|  |  |
| --- | --- |
| Screen | Spinner |
|  | En archivo **res/values/strings.xlm**  <**resources**>  <**string name="app\_name"**>ServiceMonitorBattery</**string**>  <**string name="action\_settings"**>Settings</**string**>   <**string-array name="Comandos"**>  <**item**>0 Test</**item**>   <**item**>1 Turn On</**item**>  <**item**>2 Turn Off</**item**>  <**item**>3 Get Humidity</**item**>  <**item**>4 Get Voltage</**item**>  <**item**>5 Watered Plants</**item**>  </**string-array**>  </**resources**> |

**ACTIVITY MAIN**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.example.tdal6392.bluetoothwater.MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 </**android.support.design.widget.AppBarLayout**>  
 <**include layout="@layout/content\_main"** />  
</**android.support.design.widget.CoordinatorLayout**>

**CONTENT MAIN**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:id="@+id/parentView"  
 android:clickable="true"  
 android:focusableInTouchMode="true"  
 android:orientation="vertical"  
  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.example.tdal6392.bluetoothwater.MainActivity"  
 tools:showIn="@layout/activity\_main"  
 android:background="#e3f9fa"  
 android:focusable="true"**>  
  
 <**TextView  
 android:text="TextView"  
 android:layout\_width="wrap\_content"  
 android:textSize="17sp"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/estado"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="21dp"  
 android:layout\_marginTop="37dp"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Start Service"  
 android:onClick="Iniciar\_Parar"  
 android:id="@+id/btnStart"  
 android:layout\_below="@+id/estado"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="35dp"** />  
  
 <**Button  
 android:layout\_width="100dip"  
 android:layout\_height="wrap\_content"  
 android:text="Execute"  
 android:id="@+id/btnSend"  
 android:onClick="Enviar"  
 android:layout\_marginTop="10dp"  
 android:layout\_below="@+id/btnStart"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignEnd="@+id/btnStart"** />  
  
 <**Spinner  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/spinner2"  
 android:layout\_alignBottom="@+id/btnSend"  
 android:layout\_centerHorizontal="true"** />

</**RelativeLayout**>

MANIFEST

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.tdal6392.bluetoothwater"**>  
  
<**uses-permission android:name="android.permission.BLUETOOTH"** />  
<**uses-permission android:name="android.permission.BLUETOOTH\_ADMIN"** />  
  
<**uses-permission android:name="android.permission.READ\_PHONE\_STATE"**/>  
  
<**uses-permission android:name="android.permission.RECEIVE\_SMS"** />  
<**uses-permission android:name="android.permission.READ\_SMS"** />  
<**uses-permission android:name="android.permission.SEND\_SMS"** />  
  
<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** />  
<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** />

<**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
  
 <**activity  
 android:name=".MainActivity"  
 android:screenOrientation="portrait"  
 android:label="Monitor Arduino"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
  
 <**service android:name=".ServiceBattery"** />  
  
 </**application**>  
</**manifest**>

**MAIN**

**package** com.example.tdal6392.bluetoothwater;  
**import** android.app.ActivityManager;  
**import** android.content.BroadcastReceiver;  
**import** android.content.ComponentName;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.content.ServiceConnection;  
**import** android.graphics.Color;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.os.CountDownTimer;  
**import** android.os.IBinder;  
**import** android.support.v4.content.LocalBroadcastManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.Spinner;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **final** String **ServiceName** = **"com.example.tdal6392.bluetoothwater.ServiceBattery"**;  
 ServiceBattery **mService**;  
 **boolean mBound** = **false**;  
 BroadcastReceiver **receiver**;  
  
 Button **btStart**, **btSend**;  
  
 *// VARIABLE TO BIND TO SERVICE  
 //============================  
 /\*\* Defines callbacks for service binding, passed to bindService() \*/* **private** ServiceConnection **mConnection** = **new** ServiceConnection() {  
  
 @Override  
 **public void** onServiceConnected(ComponentName className,  
 IBinder service) {  
 *// We've bound to ServiceBattery, cast the IBinder and get ServiceBattery instance* ServiceBattery.LocalBinder binder = (ServiceBattery.LocalBinder) service;  
 **mService** = binder.getService();  
 **mBound** = **true**;  
 }  
  
 @Override  
 **public void** onServiceDisconnected(ComponentName arg0) {  
 **mBound** = **false**;  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 TextView estado = (TextView) findViewById(R.id.***estado***);  
 estado.setText(**""**);  
  
