new DefaultHttpClient();

                //try/catch es un bloque que gestiona los fallos que
se pueden ocasionar
                //Dentro del bloque try colocamos las funciones que
podrian porvocar fallo
                //si esto ocurre, saltamos al bloque catch que
gestionara el error.

```



```

        try {
            //Recibimos la respuesta HTTP y ejecutamos
nuestra petición.
            //Enviamos la URL que interpretará Arduino
            HttpResponse respons = httpClient.execute(new
HttpGet(web_servicel3o));
            HttpResponse response = httpClient.execute(new
HttpGet(web_serviced13));

            //Esta clase implementa un flujo de salida en
el que los datos se
            //escriben en una matriz de bits (32bits por
defecto). El buffer crece automáticamente como se
            //escriben datos en ella.
            ByteArrayOutputStream out = new
ByteArrayOutputStream();
            //Escribe los datos obtenidos en el flujo y
escribe en la cadena enviada.
            response.getEntity().writeTo(out);
            //cierra el flujo de datos
            out.close();

            //Recibimos un respuesta de Arduino de la
lectura del pin
            //Para pines digitales 0 ó 1
            String respuesta = out.toString();

            //Si recibimos un 0, ponemos el estado del
switch a OFF
            if (respuesta.endsWith("\r\n")) {
                i13 = false;
            } else {
                i13 = true;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    //Una vez terminado el metodo anterior, llamamos a
onPostExecute
    //donde actualizara el estado del boton dependiendo
del valor recibido
    //en la cadena url.
    @Override
    protected void onPostExecute(Void aVoid) {
        //funcion que actualiza el estado del boton switch
        actualizaEstadoBoton13();
    }
    }.execute();
}

if (p12.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

            try {

```



```

        HttpResponse respons = httpClient.execute(new
HttpGet(web_servicel2o));
        HttpResponse response = httpClient.execute(new
HttpGet(web_servicedl2));
        ByteArrayOutputStream out = new
ByteArrayOutputStream();
        response.getEntity().writeTo(out);
        out.close();

        String respuesta = out.toString();

        if (respuesta.endsWith("\r\n")) {
            i12 = false;
        } else {
            i12 = true;
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void aVoid) {
    actualizaEstadoBoton12();
}
}.execute();
}

if (p11.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpClient.execute(new
HttpGet(web_servicel1o));
                HttpResponse response = httpClient.execute(new
HttpGet(web_servicedl1));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i11 = false;
                } else {
                    i11 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override

```





```
        protected void onPostExecute(Void aVoid) {
            actualizaEstadoBoton11();
        }
    }.execute();
}

if (p10.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {
        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpclient.execute(new
HttpGet(web_service10));
                HttpResponse response = httpclient.execute(new
HttpGet(web_serviced10));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i10 = false;
                } else {
                    i10 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            actualizaEstadoBoton10();
        }
    }.execute();
}

if (p9.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {
        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpclient.execute(new
HttpGet(web_service9));
                HttpResponse response = httpclient.execute(new
HttpGet(web_serviced9));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
```



```

        out.close();

        String respuesta = out.toString();

        if (respuesta.endsWith("\r\n")) {
            i9 = false;
        } else {
            i9 = true;
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void aVoid) {
    actualizaEstadoBoton9();
}
}.execute();
}

if (p8.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpclient.execute(new
HttpGet(web_service80));
                HttpResponse response = httpclient.execute(new
HttpGet(web_serviced8));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i8 = false;
                } else {
                    i8 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            actualizaEstadoBoton8();
        }
    }.execute();
}
}

```



```
if (p7.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpclient.execute(new
HttpGet(web_service7o));
                HttpResponse response = httpclient.execute(new
HttpGet(web_serviced7));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i7 = false;
                } else {
                    i7 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {

            actualizaEstadoBoton7();
        }
    }.execute();
}

if (p6.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpclient.execute(new
HttpGet(web_service6o));
                HttpResponse response = httpclient.execute(new
HttpGet(web_serviced6));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i6 = false;
                } else {
                    i6 = true;
                }

            }


```



```

    }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void aVoid) {
    actualizaEstadoBoton6();
}
}.execute();
}

if (p5.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {
        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpClient.execute(new
HttpGet(web_service5o));
                HttpResponse response = httpClient.execute(new
HttpGet(web_serviced5));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i5 = false;
                } else {
                    i5 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            actualizaEstadoBoton5();
        }
    }.execute();
}

if (p4.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {
        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

```



```

        try {
            HttpResponse respons = httpClient.execute(new
HttpGet(web_service4o));
            HttpResponse response = httpClient.execute(new
HttpGet(web_serviced4));
            ByteArrayOutputStream out = new
ByteArrayOutputStream();
            response.getEntity().writeTo(out);
            out.close();

            String respuesta = out.toString();

            if (respuesta.endsWith("\r\n")) {
                i4 = false;
            } else {
                i4 = true;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {

        actualizaEstadoBoton4();
    }
    }.execute();
}

if (p3.compareTo(D0) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpClient.execute(new
HttpGet(web_service3o));
                HttpResponse response = httpClient.execute(new
HttpGet(web_serviced3));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i3 = false;
                } else {
                    i3 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }
}

```



```
@Override
protected void onPostExecute(Void aVoid) {

    actualizaEstadoBoton3();
}
}.execute();
}

if (p2.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

            try {
                HttpResponse respons = httpClient.execute(new
HttpGet(web_service2o));
                HttpResponse response = httpClient.execute(new
HttpGet(web_serviced2));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();

                if (respuesta.endsWith("\r\n")) {
                    i2 = false;
                } else {
                    i2 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {

            actualizaEstadoBoton2();
        }
    }.execute();
}

if (p11.compareTo(PWM) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new DefaultHttpClient();

            try {
                HttpResponse response = httpClient.execute(new
HttpGet(web_servicepwm1));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
```



```

        response.getEntity().writeTo(out);
        out.close();
        String respuesta = out.toString();

        //parseInt especifica un String asignado como
un entero ASCII
        //con el valor del pin analogico.
        //Realizamos la lectura de los enteros
enviados por Arduino.
        pwm11=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
        sb11.setProgress(pwm11);

    } catch (Exception e) {
        e.printStackTrace();
    }

    return null;
}

@Override
//Actualizamos el estado del seekbar
protected void onPostExecute(Void aVoid) {

    actualizapwm11();
}

}.execute();
}

if (p10.compareTo(PWM) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse response = httpclient.execute(new
HttpGet(web_servicepwm10));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();
                String respuesta = out.toString();
                pwm10=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
                sb10.setProgress(pwm10);

            } catch (Exception e) {
                e.printStackTrace();
            }

            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {

            actualizapwm10();
        }

    }.execute();
}

```



```
    }

    if (p9.compareTo(PWM) == 0 ){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new DefaultHttpClient();

                try {
                    HttpResponse response = httpClient.execute(new
HttpGet(web_servicepwm9));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();
                    response.getEntity().writeTo(out);
                    out.close();
                    String respuesta = out.toString();
                    pwm9=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
                    sb9.setProgress(pwm9);

                } catch (Exception e) {
                    e.printStackTrace();
                }

                return null;
            }

            @Override
            protected void onPostExecute(Void aVoid) {

                actualizapwm9();
            }

        }.execute();
    }

    if (p6.compareTo(PWM) == 0 ){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new DefaultHttpClient();

                try {
                    HttpResponse response = httpClient.execute(new
HttpGet(web_servicepwm6));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();
                    response.getEntity().writeTo(out);
                    out.close();
                    String respuesta = out.toString();
                    pwm6=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
                    sb6.setProgress(pwm6);

                } catch (Exception e) {
                    e.printStackTrace();
                }

                return null;
            }

        }.execute();
    }
}
```





```
    }

    @Override
    protected void onPostExecute(Void aVoid) {

        actualizapwm6();
    }

    }.execute();
}

if (p5.compareTo(PWM) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse response = httpclient.execute(new
HttpGet(web_servicepwm5));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();
                String respuesta = out.toString();
                pwm5=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
                sb5.setProgress(pwm5);

            } catch (Exception e) {
                e.printStackTrace();
            }

            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {

            actualizapwm5();
        }

    }.execute();
}

if (p3.compareTo(PWM) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpclient = new DefaultHttpClient();

            try {
                HttpResponse response = httpclient.execute(new
HttpGet(web_servicepwm3));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();
                String respuesta = out.toString();
```



```

        pwm3=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
        sb3.setProgress(pwm3);

        } catch (Exception e) {
            e.printStackTrace();
        }

        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {

        actualizapwm3();
    }

}.execute();
}

/*****
/*
/* Metodo de invocacion al pulsar el boton "ACTUALIZAR"
/* Actualiza los botones de acuerdo a los datos recibidos
/* por Arduino
/*
/*
/*****

Button act = (Button) getActivity().findViewById(R.id.button1);
act.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {

        Toast.makeText(getActivity(), "Actualizando...",
Toast.LENGTH_LONG).show();

/*****
/*
/* Actualizacion de datos.
/* Envio de datos de Arduino a Android.
/*
/*
/*****

        if (p13.compareTo(DO) == 0 ){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void

doInBackground(Void... params) {

                    HttpClient httpClient = new
DefaultHttpClient();

                    try {
                        HttpResponse respons =
httpClient.execute(new HttpGet(web_service13o));
                        HttpResponse response =
httpClient.execute(new HttpGet(web_serviced13));

```



```

        ByteArrayOutputStream out = new
        ByteArrayOutputStream();

        response.getEntity().writeTo(out);

        out.close();

        String respuesta =

        out.toString();

        if

        (respuesta.endsWith("\r\n")) {

            i13 = false;
        } else {
            i13 = true;
        }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void

aVoid) {

        actualizaEstadoBoton13();
    }
    }.execute();
}

if (p12.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {

            HttpClient httpClient = new

DefaultHttpClient();

            try {
                HttpResponse respons =
                httpClient.execute(new HttpGet(web_servicel2o));
                HttpResponse response =
                httpClient.execute(new HttpGet(web_serviced12));
                ByteArrayOutputStream out =
                new ByteArrayOutputStream();
                response.getEntity().writeTo(out);

                out.close();

                String respuesta =

                if

                (respuesta.endsWith("\r\n")) {

                    i12 = false;
                } else {
                    i12 = true;
                }

                } catch (Exception e) {
                    e.printStackTrace();

```



```

    }
    return null;
}

@Override
protected void onPostExecute(Void

aVoid) {

    actualizaEstadoBoton12();
}
}.execute();
}

if (p11.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();

            try {
                HttpResponse respons =
httpClient.execute(new HttpGet(web_servicello));
                HttpResponse response =
httpClient.execute(new HttpGet(web_serviced11));
                ByteArrayOutputStream out =
new ByteArrayOutputStream();
                response.getEntity().writeTo(out);

                out.close();

                String respuesta =

                if

                    i11 = false;
                } else {
                    i11 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void

aVoid) {

            actualizaEstadoBoton11();
        }
    }.execute();
}

if (p10.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

```



```

        @Override
        protected Void
doInBackground(Void... params) {
    DefaultHttpClient();

        HttpClient httpClient = new
        try {
            HttpResponse respons =
httpClient.execute(new HttpGet(web_service10));
            HttpResponse response =
httpClient.execute(new HttpGet(web_serviced10));
            ByteArrayOutputStream out =
new ByteArrayOutputStream();
            response.getEntity().writeTo(out);

            out.close();

            String respuesta =

            if

                i10 = false;
            } else {
                i10 = true;
            }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void
aVoid) {

            actualizaEstadoBoton10();
        }
    }.execute();
}

if (p9.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void
doInBackground(Void... params) {
    DefaultHttpClient();