 Spinner listado = (Spinner) findViewById(R.id.***spinner2***);  
 listado.setBackgroundColor(Color.***WHITE***);  
  
 Spinner spinner = (Spinner) findViewById(R.id.***spinner2***);  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(  
 **this**, R.array.***Comandos***, android.R.layout.***simple\_spinner\_item***);  
  
 adapter.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
 spinner.setAdapter(adapter);  
  
 **btStart** = (Button) findViewById(R.id.***btnStart***);  
 **btSend** = (Button) findViewById(R.id.***btnSend***);  
  
 IntentFilter filter = **new** IntentFilter();  
 filter.addAction(**"com.hello.action"**);  
  
  
 **receiver** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String retorno = intent.getStringExtra(ServiceBattery.***SRV\_MESSAGE***);  
 Start\_Or\_Stop(retorno);  
 }  
 };  
  
 **if** (isServiceRunning())  
 {  
 **btStart**.setText(**"STOP SERVICE"**);  
 **btSend**.setEnabled(**true**);  
  
 *// BIND SERVICE* Intent intent = **new** Intent(**this**, ServiceBattery.**class**);  
 bindService(intent, **mConnection**, Context.***BIND\_AUTO\_CREATE***);  
 }  
 **else** {  
 **btStart**.setText(**"START SERVICE"**);  
 **btSend**.setEnabled(**false**);  
 }  
 }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 LocalBroadcastManager.*getInstance*(**this**).registerReceiver((**receiver**),  
 **new** IntentFilter(ServiceBattery.***SRV\_RESULT***)  
 );  
 }  
  
 @Override  
 **protected void** onStop() {  
 LocalBroadcastManager.*getInstance*(**this**).unregisterReceiver(**receiver**);  
 **super**.onStop();  
 }  
  
 *// BUTTON CLICK* **public void** Iniciar\_Parar(View v) {  
 *//if (btStart.getText().toString().equals("START SERVICE"))* **if** (!isServiceRunning())  
 ServStart();  
 **else** ServStop();  
 }  
  
 *// INICIAR SERVICIO  
 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** **public void** ServStart()  
 {  
 **try** {  
 Class aClass = Class.*forName*(**ServiceName**);  
 Intent I2 = **new** Intent(getApplicationContext(), aClass);  
 *// I2.putExtra("NumFrom", NumFrom);  
 // I2.putExtra("EMail", Email2);* startService(I2);  
 }  
 **catch**( Exception e) {  
 Toast.*makeText*(getApplicationContext(),**"Error "**+ e.toString(),Toast.***LENGTH\_LONG***).show();  
 }  
 }  
  
 **public void** Start\_Or\_Stop(String errMsg2)  
 {  
 *//ShowMsg("Code "+errMsg2);* TextView estado = (TextView) findViewById(R.id.***estado***);  
  
 **int** errMsg = 0;  
 **try** {  
 errMsg = Integer.*parseInt*(errMsg2);  
 }  
 **catch** (Exception ss)  
 {  
 *// texto, no codigo* **if** (errMsg2.length() ==1) errMsg2 = **"Code "**+ errMsg2;  
 estado.setText(errMsg2);  
 **return**;  
 }  
  
 **if** (errMsg<3) {  
 estado.setText(**"Connected"**);  
 **btStart**.setText(**"STOP SERVICE"**);  
 **btSend**.setEnabled(**true**);  
  
 *// BIND SERVICE* Intent intent = **new** Intent(**this**, ServiceBattery.**class**);  
 bindService(intent, **mConnection**, Context.***BIND\_AUTO\_CREATE***);  
 **return**;  
 }  
  
 **if** (errMsg == 3) estado.setText(**"Celular sin Bluetooth"**);  
 **if** (errMsg == 4) estado.setText(**"Need to enable Bluetooth"**);  
 **if** (errMsg == 5) estado.setText(**"HC-06 not found"**);  
 **if** (errMsg == 6) estado.setText(**"Not able to connect to HC-06"**);  
 **if** (errMsg == 7) estado.setText(**"HC-06 not found"**);  
  