        HttpClient httpClient = new

        try {
            HttpResponse respons =
httpClient.execute(new HttpGet(web_service9));
            HttpResponse response =
httpClient.execute(new HttpGet(web_serviced9));
            ByteArrayOutputStream out =
new ByteArrayOutputStream();
            response.getEntity().writeTo(out);

            out.close();

```



```

        out.toString();

        (respuesta.endsWith("\r\n")) {

            String respuesta =

                if
                    i9 = false;
                } else {
                    i9 = true;
                }

                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }

            @Override
            protected void onPostExecute(Void

aVoid) {

                actualizaEstadoBoton9();
            }
        }.execute();
    }

    if (p8.compareTo(DO) == 0 ){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void

doInBackground(Void... params) {

                HttpClient httpClient = new
                DefaultHttpClient();

                try {
                    HttpResponse respons =
                    httpClient.execute(new HttpGet(web_service8o));
                    HttpResponse response =
                    httpClient.execute(new HttpGet(web_serviced8));
                    ByteArrayOutputStream out =
                    new ByteArrayOutputStream();
                    response.getEntity().writeTo(out);

                    out.close();

                    String respuesta =

                    if
                        i8 = false;
                    } else {
                        i8 = true;
                    }

                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    return null;
                }

                @Override
                protected void onPostExecute(Void

aVoid) {

```



```

        actualizaEstadoBoton8();
    }
    }.execute();
}

if (p7.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {
        HttpClient httpClient = new
DefaultHttpClient();

        try {
            HttpResponse respons =
httpClient.execute(new HttpGet(web_service7o));
            HttpResponse response =
httpClient.execute(new HttpGet(web_serviced7));
            new ByteArrayOutputStream();
            response.getEntity().writeTo(out);

            out.close();

            String respuesta =

            if

                i7 = false;
            } else {
                i7 = true;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void

aVoid) {

        actualizaEstadoBoton7();
    }
    }.execute();
}

if (p6.compareTo(DO) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {
        HttpClient httpClient = new
DefaultHttpClient();

        try {

```



```

        HttpResponse respons =
httpClient.execute(new HttpGet(web_service6));
        HttpResponse response =
httpClient.execute(new HttpGet(web_serviced6));
        ByteArrayOutputStream out =
new ByteArrayOutputStream();
        response.getEntity().writeTo(out);

        out.close();

        String respuesta =

        if

            i6 = false;
        } else {
            i6 = true;
        }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void

aVoid) {

        actualizaEstadoBoton6();
    }
    }.execute();
}

if (p5.compareTo(D0) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {

            HttpClient httpClient = new
DefaultHttpClient();

            try {
                HttpResponse respons =
httpClient.execute(new HttpGet(web_service5));
                HttpResponse response =
httpClient.execute(new HttpGet(web_serviced5));
                ByteArrayOutputStream out =
new ByteArrayOutputStream();
                response.getEntity().writeTo(out);

                out.close();

                String respuesta =

                if

                    i5 = false;
                } else {
                    i5 = true;
                }
            }
        }
    }
}

```





```

    }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void
aVoid) {

    actualizaEstadoBoton5();
    }.execute();
}

if (p4.compareTo(DO) == 0 ){
    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void
doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();

            try {
                HttpResponse respons =
httpClient.execute(new HttpGet(web_service4o));
                HttpResponse response =
httpClient.execute(new HttpGet(web_serviced4));
                ByteArrayOutputStream out =
new ByteArrayOutputStream();
                response.getEntity().writeTo(out);

                out.close();

                String respuesta =

                if

                    i4 = false;
                } else {
                    i4 = true;
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void
aVoid) {

            actualizaEstadoBoton4();
            }.execute();
        }
}

```



```

        if (p3.compareTo(DO) == 0 ){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void
doInBackground(Void... params) {
                    HttpClient httpClient = new
DefaultHttpClient();

                    try {
                        HttpResponse respons =
httpClient.execute(new HttpGet(web_service3o));
                        HttpResponse response =
httpClient.execute(new HttpGet(web_serviced3));
                        new ByteArrayOutputStream();
                        response.getEntity().writeTo(out);

                        out.close();

                        String respuesta =

                            if
                                i3 = false;
                            } else {
                                i3 = true;
                            }

                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                        return null;
                    }

                @Override
                protected void onPostExecute(Void
aVoid) {

                    actualizaEstadoBoton3();
                }
            }.execute();
        }

        if (p2.compareTo(DO) == 0 ){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void
doInBackground(Void... params) {
                    HttpClient httpClient = new
DefaultHttpClient();

                    try {
                        HttpResponse respons =
httpClient.execute(new HttpGet(web_service2o));
                        HttpResponse response =
httpClient.execute(new HttpGet(web_serviced2));
                        new ByteArrayOutputStream();
                    }
                }
            }.execute();
        }
    }
}

```



```

response.getEntity().writeTo(out);

out.toString();

(respuesta.endsWith("0\r\n")) {

aVoid) {

        out.close();

        String respuesta =

                if

                    i2 = false;
                } else {
                    i2 = true;
                }

                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }

            @Override
            protected void onPostExecute(Void

                actualizaEstadoBoton2();
            }.execute();
        }

        if (p11.compareTo(PWM) == 0 ){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void

                    HttpClient httpClient = new

                        try {
                            HttpResponse response =
                                HttpClient.execute(new HttpGet(web_servicepwm1));
                                ByteArrayOutputStream out =
                                new ByteArrayOutputStream();
                                response.getEntity().writeTo(out);

                                    out.close();
                                    String respuesta =

                                        pwm1=
                                        sb11.setProgress(pwm1);

                                    } catch (Exception e) {
                                        e.printStackTrace();
                                    }

                                    return null;
                                }

                                @Override
                                protected void onPostExecute(Void

aVoid) {

```



```

        actualizapwm1();
    }

    }.execute();
}

if (p10.compareTo(PWM) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {

            HttpClient httpClient = new
            DefaultHttpClient();

                try {
                    HttpResponse response =
                    httpClient.execute(new HttpGet(web_servicepwm10));
                    ByteArrayOutputStream out =
                    new ByteArrayOutputStream();
                    response.getEntity().writeTo(out);

                    out.close();
                    String respuesta =

                    pwm10=
                    Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
                    sb10.setProgress(pwm10);

                } catch (Exception e) {
                    e.printStackTrace();
                }

                return null;
            }

            @Override
            protected void onPostExecute(Void

aVoid) {

                actualizapwm10();
            }

        }.execute();
}

if (p9.compareTo(PWM) == 0 ){

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void

doInBackground(Void... params) {

            HttpClient httpClient = new
            DefaultHttpClient();

                try {
                    HttpResponse response =
                    httpClient.execute(new HttpGet(web_servicepwm9));
                    ByteArrayOutputStream out =
                    new ByteArrayOutputStream();

```



```

response.getEntity().writeTo(out);
out.close();
String respuesta =
out.toString();
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
pwm9=
sb9.setProgress(pwm9);
} catch (Exception e) {
e.printStackTrace();
}
return null;
}

@Override
protected void onPostExecute(Void
aVoid) {
actualizapwm9();
}.execute();
}

if (p6.compareTo(PWM) == 0 ){
new AsyncTask<Void, Void, Void>() {
@Override
protected Void
doInBackground(Void... params) {
HttpClient httpClient = new
DefaultHttpClient();
try {
HttpResponse response =
httpClient.execute(new HttpGet(web_servicepwm6));
ByteArrayOutputStream out =
new ByteArrayOutputStream();
response.getEntity().writeTo(out);
out.close();
String respuesta =
out.toString();
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
pwm6=
sb6.setProgress(pwm6);
} catch (Exception e) {
e.printStackTrace();
}
return null;
}
@Override
protected void onPostExecute(Void
aVoid) {
actualizapwm6();
}
}
}

```



```

        }.execute();
    }

    if (p5.compareTo(PWM) == 0 ){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void

doInBackground(Void... params) {
                DefaultHttpClient();

                HttpClient httpClient = new

                    try {
                        HttpResponse response =
httpClient.execute(new HttpGet(web_servicepwm5));
                        ByteArrayOutputStream out =
new ByteArrayOutputStream();
                        response.getEntity().writeTo(out);

                            out.close();
                            String respuesta =

                                pwm5=
                                sb5.setProgress(pwm5);

                            } catch (Exception e) {
                                e.printStackTrace();
                            }

                        return null;
                    }

            @Override
            protected void onPostExecute(Void

aVoid) {

                actualizapwm5();
            }

        }.execute();
    }

    if (p3.compareTo(PWM) == 0 ){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void

doInBackground(Void... params) {
                DefaultHttpClient();

                HttpClient httpClient = new

                    try {
                        HttpResponse response =
httpClient.execute(new HttpGet(web_servicepwm3));
                        ByteArrayOutputStream out =
new ByteArrayOutputStream();
                        response.getEntity().writeTo(out);

                            out.close();

```



```

        String respuesta =
out.toString();
        Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
        pwm3=
        sb3.setProgress(pwm3);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}
@Override
protected void onPostExecute(Void
aVoid) {
        actualizapwm3();
    }
    }.execute();
}
});
}

private void actualizaEstadoBoton13() {
    if (i13) {
        btn13o.setChecked(true);
        btn13i.setChecked(true);
    } else {
        btn13o.setChecked(false);
        btn13i.setChecked(false);
    }
}

private void actualizaEstadoBoton12() {
    if (i12) {
        btn12o.setChecked(true);
        btn12i.setChecked(true);
    } else {
        btn12o.setChecked(false);
        btn12i.setChecked(false);
    }
}

private void actualizaEstadoBoton11() {
    if (i11) {
        btn11o.setChecked(true);
        btn11i.setChecked(true);
    } else {
        btn11o.setChecked(false);
        btn11i.setChecked(false);
    }
}

private void actualizaEstadoBoton10() {
    if (i10) {
        btn10o.setChecked(true);
        btn10i.setChecked(true);
    } else {
        btn10o.setChecked(false);
        btn10i.setChecked(false);
    }
}

```



```
    }  
  }  
  
  private void actualizaEstadoBoton9() {  
    if (i9) {  
      btn9o.setChecked(true);  
      btn9i.setChecked(true);  
    } else {  
      btn9o.setChecked(false);  
      btn9i.setChecked(false);  
    }  
  }  
}  
  
private void actualizaEstadoBoton8() {  
  if (i8) {  
    btn8o.setChecked(true);  
    btn8i.setChecked(true);  
  } else {  
    btn8o.setChecked(false);  
    btn8i.setChecked(false);  
  }  
}  
}  
  
private void actualizaEstadoBoton7() {  
  if (i7) {  
    btn7o.setChecked(true);  
    btn7i.setChecked(true);  
  } else {  
    btn7o.setChecked(false);  
    btn7i.setChecked(false);  
  }  
}  
}  
  
private void actualizaEstadoBoton6() {  
  if (i6) {  
    btn6o.setChecked(true);  
    btn6i.setChecked(true);  
  } else {  
    btn6o.setChecked(false);  
    btn6i.setChecked(false);  
  }  
}  
}  
  
private void actualizaEstadoBoton5() {  
  if (i5) {  
    btn5o.setChecked(true);  
    btn5i.setChecked(true);  
  } else {  
    btn5o.setChecked(false);  
    btn5i.setChecked(false);  
  }  
}  
}  
  
private void actualizaEstadoBoton4() {  
  if (i4) {  
    btn4o.setChecked(true);  
    btn4i.setChecked(true);  
  } else {  
    btn4o.setChecked(false);  
    btn4i.setChecked(false);  
  }  
}  
}  
  
private void actualizaEstadoBoton3() {  
  if (i3) {
```





```

        btn3o.setChecked(true);
        btn3i.setChecked(true);
    } else {
        btn3o.setChecked(false);
        btn3i.setChecked(false);
    }
}

private void actualizaEstadoBoton2() {
    if (i2) {
        btn2o.setChecked(true);
        btn2i.setChecked(true);
    } else {
        btn2o.setChecked(false);
        btn2i.setChecked(false);
    }
}

private void actualizapwm11() {
    sb11.setProgress(pwm11);
}

private void actualizapwm10() {
    sb10.setProgress(pwm10);
}

private void actualizapwm9() {
    sb9.setProgress(pwm9);
}

private void actualizapwm6() {
    sb6.setProgress(pwm6);
}

private void actualizapwm5() {
    sb5.setProgress(pwm5);
}

private void actualizapwm3() {
    sb3.setProgress(pwm3);
}

/*****
/*
/*
/* Accion botones
/*
/* Envio Android a Arduino
/*
/*
/*
*****/

//Metodo invocado cuando el boton switch es pulsado.
@Override
public void onClick(View v) {