 ServStop();  
 }  
  
 **public void** ServStop() {  
 **try** {  
 Class aClass = Class.*forName*(**ServiceName**);  
 Intent I1 = **new** Intent(getApplicationContext(), aClass);  
 stopService(I1);  
 }  
 **catch** (Exception es)  
 {  
 Toast.*makeText*(getApplicationContext(),**"Error "**+ es.toString(),Toast.***LENGTH\_LONG***).show();  
 **return**;  
 }  
 **btSend**.setEnabled(**false**);  
 **btStart**.setText(**"START SERVICE"**);  
 }  
  
  
 **public void** Enviar(View v)  
 {  
 Spinner spinner = (Spinner)findViewById(R.id.***spinner2***);  
 String text = spinner.getSelectedItem().toString();  
 String mensaje = Character.*toString*( text.charAt(0) );  
 **mService**.SendData(mensaje);  
 }  
  
  
 *// FUNCIONES  
 //==========* **public void** Wait(**int** tiempo)  
 {  
 **new** CountDownTimer(tiempo, 1000) {  
 **public void** onTick(**long** millisUntilFinished) { }  
 **public void** onFinish() { }  
  
 }.start();  
 }  
  
 **private boolean** isServiceRunning() {  
 ActivityManager manager = (ActivityManager) getSystemService(***ACTIVITY\_SERVICE***);  
 **for** (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.***MAX\_VALUE***)) {  
 **if** (**ServiceName**.equals(service.**service**.getClassName())) {  
 **return true**;  
 }  
 }  
 **return false**;  
 }  
  
 **public void** ShowMsg(String msg)  
 {  
 Toast.*makeText*(getApplicationContext(), msg, Toast.***LENGTH\_LONG***).show();  
 }  
  
 **public void** ENVIAR\_MAIL(String data)  
 {  
 **try** {  
 String email2 = **"Levitico17@gmail.com"**;  
 Intent email = **new** Intent(Intent.***ACTION\_SENDTO***);  
 email.setData(Uri.*parse*(**"mailto:"**+email2));  
 email.setType(**"text/plain"**);  
  
 email.putExtra(Intent.***EXTRA\_EMAIL***, **new** String[] { email2 });  
 email.putExtra(Intent.***EXTRA\_SUBJECT***, **"Arduino"**);  
 email.putExtra(Intent.***EXTRA\_TEXT***, data);  
 email.addFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***); *// when returns to your app, your app is displayed, instead of the email app.* email.addFlags(Intent.***FLAG\_FROM\_BACKGROUND***);  
  
 startActivity(Intent.*createChooser*(email, **"Send mail..."**));  
 }  
 **catch**(Exception es){  
 String ss = es.toString();  
 ss = ss.replace(**"android.content."**,**""**);  
 ShowMsg(ss);  
 }  
 }  
}

**SERVICIO**

**package** com.example.tdal6392.bluetoothwater;  
**import** android.app.Notification;  
**import** android.app.Service;  
**import** android.bluetooth.BluetoothAdapter;  
**import** android.bluetooth.BluetoothDevice;  
**import** android.bluetooth.BluetoothSocket;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.os.BatteryManager;  
**import** android.os.Binder;  
**import** android.os.IBinder;  
**import** android.provider.Telephony;  
**import** android.support.v4.app.NotificationCompat;  
**import** android.support.v4.content.LocalBroadcastManager;  
**import** android.telephony.SmsManager;  
**import** android.telephony.SmsMessage;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.util.Set;  
**import** java.util.UUID;  
**import** android.os.Handler;  
  
**public class** ServiceBattery **extends** Service {  
  
 **int errMsg** = 0;  
 String **NumFrom** = **""**;  
 BluetoothSocket **mmSocket** = **null**;  
 LocalBroadcastManager **lb**;  
 **int nivel1** = 0;  
 **boolean Bat1** = **false**;  
  
 **static final public** String ***SRV\_RESULT*** = **"com.example.tdal6392.bluetoothwater.REQUEST\_PROCESSED"**;  
 **static final public** String ***SRV\_MESSAGE*** = **"com.example.tdal6392.bluetoothwater.MSG"**;  
 *//=============* **private final** IBinder **mBinder** = **new** LocalBinder();  
 **public class** LocalBinder **extends** Binder {  
 ServiceBattery getService() {  
 **return** ServiceBattery.**this**;  
 }  
 }  
  
  
 **private final** BroadcastReceiver **receiver** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 **if** (action.equals(**"android.provider.Telephony.SMS\_RECEIVED"**)) {  
 **for** (SmsMessage sms1 : Telephony.Sms.Intents.*getMessagesFromIntent*(intent)) {  
  
 String msgBody = sms1.getMessageBody();  
  
 **if** (msgBody.length() == 1)  
 {  
 ENVIAR\_UI(msgBody);  
 **NumFrom** = sms1.getOriginatingAddress();  
 SendData(msgBody);  
 }  
  
 **if** (msgBody==**"Bat0"** || msgBody==**"bat0"**)  
 {  
 **NumFrom** = sms1.getOriginatingAddress();  
 **int** level = intent.getIntExtra(BatteryManager.***EXTRA\_LEVEL***, 0);  
 String msg = **"Battery level "** + String.*valueOf*(level)+ **"%"**;  
 ENVIAR\_SMS(msg);  
 }  
  
 **if** (msgBody==**"Bat1"** || msgBody==**"bat1"**) **Bat1** = **true**;  
 **if** (msgBody==**"Bat2"** || msgBody==**"bat2"**) **Bat1** = **false**;  
  
 } *//FOR* } *// IF* **if** (action.equals(**"android.intent.action.BATTERY\_CHANGED"**)) {  
  
   
 **int** level = intent.getIntExtra(BatteryManager.***EXTRA\_LEVEL***, 0);  
 String msg = **"Battery level "** + String.*valueOf*(level) + **"%"**;  
  
 **if** (**nivel1** == 0) {  
 **nivel1** = level - 4;  
 **if** (**NumFrom** == **""**) **NumFrom** = **"70020015"**;  
 **return**;  
 }  
  
 **if** (**nivel1** > level && **Bat1**) {  
 **nivel1** = **nivel1** -5;  
 ENVIAR\_SMS(msg);  
 }  
  
 *//if (true) return;* **if** (level > 95) SendData(**"A"**); *// high, stop charging* **if** (level < 20) {  
 SendData(**"B"**); *// low, start charging* ENVIAR\_SMS(**"Charging"**);  
 }  
 } *// IF* } *// onReceive* };  
  
  
 @Override  
 **public** IBinder onBind(Intent intent) {  
 *// Bundle extras = intent.getExtras();  
 // NumFrom = (String) extras.get("NumFrom");  
 // EMail = (String) extras.get("EMail");* **return mBinder**;  
 }  
  
 *// @Override* **public void** onCreate() {  
 **super**.onCreate();  
  
  
 *// REGISTER BROADCAST RECEIVER* IntentFilter filter1 = **new** IntentFilter();  
 filter1.addAction(**"android.provider.Telephony.SMS\_RECEIVED"**);  
 filter1.addAction(android.telephony.TelephonyManager.***ACTION\_PHONE\_STATE\_CHANGED***);  
 filter1.addAction(Intent.***ACTION\_BATTERY\_CHANGED***);  
 registerReceiver(**receiver**, filter1);  
  
 GetSocketBT();  
 *//============* **lb** = LocalBroadcastManager.*getInstance*(**this**);  
 String ErrMsg = String.*valueOf*(**errMsg**);  
 ENVIAR\_UI(ErrMsg);  
 }  
  
  
  
 @Override  
 **public int** onStartCommand(Intent intent, **int** flags, **int** startId) {  
 *// Notification notification = new NotificationCompat.Builder(this)  
 // .setOngoing(false)  
 // .setSmallIcon(android.R.color.transparent)  
 // //.setSmallIcon(R.drawable.picture)  
 // .build();  
  
 // start foreground to avoid destroy  
 //startForeground(101, notification);  
  
 //START\_STICKY will re-create the service if it is killed.* **return *START\_STICKY***;  
 }  
  
  
 @Override  
 **public void** onDestroy() {  
 **try** {  
 **mmSocket**.close();  
 }  
 **catch** (Exception e) {}  
 }  
  
  
 *// OPEN BLUETOOTH SOCKET FOR ARDUINO  
 //==================================* **public void** GetSocketBT() {  
  
 **mmSocket** = **null**;  
 **errMsg** = 0;  
  
 BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.*getDefaultAdapter*();  
  
 **if** (mBluetoothAdapter == **null**) {  
 **errMsg** = 3;  
 **return**;  
 }  
  
 **if** (!mBluetoothAdapter.isEnabled()) {  
 **errMsg**= 4;  
 **return**;  
 }  
  
 BluetoothDevice mmDevice = **null**;  
 Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();  
 **if** (pairedDevices.size() > 0) {  
 **for** (BluetoothDevice device : pairedDevices) {  
 **if** (device.getName().equals(**"HC-06"**)) {  
 mmDevice = device;  
 **break**;  
 }  
 }  
 }  
  