    //Dependiendo del identificador del switch, establecido en
el layout,
    //atenderemos el caso. (Invocacion por pulsacion).

    switch(v.getId()){

```



```

case R.id.switch12:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            //El método get recupera cualquier
información
            //identificada por la petición uri. Si
esa petición se refiere a un
            //proceso de producción de datos, es que
los datos producidos
            //que fueron devueltos son como entidad
en la respuesta y no
            // como texto de origen del proceso, a
menos que el texto
            //pasa a ser la salida del proceso.
HttpGet httpget_ = new
HttpGet(web_servicel3o);
            i13 = !i13;

            String aux = (i13) ? "1" : "0";
            //Asociamos a la cadena url, definida al
principio del programa, la
            //variable aux que toma valores de 0 ó 1.
aux = web_serviced13 + "/" + aux;
            //Recuperamos la información, esta vez con la
cadena completa para
            //que Arduino la interprete correctamente
HttpGet httpget = new HttpGet(aux);
Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i13){
        btn13i.setChecked(false);
    }else{
        btn13i.setChecked(true);
    }
    break;

case R.id.switch11:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_servicel2o);
            i12 = !i12;

            String aux = (i12) ? "1" : "0";

```



```

        aux = web_serviced12 + "/" + aux;
        HttpGet httpget = new HttpGet(aux);
        Log.i(TAG, aux);

        try {
            httpClient.execute(httpget_);
            httpClient.execute(httpget);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}.execute();

if(i12){
    btn12i.setChecked(false);
}else{
    btn12i.setChecked(true);
}
break;

case R.id.switch10:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet(web_servicello);
            i11 = !i11;

            String aux = (i11) ? "1" : "0";
            aux = web_serviced11 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i11){
        btn11i.setChecked(false);
    }else{
        btn11i.setChecked(true);
    }
    break;

case R.id.switch9:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();

```



```

        HttpGet httpget_ = new
HttpGet(web_service10o);
        i10 = !i10;

        String aux = (i10) ? "1" : "0";
        aux = web_serviced10 + "/" + aux;
        HttpGet httpget = new HttpGet(aux);
        Log.i(TAG, aux);

        try {
            httpClient.execute(httpget_);
            httpClient.execute(httpget);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}.execute();

if(i10){
    btn10i.setChecked(false);
}else{
    btn10i.setChecked(true);
}
break;

case R.id.switch8:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service9o);
            i9 = !i9;

            String aux = (i9) ? "1" : "0";
            aux = web_serviced9 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i9){
        btn9i.setChecked(false);
    }else{
        btn9i.setChecked(true);
    }
    break;

case R.id.switch7:

    new AsyncTask<Void, Void, Void>() {

```



```

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service8o);
            i8 = !i8;

            String aux = (i8) ? "1" : "0";
            aux = web_serviced8 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i8){
        btn8i.setChecked(false);
    }else{
        btn8i.setChecked(true);
    }
    break;

case R.id.switch6:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service7o);
            i7 = !i7;

            String aux = (i7) ? "1" : "0";
            aux = web_serviced7 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i7){
        btn7i.setChecked(false);
    }else{
        btn7i.setChecked(true);
    }
    break;

```



```

case R.id.switch5:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service60);

            i6 = !i6;

            String aux = (i6) ? "1" : "0";
            aux = web_serviced6 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i6){
        btn6i.setChecked(false);
    }else{
        btn6i.setChecked(true);
    }
    break;

case R.id.switch4:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service50);

            i5 = !i5;

            String aux = (i5) ? "1" : "0";
            aux = web_serviced5 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i5){

```



```

        btn5i.setChecked(false);
    }else{
        btn5i.setChecked(true);
    }
    break;

case R.id.switch3:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service4o);
            i4 = !i4;

            String aux = (i4) ? "1" : "0";
            aux = web_serviced4 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    if(i4){
        btn4i.setChecked(false);
    }else{
        btn4i.setChecked(true);
    }
    break;

case R.id.switch2:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service3o);
            i3 = !i3;

            String aux = (i3) ? "1" : "0";
            aux = web_serviced3 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

```



```

        return null;
    }
}.execute();

if(i3){
    btn3i.setChecked(false);
}else{
    btn3i.setChecked(true);
}
break;

case R.id.switch1:

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpget_ = new
HttpGet(web_service2o);

            i2 = !i2;

            String aux = (i2) ? "1" : "0";
            aux = web_serviced2 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget_);
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }

            return null;
        }
    }.execute();

    if(i2){
        btn2i.setChecked(false);
    }else{
        btn2i.setChecked(true);
    }
    break;
}

}

//Metodo de invocacion de llamada al pulsar el seekbar
//Consta de tres funciones:
//onProgressChanged se activa cuando el seekbar esta siendo
moviendo por el usuario.
//aqui es donde enviremos la peticion para ver de forma continua
el desarrollo del
// pin analogico pwm en tiempo real.
@Override
public void onProgressChanged(SeekBar seekBar, int progress,
boolean fromUser) {
    Log.v("", "" + seekBar);

    //Al igual que con el switch, a traves del identificador del
layout, sabremos
//que seekbr ha sido pulsado y desplazado.

```





```
switch (seekBar.getId()) {

    case R.id.seekBar6:
        txt11.setText("" + progress);
        pwm11 = progress;

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new
DefaultHttpClient();
                aol1=String.valueOf(pwm11);

                String aux = aol1;
                aux = web_servicepwm11 + "/" + aux;
                HttpGet httpget = new HttpGet(aux);
                Log.i(TAG, aux);

                try {
                    httpClient.execute(httpget);
                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }
        }.execute();

        sb11.setProgress(pwm11);
        break;

    case R.id.seekBar5:
        txt10.setText("" + progress);
        pwm10=progress;

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new
DefaultHttpClient();
                aol0=String.valueOf(pwm10);

                String aux = aol0;
                aux = web_servicepwm10 + "/" + aux;
                HttpGet httpget = new HttpGet(aux);
                Log.i(TAG, aux);

                try {
                    httpClient.execute(httpget);
                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }
        }.execute();

        sb10.setProgress(pwm10);
        break;

    case R.id.seekBar4:
        txt9.setText("" + progress);
```



```
pwm9=progress;

new AsyncTask<Void, Void, Void>() {

    @Override
    protected Void doInBackground(Void... params) {
        HttpClient httpClient = new
DefaultHttpClient();
        ao9=String.valueOf(pwm9);

        String aux = ao9;
        aux = web_servicepwm9 + "/" + aux;
        HttpGet httpget = new HttpGet(aux);
        Log.i(TAG, aux);

        try {
            httpClient.execute(httpget);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}.execute();

sb9.setProgress(pwm9);
break;

case R.id.seekBar3:
    txt6.setText("" + progress);
    pwm6=progress;

    new AsyncTask<Void, Void, Void>() {

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
            ao6=String.valueOf(pwm6);

            String aux = ao6;
            aux = web_servicepwm6 + "/" + aux;
            HttpGet httpget = new HttpGet(aux);
            Log.i(TAG, aux);

            try {
                httpClient.execute(httpget);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }.execute();

    sb6.setProgress(pwm6);
    break;

case R.id.seekBar2:
    txt5.setText("" + progress);
    pwm5=progress;

    new AsyncTask<Void, Void, Void>() {

        @Override
```



```

        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();
                ao5=String.valueOf(pwm5);

                String aux = ao5;
                aux = web_servicepwm5 + "/" + aux;
                HttpGet httpget = new HttpGet(aux);
                Log.i(TAG, aux);

                try {
                    httpClient.execute(httpget);
                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }
        }.execute();

        sb5.setProgress(pwm5);
        break;

    case R.id.seekBar1:
        txt3.setText("" + progress);
        pwm3=progress;

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new
DefaultHttpClient();
                    ao3=String.valueOf(pwm3);

                    String aux = ao3;
                    aux = web_servicepwm3 + "/" + aux;
                    HttpGet httpget = new HttpGet(aux);
                    Log.i(TAG, aux);

                    try {
                        httpClient.execute(httpget);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    return null;
                }
            }.execute();

            sb3.setProgress(pwm3);
            break;
        }

        //Funcion onStartTrackingTouch. Aqui se describirán las
instrucciones necesarias
        //cuando el usuario inicia el movimiento.(Por ejemplo, un aviso de
que ha comenzado
        //a desplazar el seekbar)
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
        }

        //onStopTrackingTouch, se inicia cuando el usuario deja de mover el
seekbar

```



```
@Override
public void onStopTrackingTouch(SeekBar seekBar) {
}

//Funcion declarada en la clase AjustesFragment por si no consigue
pasar los
//parametros. Indica el valor del pin2 que debe tomar.
public void setPin2(String p22) {
    // TODO Auto-generated method stub
    this.p2=p22;
}
}
```



## DrawerItem.java

```
package com.example.android.navigationdrawerexample;

/*****
*/
/*
*/
/* Clase creada para comunicar los iconos y titulos del menu lateral
*/
/* entre los diferentes freagments
*/
/* Los recursos obtenidos reciben instrucciones en la Actividad principal
*/
/* y posteriormente se derivan a los fragmentos
*/
*/
*/
/*****
*****/
public class DrawerItem {

    private String name;
    private int iconId;

    public DrawerItem(String name, int iconId) {
        this.name = name;
        this.iconId = iconId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getIconId() {
        return iconId;
    }

    public void setIconId(int iconId) {
        this.iconId = iconId;
    }
}
```



## DrawerListAdapter.java

```

package com.example.android.navigationdrawerexample;

import java.util.List;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;

/*****
****/
/*
*/
/* Esta clase actualiza el menu lateral en forma de lista con los
*/
/* parametros del texview e image creados en el layout
*/
/* Dependiendo de la posicion y la llamada a la clase DrawerItem, el
icono */
/* de la image y el texto se actualizaran dependiendo de la posicion en
la */
/* que el usuario seleccione la opcion del menu
*/
/*
*/
/*****
****/

public class DrawerListAdapter extends ArrayAdapter {

    public DrawerListAdapter(Context context, List objects) {
        super(context, 0, objects);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {

        if(convertView == null){
            LayoutInflater inflater =
            (LayoutInflater)parent.getContext().
            getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(R.layout.drawer_list_item,
            null);
        }

        ImageView icon = (ImageView) convertView.findViewById(R.id.icon);
        TextView name = (TextView) convertView.findViewById(R.id.text1);

        DrawerItem item = (DrawerItem) getItem(position);
        icon.setImageResource(item.getIconId());
        name.setText(item.getName());

        return convertView;
    }

}

```



## FileOperations.java

```

package com.example.android.navigationdrawerexample;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import android.util.Log;

/*****
****/
/*
*/
/* En esta clase se crea el fichero de lectura y escritura donde se
guardará*/
/* la configuracion de los pines de la clase AjustesFragment
*/
/*
*/
/*****
****/
public class FileOperations {

    public FileOperations() {

        //Escritura en el fichero en la ruta determinada.
        public Boolean write(String fname,String fpin){
            try {

                String fpath = "/sdcard/Arduino_Yun/"+fname+".txt";
                File file = new File(fpath);

                //Si el fichero no existe lo crea
                if (!file.exists()) {
                    file.createNewFile();
                }

                //FileWriter escribe un archivo en una parte especifica del
                sistema
                //getAbsolutePath devuelve un archivo nuevo usando la ruta
                absoluta.
                FileWriter fw = new FileWriter(file.getAbsolutePath());

                //Escribe el texto en un flujo de datos almacenando los
                caracteres
                //con el fin de asegurar la escritura de los caracteres
                individuales.
                BufferedWriter bw = new BufferedWriter(fw);
                //Llama al flujo de datos con el valor "fpin" que recibio la
                funcion.
                bw.write(fpin);
                //Cierra el flujo de datos
                bw.close();

                Log.d("Sucess", "Sucess");
                return true;

            } catch (IOException e) {

```



```
        e.printStackTrace();
        return false;
    }
}

//Lectura del fichero con el nombre guardado en la variable
"fname".
public String read(String fname){

    //Lee el texto del flujo de datos almacenado
    //con el fin de asegurar la lectura de los caracteres individuales.
    //Inicializa la lectura con valor null para evitar lecturas
    erroneas o antiguas.
    BufferedReader br = null;
    String response = null;

    try {

        //Inicializa el flujo de datos
        StringBuffer output = new StringBuffer();

        String fpath = "/sdcard/Arduino_Yun/"+fname+".txt";

        //Lee del flujo de datos situado en la ruta destino.
        br = new BufferedReader(new FileReader(fpath));
        String line = "";

        //Lee hasta que finalice la línea del fichero.
        while ((line = br.readLine()) != null) {
            output.append(line);
        }
        //Cierra el flujo de datos
        br.close();
        //Devuelve una cadena con el valor del parametro leído.
        response = output.toString();

    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }

    return (response);
}
}
```





## FragmentSplashActivity.java

```

package com.example.android.navigationdrawerexample;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;

/*****
/*
/*
/*
/*
/*
/*
/*
/*****

public class FragmentSplashActivity extends Activity{

    long Delay = 3000;

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

        // Eliminamos el ActionBar
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        // Mostramos la vista creada en el layout .xml
        setContentView(R.layout.fragment_splash_activity);

        //Constructor cabecera. Crea un hilo y no atiende a otras
ejecuciones.
        //Realiza sus instrucciones y finaliza.
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                //En el momento de iniciar la aplicación, pasaremos de
esta actividad
                //a la actividad principal a traves del parametro Intent.
                startActivity(new Intent(FragmentSplashActivity.this,
MainActivity.class));
                finish();
            }
        }, 3000); //Duracion de la vista que verá el usuario al iniciar
la aplicacion.

    }

}

```



## MainActivity.java

```
package com.example.android.navigationdrawerexample;

import java.util.ArrayList;

import android.app.Fragment;
import android.app.FragmentManager;
import android.content.Intent;
import android.content.res.Configuration;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.ActionBarDrawerToggle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;

/*****
 *
 * Actividad principal donde se crea la estructura principal
 * de la aplicacion. En este caso es un cajon con menu
 * lateral oculto, llamado navigationdrawer.
 *
 *****/

public class MainActivity extends FragmentActivity implements
AjustesFragment.OnFragmentClickListener {

    /*****
     *
     * Definicion de los atributos de la clase
     *
     *****/
    private DrawerLayout mDrawerLayout;
    private ListView mDrawerList;
    private ActionBarDrawerToggle mDrawerToggle;

    private CharSequence mDrawerTitle;
    private CharSequence mTitle;
    private String[] OpcionesTitles;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Mostramos el contenido del layout creado.
        setContentView(R.layout.activity_main);

        //getTitle asocia el titulo de la aplicacion a un parametro.
        mTitle = mDrawerTitle = getTitle();
        //Devuelve los recursos asociados y el vector donde se han
definido
        //los titulos del menu desplegable

```



```
OpcionesTitles = getResources().getStringArray(R.array.opciones);
//Definicion de atributos creados en el layout
mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
mDrawerList = (ListView) findViewById(R.id.left_drawer);

//Instancia un archivo layout .xml correspondiente al objeto
vista.
LayoutInflater inflater = getLayoutInflater();
//ViewGroup (Vista padre). Vista que contiene otras vistas(Vistas
hijos).
//Inflamos el layout cabecera que contendra el menu lateral.
ViewGroup header = (ViewGroup)
inflater.inflate(R.layout.header,mDrawerList,
false);
//Colocamos la cabecera en la parte superior del menu lateral.
mDrawerList.addHeaderView(header, null, false);

//Llamamos a un vector que enviaremos a otra clase donde
// gestionara los iconos y textos del menu lateral
ArrayList<DrawerItem> items = new ArrayList<DrawerItem>();
items.add(new
DrawerItem(OpcionesTitles[0],R.drawable.ic_action_settings));
items.add(new
DrawerItem(OpcionesTitles[1],R.drawable.ic_action_gamepad));
items.add(new
DrawerItem(OpcionesTitles[2],R.drawable.ic_action_network_wifi));

//Establece una sombra a medida que se superpone el contenido
//principal cuando se abre el cajón (menu lateral)
mDrawerLayout.setDrawerShadow(R.drawable.drawer_shadow,
GravityCompat.START);

//Configura la vista de la lista del cajón con elementos
mDrawerList.setAdapter(new DrawerListAdapter(this, items));

//Metodo de llamada cuando el boton del menu es pulsado
mDrawerList.setOnItemClickListener(new
DrawerItemClickListener());

//Habilita el icono de la aplicacion en la ActionBar para que se
//comporte como un boton y pueda cambiar la navegacion del cajon
getActionBar().setDisplayHomeAsUpEnabled(true);
getActionBar().setHomeButtonEnabled(true);

mDrawerToggle = new ActionBarDrawerToggle(
this, /* Actividad donde se desarrolla
*/
mDrawerLayout, /* Objeto DrawerLayout */
R.drawable.ic_drawer, /* Imagen donde abrir el menu */
R.string.drawer_open, /* "open drawer" descripcion para
acceder */
R.string.drawer_close /* "close drawer" descripcion para
acceder */
) {
public void onDrawerClosed(View view) {
getActionBar().setTitle(mTitle);
invalidateOptionsMenu(); // crea la llamada a
onPrepareOptionsMenu().
}

public void onDrawerOpened(View drawerView) {
getActionBar().setTitle(mDrawerTitle);
```



```

        invalidateOptionsMenu(); // crea la llamada a
onPrepareOptionsMenu().
    }
};
mDrawerLayout.setDrawerListener(mDrawerToggle);

//Al iniciar la aplicacion, mostramos la seleccion 1 del menu
lateral (Ajustes)
if (savedInstanceState == null) {
    selectItem(1);
}

}

//Llamada para invocar el menu superior en la ActionBar.
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main, menu);
    return super.onCreateOptionsMenu(menu);
}

//Llamada para invocar las opciones del menu superior
@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    // Si el menu lateral esta abierto, oculta los items contenidos
en la vista
    boolean drawerOpen = mDrawerLayout.isDrawerOpen(mDrawerList);
    return super.onPrepareOptionsMenu(menu);
}

//Llamada para invocar las opciones del menu lateral
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    if (mDrawerToggle.onOptionsItemSelected(item)) {
        return true;
    }

    switch(item.getItemId()) {
    case R.id.ic_action_copy:
        //Accedemos a una vista emergente de archivos compatibles
que puedan
        //acceder a la accion enviada por la intencion (intent).
        //En este caso una ruta a un directorio especifico.
        Intent intent = new Intent(Intent.ACTION_VIEW);

        intent.setDataAndType(Uri.parse("file:///mnt/sdcard/Arduino_Yun"),
        "**/*");
        startActivity(intent);

        return true;

    case R.id.action_tutorial:
        //Con Intent pasamos de la actividad principal
        //a la actividad "tutorial" donde encontramos la ayuda
para la app.
        startActivity(new Intent(MainActivity.this,
Tutorial.class));
    default:
    }

    return super.onOptionsItemSelected(item);
}
}

```



```
//Metodo de invocacion de llamada del ListView en el menu lateral */
private class DrawerItemClickListener implements
ListView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        selectItem(position);
        //parent. AdapterView donde se ha hecho click.
        //view. Vista seleccionada del vector.
        //position. Posicion de la vista en el adapter.
        //id. identificador del item clicado.
    }
}

//Funcion que gestiona la posicion del item seleccionado
private void selectItem(int position) {
    //Declaracion del parametro fragment como null.
    Fragment fragment = null;

    //Dependiendo de la posicion seleccionada accederemos a un
fragment determinado.
    switch (position) {
        case 1:
            fragment = new AjustesFragment();
            break;
        case 2:
            fragment = new ControlesFragment();
            break;
        case 3:
            fragment = new SensoresFragment();
            break;

        default:
            break;
    }

    //Si el fragment no esta vacio, los gestionamos a traves de una
interfaz.
    if (fragment != null) {
        FragmentManager fragmentManager = getFragmentManager();

        fragmentManager.beginTransaction().replace(R.id.content_frame,
fragment).commit();

        //Comprueba el estado de la posicion del menu lateral
mDrawerList.setItemChecked(position, true);
        //Pone el valor del item actual seleccionado.
mDrawerList.setSelection(position);
        //Asociamos la seleccion anterior al nombre de los titulos.
        //Le restamos una posicion ya que incluimos en primera
posicion
        //del ListView la una cabecera que no contiene titulo ni puede
ser seleccionada.
        setTitle(OpcionesTitles[position-1]);
        //Cierra el cajon
mDrawerLayout.closeDrawer(mDrawerList);
    } else {
        //Implementacion para depurar el error si el fragment no se
ha creado.
        Log.e("MainActivity", "Error in creating fragment");
    }
}

//Cambia el titulo asociado con esta actividad.
```



```
@Override
public void setTitle(CharSequence title) {
    mTitle = title;
    getActionBar().setTitle(mTitle);
}

//Llamada cuando la actividad se ha completado
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Sincroniza el estado del cajon despues de invocar
onRestoreInstanceState.
    mDrawerToggle.syncState();
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    // Pasa cualquier configuracion cambiada al cajon de navegacion
mDrawerToggle.onConfigurationChanged(newConfig);
}

//La Interfaz creada en el Fragment Ajustese necesita pasar por la
actividad
//principal para enviarla a otro fragmento.
//Esta funcion recibe los parametros a pasar al fragment.
@Override
public void OnFragmentClick(String p2, String p3, String p4,
String p5,
String p6, String p7, String p8, String p9, String
p10,
String p11, String p12, String p13) {

    ControlesFragment controlfrag = new ControlesFragment();

    //Inicializa los argumentos a pasar al fragment
Bundle args = new Bundle();

    //Enviamos las variables al fragment declarado anteriormente
y le asignamos
    //un nombre clave.
    //Con setArguments enviamos los argumentos que necesitamos
args.putString(controlfrag.CONFIG_PIN, p2);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN3, p3);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN4, p4);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN5, p5);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN6, p6);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN7, p7);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN8, p8);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN9, p9);
controlfrag.setArguments(args);
}
```



```
args.putString(controlfrag.CONFIG_PIN10, p10);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN11, p11);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN12, p12);
controlfrag.setArguments(args);

args.putString(controlfrag.CONFIG_PIN13, p13);
controlfrag.setArguments(args);

//Importante declarar primero setArguments antes que iniciar
el fragment
//ya que si no los argumentos a traves de Bundle no se
pasarán.
//Ahora remplazamos el fragment actual por el fragment destino
que queremos mostrar.
getFragmentManager().beginTransaction()
    .replace(R.id.content_frame, controlfrag).commit();
    }
}
```



## SensoresFragment.java