 **if** (mmDevice == **null**) {  
 **errMsg**= 5;  
 **return**;  
 }  
  
 **try** {  
 UUID MY\_UUID = UUID.*fromString*(**"00001101-0000-1000-8000-00805F9B34FB"**);  
 **mmSocket** = mmDevice.createRfcommSocketToServiceRecord(MY\_UUID);  
  
 mBluetoothAdapter.*getDefaultAdapter*().cancelDiscovery();  
 **mmSocket**.connect();  
  
 } **catch** (Exception e) {  
 **errMsg**= 6;  
 **return**;  
 }  
  
 **if** (**mmSocket** == **null**)  
 {  
 **errMsg**= 7;  
 **return**;  
 }  
  
 *// Turn On Light 3 seconds* SendData(**"0"**);  
 ListenForData();  
 }  
  
  
 *// SEND DATA TO ARDUINO  
 // ====================* **public void** SendData(String msg) {  
 **try** {  
 **mmSocket**.getOutputStream().write(msg.toString().getBytes());  
 } **catch** (Exception e) {  
 ENVIAR\_SMS(**"Data not sent"** + e.toString());  
 ENVIAR\_UI(**"Data not sent"**);  
 }  
 }  
  
  
 *// LISTEN TO INCOMING DATA FROM ARDUINO  
 //=====================================* **public void** ListenForData()  
 {  
 **final** Handler handler = **new** Handler();  
 **final byte** delimiter = 42; *// 42="\*" 10 = new line ascii* Thread workerThread = **new** Thread(**new** Runnable() {  
 **public void** run() {  
  
 InputStream mmInputStream = **null**;  
  
 **try** {  
 mmInputStream = **mmSocket**.getInputStream();  
 }  
 **catch** (Exception e) {}  
  
 **byte**[] readBuffer = **new byte**[1024];  
 **int** NumChars = 0;  
 **boolean** stopWorker= **false**; *// STOPS IF CATCH, ERROR  
  
 // WHILE TRUE* **while**(!Thread.*currentThread*().isInterrupted() && !stopWorker) {  
 **try** {  
  
 **int** bytesAvailable = mmInputStream.available();  
 **if**(bytesAvailable > 0) {  
 **byte**[] packetBytes = **new byte**[bytesAvailable];  
 mmInputStream.read(packetBytes);  
 **for**(**int** i=0;i<bytesAvailable;i++) {  
 **byte** b = packetBytes[i];  
  
 **if**(b == delimiter) {  
  
 *// COPY readBuffer -> encodedBytes  
 // encodedBytes.length = NumChars;* **byte**[] encodedBytes = **new byte**[NumChars];  
 System.*arraycopy*(readBuffer, 0, encodedBytes, 0, encodedBytes.**length**);  
  
 **final** String data = **new** String(encodedBytes, **"US-ASCII"**);  
 NumChars = 0;  
  
 handler.post(**new** Runnable() {  
 **public void** run() {  
  
 ENVIAR\_SMS(data);  
 ENVIAR\_UI(data);  
 }  
 });  
 }  
 **else** {  
 readBuffer[NumChars++] = b;  
 }  
 }  
 }  
 }  
 **catch** (IOException ex) {  
 ENVIAR\_SMS(**"Catch "**+ex.toString());  
 stopWorker = **true**;  
 }  
 }  
 }  
 });  
  
 workerThread.start();  
 }  
  
 **public void** ENVIAR\_SMS(String data) {  
 **try** {  
 String mensaje = **"2- "** + data;  
 **if** (**NumFrom** != **""**) {  
 SmsManager sms = SmsManager.*getDefault*();  
 sms.sendTextMessage(**NumFrom**, **null**, mensaje, **null**, **null**);  
 }  
 }  
 **catch**(Exception es)  
 {  
 ENVIAR\_UI(es.toString());  
 }  
 }  
  
 **public void** ENVIAR\_UI(String msg)  
 {  
 **try** {  
 Intent intent = **new** Intent(***SRV\_RESULT***);  
 intent.putExtra(***SRV\_MESSAGE***, msg);  
 **lb**.sendBroadcast(intent);  
 }  
 **catch**(Exception es)  
 {  
  
 }  
 }  
}