```

package com.example.android.navigationdrawerexample;

import java.io.ByteArrayOutputStream;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.app.Fragment;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;

public class SensoresFragment extends Fragment {

    /**
     *
     * Definicion de los atributos de la clase y las direcciones */
    /**
     *
     *
     */

    public static final String TAGSensores = "Inicio";
    private String web_servicea0 =
"http://192.168.240.1/arduino/analog/A0";
    private String web_servicea1 =
"http://192.168.240.1/arduino/analog/A1";
    private String web_servicea2 =
"http://192.168.240.1/arduino/analog/A2";
    private String web_servicea3 =
"http://192.168.240.1/arduino/analog/A3";
    private String web_servicea4 =
"http://192.168.240.1/arduino/analog/A4";
    private String web_servicea5 =
"http://192.168.240.1/arduino/analog/A5";

    private ProgressBar pba0, pba1, pba2, pba3, pba4, pba5;
    private TextView txta0, txta1, txta2, txta3, txta4,txta5;
    private Button btn, btnnot;
    private String a0, a1, a2, a3, a4, a5;
    private String A0, A1, A2, A3, A4, A5;

    public String pina0 = "PinA0";
    public String pina1 = "PinA1";
    public String pina2 = "PinA2";
    public String pina3 = "PinA3";
    public String pina4 = "PinA4";
    public String pina5 = "PinA5";

    public String AI="AI";

```





```

/*****/
/*
    */
/* Constructor vacío necesario para que nuestra aplicación */
/* no tenga fallos a la hora de recrear el Fragment. */
/*
    */
/*****/
public SensoresFragment() {
}

/*****/
/*
    */
/* --OnCreateView-- */
/* El sistema lo llama cuando es el momento de que el */
/* fragmento dibuje la IU por primera vez. */
/* Para dibujar una IU para el fragmento, hay que devolver */
/* un View desde este método que es la raíz del layout del */
/* fragmento. */
/* Se puede devolver null si el fragmento no suministra */
/* una IU. */
/*
    */
/*****/

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {

    View v = inflater.inflate(R.layout.fragment_sensores, container,
false);

/*****/
/*
    */
/* Definición e inicialización de los objetos. */
/*
    */
/*
    */
/*****/

pba0 = (ProgressBar) v.findViewById(R.id.progressBar1);
pba1 = (ProgressBar) v.findViewById(R.id.progressBar2);
pba2 = (ProgressBar) v.findViewById(R.id.progressBar3);
pba3 = (ProgressBar) v.findViewById(R.id.progressBar4);
pba4 = (ProgressBar) v.findViewById(R.id.progressBar5);
pba5 = (ProgressBar) v.findViewById(R.id.progressBar6);
txta5 = (TextView) v.findViewById(R.id.textView12);
txta4 = (TextView) v.findViewById(R.id.textView11);
txta3 = (TextView) v.findViewById(R.id.textView10);
txta2 = (TextView) v.findViewById(R.id.textView9);
txta1 = (TextView) v.findViewById(R.id.textView8);
txta0 = (TextView) v.findViewById(R.id.textView7);

//progressbar declarada con un valor maximo de 1024 (Valor
analogico)
pba0.setMax(1024);
pba1.setMax(1024);
pba2.setMax(1024);
pba3.setMax(1024);
pba4.setMax(1024);
pba5.setMax(1024);

//Posicion inicial del progressbar
pba0.setProgress(0);

```



```
pba1.setProgress(0);
pba2.setProgress(0);
pba3.setProgress(0);
pba4.setProgress(0);
pba5.setProgress(0);

btn = (Button) v.findViewById(R.id.button1);
btnnot = (Button) v.findViewById(R.id.button2);

return v;
}

/*****
/*
*/
/* Metodo de llamada cuando el fragmento esta siendo visible */
/* ante el usuario.
*/
/*
*/
*****/

@Override
public void onStart() {
    super.onStart();

    //Crea una nueva clase para realizar la invocacion posterior.
    FileOperations fop = new FileOperations();
    //Envio de parametros a la nueva clase
    String cargaa0 = fop.read(pina0);

    if(cargaa0 != null){
        //Si el fichero existe, leemos de la memoria interna y le
        asociamos ese valor
        //al textView que se creo a traves del layout
        txta0.setText(cargaa0);
        //Asignamos ese valor a una variable
        a0=txta0.getText().toString();

        //CompareTo compara los String recibidos, devolviendo el
        valor cero si son iguales,
        // -1 s es menor o 1 si es mayor.
        if(a0.compareTo(AI) == 0){

            //Creacion de una tarea asincrona. Necesaria para
            evitar que una tarea bloquee
            //el hilo principal de la interfaz de usuario.
            new AsyncTask<Void, Void, Void>() {

                //Llamada despues del OnPreExecute(En este caso
                no era necesario implementarlo)
                //Es aqui donde se realiza la tarea principal
                @Override
                protected Void doInBackground(Void... params) {
                    //HttpClient crea una interfaz para un cliente
                    HTTP.

                    //El cliente encapsula los objetos necesarios
                    para ejecutar las peticiones HTTP
                    //durante la autentificacion, gestion de la
                    conexion y otras características
                    HttpClient httpclient = new
                    DefaultHttpClient();
```



```

//try/catch es un bloque que gestiona los fallos que se pueden ocasionar
//Dentro del bloque try colocamos las funciones que podrian porvocar fallo
//si esto ocurre, saltamos al bloque catch que gestionara el error.
        try {
//Recibimos la respuesta HTTP y ejecutamos nuestra peticion.
//Enviamos la URL que interpretará Arduino
        HttpResponse response = httpClient.execute(new HttpGet(web_servicea0));
//Esta clase implementa un flujo de salida en el que los datos se por defecto). El buffer crece automáticamente como se //escriben en una matriz de bits (32bits //escriben datos en ella.
        ByteArrayOutputStream out = new ByteArrayOutputStream();
//Escribe los datos obtenidos en el flujo y escribe en la cadena enviada.
        response.getEntity().writeTo(out);
//cierra el flujo de datos
        out.close();

//Recibimos una respuesta de Arduino de la lectura del pin
//Para pines analogicos, el valor del pin.
        String respuesta = out.toString();

//Recibiremos una respuesta ASCII con el valor del pin analogico
        int aux=Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
//Se convierte a entero y se actualiza la progressBar.
        pba0.setProgress(aux);

//Se convierte a String y lo guardamos en el textView para
//visualizarlo en la vista del layout
        A0 = String.valueOf(aux);
        txta0.setText(""+ A0);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

//Una vez terminado el metodo anterior, llamamos a onPostExecute
//En este caso no se realiza nada.
@Override
protected void onPostExecute(Void aVoid) {

}

}.execute();
}

```



```

    }

    String cargaal = fop.read(pinal);

    if(cargaal != null){

        txtal.setText(cargaal);
        a1=txtal.getText().toString();

        if(a1.compareTo(AI) == 0){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void doInBackground(Void... params) {
                    HttpClient httpclient = new
DefaultHttpClient();

                    try {
                        HttpResponse response =
httpclient.execute(new HttpGet(web_serviceal));
                        ByteArrayOutputStream out = new
ByteArrayOutputStream();

                        response.getEntity().writeTo(out);
                        out.close();

                        String respuesta = out.toString();

                        int aux =
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                        pbal.setProgress(aux);

                        A1 = String.valueOf(aux);
                        txtal.setText(""+ A1);

                    } catch (Exception e) {
                        e.printStackTrace();
                    }

                    return null;
                }

                @Override
                protected void onPostExecute(Void aVoid) {

                }

            }.execute();
        }
    }

    String cargaa2 = fop.read(pina2);

    if(cargaa2 != null){

        txta2.setText(cargaa2);
        a2=txta2.getText().toString();

        if(a2.compareTo(AI) == 0){

            new AsyncTask<Void, Void, Void>() {

```



```

        @Override
        protected Void doInBackground(Void... params) {
            HttpClient httpClient = new
DefaultHttpClient();

            try {
                HttpResponse response =
httpClient.execute(new HttpGet(web_servicea2));
                ByteArrayOutputStream out = new
ByteArrayOutputStream();
                response.getEntity().writeTo(out);
                out.close();

                String respuesta = out.toString();
                //Si todo ha ido bien, recibiremos una
                respuesta ASCII con el valor de A0
                // Se convierte a entero y se actualiza
                la progressBar.
                int aux=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                pba2.setProgress(aux);

                A2 = String.valueOf(aux);
                txta2.setText(""+ A2);

            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {
        }.execute();
    }
}

String cargaa3 = fop.read(pina3);

if(cargaa3 != null){

    txta3.setText(cargaa3);
    a3=txta3.getText().toString();

    if(a3.compareTo(AI) == 0){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new
DefaultHttpClient();

                try {
                    HttpResponse response =
httpClient.execute(new HttpGet(web_servicea3));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();
                    response.getEntity().writeTo(out);
                    out.close();

                    String respuesta = out.toString();

```



```

//Si todo ha ido bien, recibiremos una
respuesta ASCII con el valor de A0
// Se convierte a entero y se actualiza
la progressBar.
        int aux=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

        pba3.setProgress(aux);

        A3 = String.valueOf(aux);
        txta3.setText(""+ A3);

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void aVoid) {
}
}.execute();
}

String cargaa4 = fop.read(pina4);

if(cargaa4 != null){

    txta4.setText(cargaa4);
    a4=txta4.getText().toString();

    if(a4.compareTo(A1) == 0){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpClient = new
DefaultHttpClient();

                try {
                    HttpResponse response =
httpClient.execute(new HttpGet(web_servicea4));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();

                    response.getEntity().writeTo(out);
                    out.close();

                    String respuesta = out.toString();
                    //Si todo ha ido bien, recibiremos una
respuesta ASCII con el valor de A0
                    // Se convierte a entero y se actualiza
la progressBar.
                    int aux=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                    pba4.setProgress(aux);

                    A4 = String.valueOf(aux);
                    txta4.setText(""+ A4);

                } catch (Exception e) {
                    e.printStackTrace();

```



```

    }
    return null;
}

@Override
protected void onPostExecute(Void aVoid) {
}
}.execute();
}

String cargaa5 = fop.read(pina5);

if(cargaa5 != null){

    txta5.setText(cargaa5);
    a5=txta5.getText().toString();

    if(a5.compareTo(AI) == 0){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                HttpClient httpclient = new
DefaultHttpClient();

                try {
                    HttpResponse response =
httpclient.execute(new HttpGet(web_servicea5));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();

                    response.getEntity().writeTo(out);
                    out.close();

                    String respuesta = out.toString();
                    //Si todo ha ido bien, recibiremos una
respuesta ASCII con el valor de A0
                    // Se convierte a entero y se actualiza
la progressBar.
                    int aux=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                    pba5.setProgress(aux);

                    A5 = String.valueOf(aux);
                    txta5.setText(""+ A5);

                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }

            @Override
            protected void onPostExecute(Void aVoid) {
            }
        }.execute();
    }

}

//Metodo de invocacion al pulsar el boton
btn.setOnClickListener( new OnClickListener(){

```



```

@Override
public void onClick(View v) {

    FileOperations fop = new FileOperations();
    String cargaa0 = fop.read(pina0);

    if(cargaa0 != null){

        txta0.setText(cargaa0);
        a0=txta0.getText().toString();

        if(a0.compareTo(AI) == 0){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void doInBackground(Void...
params) {

                    HttpClient httpclient = new
DefaultHttpClient();

                    try {
                        HttpResponse response =
httpclient.execute(new HttpGet(web_servicea0));
                        ByteArrayOutputStream out = new
ByteArrayOutputStream();
                        response.getEntity().writeTo(out);

                        out.close();

                        String respuesta =

                                int aux=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                                pba0.setProgress(aux);

                                A0 = String.valueOf(aux);
                                txta0.setText(""+ A0);

                            } catch (Exception e) {
                                e.printStackTrace();
                            }
                            return null;
                        }

                @Override
                protected void onPostExecute(Void aVoid)
{

                    }

                }.execute();
            }

        }

        String cargaal = fop.read(pinal);

        if(cargaal != null){

            txtal.setText(cargaal);
            al=txtal.getText().toString();

            if(al.compareTo(AI) == 0){

```





```

        new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void...
params) {
                HttpClient httpclient = new
DefaultHttpClient();

                try {
                    HttpResponse response =
httpclient.execute(new HttpGet(web_serviceal));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();
                    response.getEntity().writeTo(out);
                    out.close();
                    String respuesta =
out.toString();

                    int aux =
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                    pba1.setProgress(aux);

                    A1 = String.valueOf(aux);
                    txta1.setText(""+ A1);

                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }

            @Override
            protected void onPostExecute(Void aVoid)
{
            }
        }.execute();
    }

    String cargaa2 = fop.read(pina2);

    if(cargaa2 != null){

        txta2.setText(cargaa2);
        a2=txta2.getText().toString();

        if(a2.compareTo(AI) == 0){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void doInBackground(Void...
params) {
                    HttpClient httpclient = new
DefaultHttpClient();

                    try {
                        HttpResponse response =
httpclient.execute(new HttpGet(web_servicea2));

```



```

        ByteArrayOutputStream out = new
ByteArrayOutputStream();
response.getEntity().writeTo(out);
        out.close();

out.toString();
        String respuesta =

        Integer.parseInt(respuesta.replaceAll("[\\D]", ""));
        int aux=
        pba2.setProgress(aux);

        A2 = String.valueOf(aux);
        txta2.setText(""+ A2);

        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid)
{
    }
    }.execute();
}

String cargaa3 = fop.read(pina3);

if(cargaa3 != null){

    txta3.setText(cargaa3);
    a3=txta3.getText().toString();

    if(a3.compareTo(AI) == 0){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void...
params) {

                HttpClient httpClient = new
DefaultHttpClient();

                try {
                    HttpResponse response =
httpClient.execute(new HttpGet(web_servicea3));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();

                    response.getEntity().writeTo(out);

                    out.close();

                    String respuesta =

                    Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                    pba3.setProgress(aux);

```



```

        A3 = String.valueOf(aux);
        txta3.setText(""+ A3);

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void aVoid)
{
}

}.execute();
}
}

String cargaa4 = fop.read(pina4);

if(cargaa4 != null){

    txta4.setText(cargaa4);
    a4=txta4.getText().toString();

    if(a4.compareTo(AI) == 0){

        new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void...
params) {

                HttpClient httpclient = new
DefaultHttpClient();

                try {
                    HttpResponse response =
httpclient.execute(new HttpGet(web_servicea4));
                    ByteArrayOutputStream out = new
ByteArrayOutputStream();

                    response.getEntity().writeTo(out);

                    out.close();

                    String respuesta =

                    int aux=
Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                    pba4.setProgress(aux);

                    A4 = String.valueOf(aux);
                    txta4.setText(""+ A4);

                } catch (Exception e) {
                    e.printStackTrace();
                }
                return null;
            }

            @Override

```



```

                protected void onPostExecute(Void aVoid)
            {
                }
            }.execute();
        }
    }

    String cargaa5 = fop.read(pina5);

    if(cargaa5 != null){

        txta5.setText(cargaa5);
        a5=txta5.getText().toString();

        if(a5.compareTo(AI) == 0){

            new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void doInBackground(Void...
params) {

                    HttpClient httpclient = new
DefaultHttpClient();

                    try {
                        HttpResponse response =
httpclient.execute(new HttpGet(web_servicea5));
                        ByteArrayOutputStream out = new
ByteArrayOutputStream();
                        response.getEntity().writeTo(out);
                        out.close();

                        String respuesta =

                        Integer.parseInt(respuesta.replaceAll("[\\D]", ""));

                        pba5.setProgress(aux);

                        A5 = String.valueOf(aux);
                        txta5.setText(""+ A5);

                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    return null;
                }

                @Override
                protected void onPostExecute(Void aVoid)
            {
                }
            }.execute();
        }
    }
}
    });
}
}
}

```



## Tutorial.java

```

package com.example.android.navigationdrawerexample;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

/*****
/*
/*
/* Actividad creada para ofrecer una ayuda al usuario
/* Creamos un layout con una ayuda interactiva que
/* facilite el uso y apredinzaje
*/
*/
*/
*/
/*****
public class Tutorial extends Activity {

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

        //Declaracion del objeto
        final ImageButton button1 = (ImageButton)
findViewById(R.id.imageButton1);
        //Llamada invocada al pulsar el boton.
        button1.setOnClickListener (new OnClickListener() {

            public void onClick(View v) {

                //Declaracion de los objetos en la vista del
xml
                TextView textView1 =
(TextView) findViewById(R.id.textView1);
                //Le asociamos al textView un texto creado en
el fichero "string".
                textView1.setText(R.string.exControles1);

                ImageView imageView =
                    (ImageView)
findViewById(R.id.imageView1);
                //Ocultamos la imagen inicial creado en el
layout
                imageView.setVisibility(View.INVISIBLE);

                Button btnajustes =
(Button) findViewById(R.id.button1);
                //ocultamos la imagen del boton
                btnajustes.setVisibility(View.INVISIBLE);

```



```

        TextView textview2 =
(TextView) findViewById(R.id. textView2);
        textview2.setText(R.string. exControles2);

        //Nueva llamada al pulsar de nuevo el boton
        button1.setOnClickListener (new
OnClickListener() {

                                //Ahora los objetos mostrados seran
                                //imagenes y asignar diferentes textos
                                //del archivo string.
                                @Override
                                public void onClick(View v) {
                                    // TODO Auto-generated method stub

                                    final ImageButton button1 =
(ImageButton) findViewById(R.id. imageButton1);

                                    TextView textview1 =
(TextView) findViewById(R.id. textView1);

                                    textview1.setText(R.string. exactionbar1);

                                    ImageView imageView =
                                    (ImageView)
findViewById(R.id. imageView1);

                                    imageView.setVisibility(View. VISIBLE);

                                    imageView.setImageResource(R.drawable. ic_action_copyw);

                                    Button btnajustes =
(Button) findViewById(R.id. button1);

                                    btnajustes.setVisibility(View. INVISIBLE);

                                    TextView textview2 =
(TextView) findViewById(R.id. textView2);

                                    textview2.setVisibility(View. INVISIBLE);

                                    button1.setVisibility(View. INVISIBLE);
                                }

                                });

        });

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflador para mostrar el menu. Mostramos los iconos y
        textos que tengamos
        //en en archivo res/menu/main.xml
        getMenuInflater().inflate(R.menu. main, menu);
        return true;
    }

    //Selección del actionBar. Menu superior.

```



```
//Dependiendo del icono seleccionado realizaremos unas
instrucciones
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    switch(item.getItemId()) {
        case R.id.ic_action_copy:
            //Accedemos a una vista emergente de archivos compatibles
que puedan
            //acceder a la accion enviada por la intencion (intent).
            //En este caso una ruta a un directorio especifico.
            Intent intent = new Intent(Intent.ACTION_VIEW);

intent.setDataAndType(Uri.parse("file:///mnt/sdcard/Arduino_Yun"),
            "**/*");
            startActivity(intent);

            return true;

        default:

            return super.onOptionsItemSelected(item);
    }
}
}
```



## activity\_main.xml

```
<!-- Un DrawerLayout es lanzado para ser usado en la primera y mas
externa capa de la vista
usando match_parent para rellenar el espacio a lo alto y ancho. -->
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Como vista principal lanzada, consume todo el espacio disponible
        usando match_parent a ambos lados. -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
    </FrameLayout>

    <!-- android:layout_gravity="start" sirve para tratar el menu
        como deslizante de izquierda a derecha -->

    <ListView
        android:id="@+id/left_drawer"
        android:layout_width="250dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="@color/LightGrey"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="0dp" />

</android.support.v4.widget.DrawerLayout>
```





## drawer\_list\_item.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:padding="5dp">

    <TextView
        android:id="@+id/text1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:gravity="center_vertical"
        android:minHeight="?android:attr/listPreferredItemHeightSmall"
        android:paddingLeft="70dp"
        android:paddingRight="30dp"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="@color/Black" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/icon"
        android:layout_alignParentLeft="true"
        android:layout_marginRight="16dp"
        android:gravity="center_vertical" />
</RelativeLayout>
```



## fragment\_ajustes.xml

```
<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    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"-->

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="0dp"
        tools:context="com.example.arduinoyun.FirstActivity" >

        <ImageView
            android:id="@+id/imageView1"
            android:layout_width="200dp"
            android:layout_height="210dp"
            android:layout_alignParentLeft="true"
            android:layout_alignParentRight="true"
            android:layout_marginTop="50dp"
            android:src="@drawable/arduino_yun" />

        <LinearLayout
            android:id="@+id/linearLayout1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="322dp" >

            <TextView
                android:id="@+id/textView1"
                android:layout_width="110dp"
                android:layout_height="20dp"
                android:layout_marginLeft="60dp"
                android:layout_marginTop="5dp"
                android:text="@string/dirip" />

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="20dp"
                android:layout_marginLeft="68dp"
                android:layout_marginTop="5dp"
                android:text="@string/portip" />

        </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_marginTop="325dp" >

        <MultiAutoCompleteTextView
            android:id="@+id/multiAutoCompleteTextView1"
```



```
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="16dp"
        android:ems="10" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="50dp"
    android:layout_marginTop="16dp"
    android:layout_weight="1"
    android:ems="10"
    android:inputType="number" />

</LinearLayout>

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="700dp"
    android:layout_height="50dp"
    android:layout_alignParentTop="true"
    android:layout_marginTop="10dp"
    android:src="@drawable/pin13" />

<TextView
    android:id="@+id/LblMensaje2"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="305dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje3"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="282dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje4"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="263dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje5"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="240dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje6"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="218dp"
```



```
        android:layout_marginTop="4dp"
        android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje7"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="200dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje8"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="178dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje9"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="156dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje10"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="134dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje11"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="113dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje12"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="94dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensaje13"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="75dp"
    android:layout_marginTop="4dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensajeA0"
    android:layout_width="22dp"
    android:layout_height="15dp"
```



```
        android:layout_marginLeft="215dp"
        android:layout_marginTop="307dp"
        android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensajeA1"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="239dp"
    android:layout_marginTop="307dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensajeA2"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="262dp"
    android:layout_marginTop="307dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensajeA3"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="286dp"
    android:layout_marginTop="307dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensajeA4"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="310dp"
    android:layout_marginTop="307dp"
    android:textSize="8sp" />

<TextView
    android:id="@+id/LblMensajeA5"
    android:layout_width="22dp"
    android:layout_height="15dp"
    android:layout_marginLeft="333dp"
    android:layout_marginTop="307dp"
    android:textSize="8sp" />

<Spinner
    android:id="@+id/spinner2"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="42dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner3"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="62dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner4"
```



```
        android:layout_width="20dp"
        android:layout_height="25dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="81dp"
        android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner5"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="100dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner6"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="119dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner7"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="139dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner8"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="168dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner9"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="187dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner10"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="206dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner11"
    android:layout_width="20dp"
    android:layout_height="25dp"
```



```
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="225dp"
        android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner12"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="244dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinner13"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="263dp"
    android:layout_marginTop="14dp" />

<Spinner
    android:id="@+id/spinnerA0"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="125dp"
    android:layout_marginTop="278dp" />

<Spinner
    android:id="@+id/spinnerA1"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="101dp"
    android:layout_marginTop="278dp" />

<Spinner
    android:id="@+id/spinnerA2"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="77dp"
    android:layout_marginTop="278dp" />

<Spinner
    android:id="@+id/spinnerA3"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="53dp"
    android:layout_marginTop="278dp" />

<Spinner
    android:id="@+id/spinnerA4"
    android:layout_width="20dp"
    android:layout_height="25dp"
```



```
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="30dp"
        android:layout_marginTop="278dp" />

<Spinner
    android:id="@+id/spinnerA5"
    android:layout_width="20dp"
    android:layout_height="25dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="6dp"
    android:layout_marginTop="278dp" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="500dp"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/imageView1"
    android:src="@drawable/pinref" />

<Button
    android:id="@+id/button1"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="180dp"
    android:layout_marginTop="425dp"
    android:text="@string/configip"
    android:textSize="15sp" />

<Button
    android:id="@+id/button2"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="175dp"
    android:layout_marginTop="425dp"
    android:text="@string/save"
    android:textSize="15sp" />

</RelativeLayout>

</FrameLayout>
```





## fragment\_controles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    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" >

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="50dp" >

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

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

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content" >

                    <Switch
                        android:id="@+id/switch1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:onClick="click2"
                        android:text="DO2" />

                    <TextView
                        android:id="@+id/textViewPrueba1"
                        android:layout_width="45dp"
                        android:layout_height="wrap_content"
                        android:layout_marginRight="0dp"
                        android:visibility="invisible" />

                    <CheckBox
                        android:id="@+id/CheckBox1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="0dp"
                        android:enabled="false"
                        android:text="DI2" />

                </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content" >

                <Switch
                    android:id="@+id/switch2"
```



```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click3"
        android:text="DO3" />

<TextView
    android:id="@+id/textViewPrueba2"
    android:layout_width="45dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="0dp"
    android:visibility="invisible" />

<CheckBox
    android:id="@+id/CheckBox2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="DI3" />
</LinearLayout>

<TextView
    android:id="@+id/textView1"
    android:layout_width="40dp"
    android:layout_height="20dp"
    android:layout_gravity="center"
    android:text="PWM"

android:textAppearance="?android:attr/textAppearanceSmall" />

<SeekBar
    android:id="@+id/seekBar1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginRight="25dp"
    android:max="255" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click4"
        android:text="DO4" />

    <TextView
        android:id="@+id/textViewPrueba3"
        android:layout_width="45dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI4" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
```



```
android:layout_height="wrap_content"
android:layout_marginLeft="0dp" >

<Switch
    android:id="@+id/switch4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="click5"
    android:text="DO5" />

<TextView
    android:id="@+id/textViewPrueba4"
    android:layout_width="45dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="0dp"
    android:visibility="invisible" />

<CheckBox
    android:id="@+id/CheckBox4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="DI5" />
</LinearLayout>

<TextView
    android:id="@+id/textView2"
    android:layout_width="40dp"
    android:layout_height="20dp"
    android:layout_gravity="center"
    android:text="PWM"

android:textAppearance="?android:attr/textAppearanceSmall" />

<SeekBar
    android:id="@+id/seekBar2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginRight="25dp"
    android:max="255" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click6"
        android:text="DO6" />

    <TextView
        android:id="@+id/textViewPrueba5"
        android:layout_width="45dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
```



```
        android:text="DI6" />
</LinearLayout>

<TextView
    android:id="@+id/textView3"
    android:layout_width="40dp"
    android:layout_height="20dp"
    android:layout_gravity="center"
    android:text="PWM"

android:textAppearance="?android:attr/textAppearanceSmall" />

<SeekBar
    android:id="@+id/seekBar3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginRight="25dp"
    android:max="255" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click7"
        android:text="DO7" />

    <TextView
        android:id="@+id/textViewPrueba6"
        android:layout_width="45dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI7" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click8"
        android:text="DO8" />

    <TextView
        android:id="@+id/textViewPrueba7"
        android:layout_width="45dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
```



```
        android:id="@+id/CheckBox7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI8" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click9"
        android:text="DO9" />

    <TextView
        android:id="@+id/textViewPrueba8"
        android:layout_width="45dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI9" />
</LinearLayout>

<TextView
    android:id="@+id/textView4"
    android:layout_width="40dp"
    android:layout_height="20dp"
    android:layout_gravity="center"
    android:text="PWM"

    android:textAppearance="?android:attr/textAppearanceSmall" />

<SeekBar
    android:id="@+id/seekBar4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginRight="25dp"
    android:max="255" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click10"
        android:text="D010" />

    <TextView
        android:id="@+id/textViewPrueba9"
        android:layout_width="38dp"
        android:layout_height="wrap_content"
```



```
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

<CheckBox
    android:id="@+id/CheckBox9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="DI10" />
</LinearLayout>

<TextView
    android:id="@+id/textView5"
    android:layout_width="40dp"
    android:layout_height="20dp"
    android:layout_gravity="center"
    android:text="PWM"

android:textAppearance="?android:attr/textAppearanceSmall" />

<SeekBar
    android:id="@+id/seekBar5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginRight="30dp"
    android:layout_weight="1"
    android:max="255" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click11"
        android:text="D011" />

    <TextView
        android:id="@+id/textViewPrueba10"
        android:layout_width="38dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI11" />
</LinearLayout>

<TextView
    android:id="@+id/textView6"
    android:layout_width="40dp"
    android:layout_height="20dp"
    android:layout_gravity="center"
    android:text="PWM"

android:textAppearance="?android:attr/textAppearanceSmall" />

<SeekBar
```



```
        android:id="@+id/seekBar6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="40dp"
        android:layout_marginRight="30dp"
        android:layout_weight="1"
        android:max="255" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click12"
        android:text="D012" />

    <TextView
        android:id="@+id/textViewPrueba11"
        android:layout_width="38dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI12" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Switch
        android:id="@+id/switch12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="click13"
        android:text="D013" />

    <TextView
        android:id="@+id/textViewPrueba12"
        android:layout_width="38dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="0dp"
        android:visibility="invisible" />

    <CheckBox
        android:id="@+id/CheckBox12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="DI13" />

</LinearLayout>
</LinearLayout>
</LinearLayout>
</ScrollView>

<Button
    android:id="@+id/button1"
```



```
android:layout_width="150dp"  
android:layout_height="40dp"  
android:layout_marginLeft="85dp"  
android:layout_marginTop="495dp"  
android:onClick="actualizar"  
android:text="@string/act" />
```

```
</FrameLayout>
```





## fragment\_sensores.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    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" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="10dp"
        android:orientation="vertical" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="30dp" >

            <TextView
                android:id="@+id/textView1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="5dp"
                android:text="A0"
                android:textSize="20sp" />

            <ProgressBar
                android:id="@+id/progressBar1"
                style="?android:attr/progressBarStyleHorizontal"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:max="1024"
                android:visibility="visible" />

        </LinearLayout>

        <TextView
            android:id="@+id/textView7"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="0" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="30dp"
            android:layout_marginTop="20dp" >

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="5dp"
                android:text="A1"
                android:textSize="20sp" />

            <ProgressBar
                android:id="@+id/progressBar2"
                style="?android:attr/progressBarStyleHorizontal"
                android:layout_width="match_parent"
```



```
        android:layout_height="match_parent"
        android:max="1024"
        android:visibility="visible" />

</LinearLayout>

<TextView
    android:id="@+id/textView8"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="0" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:layout_marginTop="20dp" >

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="A2"
        android:textSize="20sp" />

    <ProgressBar
        android:id="@+id/progressBar3"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:max="3072"
        android:visibility="visible" />

</LinearLayout>

<TextView
    android:id="@+id/textView9"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="0" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:layout_marginTop="20dp" >

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="A3"
        android:textSize="20sp" />

    <ProgressBar
        android:id="@+id/progressBar4"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:max="4096"
        android:visibility="visible" />

</LinearLayout>
```



```
<TextView
    android:id="@+id/textView10"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="0" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:layout_marginTop="20dp" >

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="A4"
        android:textSize="20sp" />

    <ProgressBar
        android:id="@+id/progressBar5"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:max="5120"
        android:visibility="visible" />

</LinearLayout>

<TextView
    android:id="@+id/textView11"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="0" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:layout_marginTop="20dp" >

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="A5"
        android:textSize="20sp" />

    <ProgressBar
        android:id="@+id/progressBar6"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:max="6144"
        android:visibility="visible" />

</LinearLayout>

<TextView
    android:id="@+id/textView12"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
```



```
        android:text="0" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView13"
        android:layout_width="200dp"
        android:layout_height="30dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="40dp"
        android:text="@string/sens"
        android:textSize="18sp" />

    <Button
        android:id="@+id/button1"
        android:layout_width="100dp"
        android:layout_height="40dp"
        android:layout_marginTop="40dp"
        android:text="ON" />

</LinearLayout>

</LinearLayout>

</FrameLayout>
```



## fragment\_splash\_activity.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:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/White"
    android:gravity="center_vertical"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="50dp"
        android:layout_marginTop="100dp"
        android:src="@drawable/ic_splash" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="0dp"
        android:src="@drawable/nombre" />

</LinearLayout>
```



## header.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="vertical" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="250dp"
        android:layout_height="125dp"
        android:background="@color/White"
        android:src="@drawable/uva_eii" />

</LinearLayout>
```



## tutorial.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="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:text="@string/exAjustes1" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="300dp"
        android:layout_height="40dp"
        android:layout_marginLeft="15dp"
        android:src="@drawable/pin13"
        android:visibility="visible" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="60dp"
        android:layout_marginTop="15dp"
        android:clickable="false"
        android:text="@string/save" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="@string/exAjustes2" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp" >

        <ImageButton
            android:id="@+id/imageButton1"
            android:layout_width="70dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="250dp"
            android:src="@drawable/ic_action_next_item" />

    </LinearLayout>
</LinearLayout>
```



## main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/action_tutorial"
        android:icon="@drawable/ic_action_help"
        android:title="@string/action_tutorial"
        android:showAsAction="ifRoom|withText" />
  <item android:id="@+id/ic_action_copy"
        android:icon="@drawable/ic_action_copy"
        android:title="@string/ic_action_copy"
        android:showAsAction="ifRoom|withText" />
</menu>
```





## colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <color name="White">#FFFFFF</color>
    <color name="Ivory">#FFFFF0</color>
    <color name="LightYellow">#FFFFE0</color>
    <color name="Yellow">#FFFF00</color>
    <color name="Snow">#FFFAFA</color>
    <color name="FloralWhite">#FFFAF0</color>
    <color name="LemonChiffon">#FFFACD</color>
    <color name="Cornsilk">#FFF8DC</color>
    <color name="Seashell">#FFF5EE</color>
    <color name="LavenderBlush">#FFF0F5</color>
    <color name="PapayaWhip">#FFEFD5</color>
    <color name="BlanchedAlmond">#FFEBCD</color>
    <color name="MistyRose">#FFE4E1</color>
    <color name="Bisque">#FFE4C4</color>
    <color name="Moccasin">#FFE4B5</color>
    <color name="NavajoWhite">#FFDEAD</color>
    <color name="PeachPuff">#FFDAB9</color>
    <color name="Gold">#FFD700</color>
    <color name="Pink">#FFC0CB</color>
    <color name="LightPink">#FFB6C1</color>
    <color name="Orange">#FFA500</color>
    <color name="LightSalmon">#FFA07A</color>
    <color name="DarkOrange">#FF8C00</color>
    <color name="Coral">#FF7F50</color>
    <color name="HotPink">#FF69B4</color>
    <color name="Tomato">#FF6347</color>
    <color name="OrangeRed">#FF4500</color>
    <color name="DeepPink">#FF1493</color>
    <color name="Fuchsia">#FF00FF</color>
    <color name="Magenta">#FF00FF</color>
    <color name="Red">#FF0000</color>
    <color name="OldLace">#FDF5E6</color>
    <color name="LightGoldenrodYellow">#FAFAD2</color>
    <color name="Linen">#FAF0E6</color>
    <color name="AntiqueWhite">#FAEBD7</color>
    <color name="Salmon">#FA8072</color>
    <color name="GhostWhite">#F8F8FF</color>
    <color name="MintCream">#F5FFFA</color>
    <color name="WhiteSmoke">#F5F5F5</color>
    <color name="Beige">#F5F5DC</color>
    <color name="Wheat">#F5DEB3</color>
    <color name="SandyBrown">#F4A460</color>
    <color name="Azure">#F0FFFF</color>
    <color name="Honeydew">#F0FFF0</color>
    <color name="AliceBlue">#F0F8FF</color>
    <color name="Khaki">#F0E68C</color>
    <color name="LightCoral">#F08080</color>
    <color name="PaleGoldenrod">#EEE8AA</color>
    <color name="Violet">#EE82EE</color>
    <color name="DarkSalmon">#E9967A</color>
    <color name="Lavender">#E6E6FA</color>
    <color name="LightCyan">#E0FFFF</color>
    <color name="BurlyWood">#DEB887</color>
    <color name="Plum">#DDA0DD</color>
    <color name="Gainsboro">#DCDCDC</color>
    <color name="Crimson">#DC143C</color>
    <color name="PaleVioletRed">#DB7093</color>
    <color name="Goldenrod">#DAA520</color>
```



```
<color name="Orchid">#DA70D6</color>
<color name="Thistle">#D8BFD8</color>
<color name="LightGrey">#D3D3D3</color>
<color name="Tan">#D2B48C</color>
<color name="Chocolate">#D2691E</color>
<color name="Peru">#CD853F</color>
<color name="IndianRed">#CD5C5C</color>
<color name="MediumVioletRed">#C71585</color>
<color name="Silver">#C0C0C0</color>
<color name="DarkKhaki">#BDB76B</color>
<color name="RosyBrown">#BC8F8F</color>
<color name="MediumOrchid">#BA55D3</color>
<color name="DarkGoldenrod">#B8860B</color>
<color name="FireBrick">#B22222</color>
<color name="PowderBlue">#B0E0E6</color>
<color name="LightSteelBlue">#B0C4DE</color>
<color name="PaleTurquoise">#AFEEEE</color>
<color name="GreenYellow">#ADFF2F</color>
<color name="LightBlue">#ADD8E6</color>
<color name="DarkGray">#A9A9A9</color>
<color name="Brown">#A52A2A</color>
<color name="Sienna">#A0522D</color>
<color name="YellowGreen">#9ACD32</color>
<color name="DarkOrchid">#9932CC</color>
<color name="PaleGreen">#98FB98</color>
<color name="DarkViolet">#9400D3</color>
<color name="MediumPurple">#9370DB</color>
<color name="LightGreen">#90EE90</color>
<color name="DarkSeaGreen">#8FBC8F</color>
<color name="SaddleBrown">#8B4513</color>
<color name="DarkMagenta">#8B008B</color>
<color name="DarkRed">#8B0000</color>
<color name="BlueViolet">#8A2BE2</color>
<color name="LightSkyBlue">#87CEFA</color>
<color name="SkyBlue">#87CEEB</color>
<color name="Gray">#808080</color>
<color name="Olive">#808000</color>
<color name="Purple">#800080</color>
<color name="Maroon">#800000</color>
<color name="Aquamarine">#7FFFD4</color>
<color name="Chartreuse">#7FFF00</color>
<color name="LawnGreen">#7CFC00</color>
<color name="MediumSlateBlue">#7B68EE</color>
<color name="LightSlateGray">#778899</color>
<color name="SlateGray">#708090</color>
<color name="OliveDrab">#6B8E23</color>
<color name="SlateBlue">#6A5ACD</color>
<color name="DimGray">#696969</color>
<color name="MediumAquamarine">#66CDAA</color>
<color name="CornflowerBlue">#6495ED</color>
<color name="CadetBlue">#5F9EA0</color>
<color name="DarkOliveGreen">#556B2F</color>
<color name="Indigo">#4B0082</color>
<color name="MediumTurquoise">#48D1CC</color>
<color name="DarkSlateBlue">#483D8B</color>
<color name="SteelBlue">#4682B4</color>
<color name="RoyalBlue">#4169E1</color>
<color name="Turquoise">#40E0D0</color>
<color name="MediumSeaGreen">#3CB371</color>
<color name="LimeGreen">#32CD32</color>
<color name="DarkSlateGray">#2F4F4F</color>
<color name="SeaGreen">#2E8B57</color>
<color name="ForestGreen">#228B22</color>
<color name="LightSeaGreen">#20B2AA</color>
<color name="DodgerBlue">#1E90FF</color>
```



```
<color name="MidnightBlue">#191970</color>
<color name="Aqua">#00FFFF</color>
<color name="Cyan">#00FFFF</color>
<color name="SpringGreen">#00FF7F</color>
<color name="Lime">#00FF00</color>
<color name="MediumSpringGreen">#00FA9A</color>
<color name="DarkTurquoise">#00CED1</color>
<color name="DeepSkyBlue">#00BFFF</color>
<color name="DarkCyan">#008B8B</color>
<color name="Teal">#008080</color>
<color name="Green">#008000</color>
<color name="DarkGreen">#006400</color>
<color name="Blue">#0000FF</color>
<color name="MediumBlue">#0000CD</color>
<color name="DarkBlue">#00008B</color>
<color name="Navy">#000080</color>
<color name="Black">#000000</color>
</resources>
```



## dimens.xml

```
<resources>  
    <!-- Default screen margins, per the Android Design guidelines. -->  
    <dimen name="activity_horizontal_margin">16dp</dimen>  
    <dimen name="activity_vertical_margin">16dp</dimen>  
</resources>
```



## strings.xml

```
<resources>
  <string name="app_name">Arduino Yun</string>
  <string-array name="opciones">
    <item>Ajustes</item>
    <item>Controles</item>
    <item>Sensores</item>
  </string-array>

  <string name="drawer_open">Open navigation drawer</string>
  <string name="drawer_close">Close navigation drawer</string>
  <string name="action_tutorial">Ayuda</string>
  <string name="ic_action_copy">Pines.txt</string>
  <string name="app_not_available">Sorry, there's no web browser
available</string>

  <string name="configip">CONECTAR</string>
  <string name="dirip">Direccion IP</string>
  <string name="portip">Puerto</string>
  <string name="inicio">Alberto Frades &#169;</string>
  <string name="act">Actualizar</string>
  <string name="btnaj">Ajustes</string>
  <string name="btnco">Controles</string>
  <string name="btnse">Sensores</string>
  <string name="save">GUARDAR AJUSTES</string>
  <string name="cargar">CARGAR AJUSTES</string>
  <string name="actsens">LECTURA DE SENSORES</string>
  <string name="sig">SIGUIENTE</string>
  <string name="sens">LECTURA SENSORES</string>
  <string name="notificaciones">NOTIFICACIONES</string>

  <string name="exAjustes1">\tEn Ajustes, se configurarán los pines
necesarios
  para manejar Arduino.\n
  En cada pin se mostrará un desplegable con las opciones
disponibles.\n\n
  Una vez terminado se deberán guardar dichos ajustes. </string>

  <string name="exAjustes2"> Simbologia en los pines:\n\n
\t&#60; &#60; &#60;\t\t Recupera el ultimo valor guardado.\n
\tN/C\t\t Pin NO CONECTADO\n
\tDI\t\t\t\t Digital Input\n
\tDO\t\t\t\t Digital Output\n
\tPWM\t\t\t\t Analog Value</string>

  <string name="exControles1"> \n
  Seleccionando Controles en el menu desplegable de la izquierda,
se
  accederá al control de la placa arduino siguiendo la
configuracion establecida.\n\n
  Para actualizar el estado, pulsar el boton inferior de la
pantalla.</string>

  <string name="exControles2"> \n
  En el apartado Sensores, se muestran las lecturas de los pines
analógicos.\n\n
  Para realizar un seguimiento continuo de los sensores, activar
el boton inferior.</string>

  <string name="exactionbar1"> \n
```



En el menu superior, a la derecha tenemos un icono.\n\n Pulsando sobre él, nos abrirá el directorio donde se encuentra la configuracion guardada en ficheros .txt de los pines de la placa Arduino.\n Desde ahí también podremos modificarlos, renombrandolos siempre con la nomenclatura explicada anteriormente.</string>

</resources>



## AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.navigationdrawerexample"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk android:minSdkVersion="14" android:targetSdkVersion="21" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:icon="@drawable/ic_launcher_ofi"
        android:theme="@android:style/Theme.Holo.Light.DarkActionBar">

        <activity
            android:name=".FragmentSplashActivity"
            android:screenOrientation="portrait"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>

        <activity
            android:name=".MainActivity"
            android:screenOrientation="portrait">
        </activity>

        <activity
            android:name=".Tutorial"
            android:screenOrientation="portrait">
        </activity>

    </application>
</manifest>
```



## Bridge.ino

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>

// Escucha el por defecto el puerto 5555.
//WebServer recibirá en ese puerto todas las peticiones enviadas
//a Arduino Yun.
YunServer server;

void setup() {
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
  Bridge.begin();
  digitalWrite(13, HIGH);

  // Escucha la conexión de entrada solo desde un host local
  server.listenOnLocalhost();
  server.begin();
}

void loop() {
  YunClient client = server.accept();

  // si hay un nuevo cliente procesa la petición
  if (client) {
    process(client);

    // Cierra la conexión.
    client.stop();
  }

  delay(50); // Retardo de 50ms. Tiempo entre escucha de peticiones
}

void process(YunClient client) {
  // lee el comando
  String command = client.readStringUntil('/');

  // si es digital
  if (command == "digital") {
    digitalCommand(client);
  }

  // si el comando es analogico
  if (command == "analog") {
    analogCommand(client);
  }

  // si el comando es modo
  if (command == "mode") {
    modeCommand(client);
  }
}

void digitalCommand(YunClient client) {
  int pin, value;

  // lee el pin
  pin = client.parseInt();
```





```
// Si el carácter es '/' significa que tenemos una URL
// con un valor como "/digital/13/1"
if (client.read() == '/') {
    value = client.parseInt();
    digitalWrite(pin, value);
}
else {
    value = digitalRead(pin);
}

client.print(F("Pin D"));
client.print(pin);
client.print(F(" set to "));
client.println(value);

String key = "D";
key += pin;
Bridge.put(key, String(value));
}

void analogCommand(YunClient client) {
    int pin, value;

    // lee el pin
    pin = client.parseInt();

    // Si el carácter es '/' significa que tenemos una URL
    // con un valor como "/analog/5/120"

    if (client.read() == '/') {
        // lee el valor y ejecuta el comando
        value = client.parseInt();
        analogWrite(pin, value);

        client.print(F("Pin D"));
        client.print(pin);
        client.print(F(" set to analog "));
        client.println(value);

        String key = "D";
        key += pin;
        Bridge.put(key, String(value));
    }
    else {
        // lee el valor del pin analogico
        value = analogRead(pin);

        client.print(F("Pin A"));
        client.print(pin);
        client.print(F(" reads analog "));
        client.println(value);

        String key = "A";
        key += pin;
        Bridge.put(key, String(value));
    }
}

void modeCommand(YunClient client) {
    int pin;

    // lee el pin
    pin = client.parseInt();
```



```
//Si el siguiente carácter no es '/' significa que
// tenemos un error y la URL esta mal declarada
if (client.read() != '/') {
    client.println(F("error"));
    return;
}

String mode = client.readStringUntil('\r');

if (mode == "input") {
    pinMode(pin, INPUT);
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" configurado as INPUT!"));
    return;
}

if (mode == "output") {
    pinMode(pin, OUTPUT);
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" configurado as OUTPUT!"));
    return;
}

client.print(F("error"));
client.print(mode);
}
```