ANEXO B

Métodos programáticos

SÍNTESE
Neste anexo apresenta-se, em parte, os métodos programáticos usados nas atividades principais. Trata-se de um pequena parte da programação mas que representa bem o que é programar em android

A37022
Utilização de Plataformas de Fonte Aberta no Controlo de Condição

B-1. Atividade principal

```
package com.example.joaosousa.myapplicationdemo;

import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.preference.PreferenceManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;

import com.phidgets.Phidget;
import com.phidgets.PhidgetException;
import com.phidgets.SpatialPhidget;
import com.phidgets.event.AttachEvent;
import com.phidgets.event.AttachListener;
import com.phidgets.event.DetachEvent;
import com.phidgets.event.DetachListener;


public class MainActivity extends ActionBarActivity {

    SharedPreferences pref_general;
    SpatialPhidget spatial;

    double DR;double N,T,Fmax,Fa;

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

        pref_general = PreferenceManager.getDefaultSharedPreferences(this);

        String op1 = pref_general.getString("max_freq_list","125");
        String op2 = pref_general.getString("time_text","0.8");

        T = Double.valueOf(op2);Fmax = Double.valueOf(op1);
        N = (int)(T*2.56*Fmax);Fa=Fmax*2;
        DR = (1/(Fa))*1000;

        TextView frequencyView = (TextView) findViewById(R.id.max_freq_camp);
        frequencyView.setText(op1 + " Hz");
        TextView timeView = (TextView) findViewById(R.id.reading_time_camp);
        timeView.setText(op2 + getString(R.string.main_screen_time));
        TextView dataView = (TextView) findViewById(R.id.gap_time_camp);
        dataView.setText(DR + getString(R.string.main_screen_timemili));

        try {
            System.out.println("ESTOU AQUI try");
            com.phidgets.usb.Manager.Initialize(this);
            spatial = new SpatialPhidget();
            spatial.addAttachListener(new AttachListener() {
                public void attached(final AttachEvent attachevent) {
                    AttachDetachRunnable handler = new
AttachDetachRunnable(attachevent.getSource(),
                            true);
                    synchronized (handler) {
                        runOnUiThread(handler);
                        try {
                            handler.wait();
                        } catch (InterruptedException exception) {
```

```
                            exception.printStackTrace();
                            System.out.println("ERRO main activity
addAttachListener");
                        }
                    }
                }
            });
            spatial.addDetachListener(new DetachListener() {
                public void detached(final DetachEvent onAttachEvent) {
                    AttachDetachRunnable handler = new AttachDetachRunnable(
                            onAttachEvent.getSource(), false);
                    synchronized(handler)
                    {
                        runOnUiThread(handler);
                        try {
                            handler.wait();
                        } catch (InterruptedException exception) {
                            exception.printStackTrace();
                            System.out.println("ERRO main activity
addDetachListener");
                        }
                    }
                }
            });
            spatial.openAny();
        }
        catch (PhidgetException phidgetException){
            phidgetException.printStackTrace();
            System.out.println("ERRO main activity try");
        }
    }

    @Override
    protected void onPostResume() {
        super.onPostResume();
        try {
            com.phidgets.usb.Manager.Initialize(this);
            spatial = new SpatialPhidget();
            spatial.addAttachListener(new AttachListener() {
                public void attached(final AttachEvent attachevent) {
                    AttachDetachRunnable handler = new
AttachDetachRunnable(attachevent.getSource(),
                            true);
                    synchronized (handler) {
                        runOnUiThread(handler);
                        try {
                            handler.wait();
                        } catch (InterruptedException exception) {
                            exception.printStackTrace();
                            System.out.println("ERRO main activity
addAttachListener tryPostResume");
                        }
                    }
                }
            });
            spatial.addDetachListener(new DetachListener() {
                public void detached(final DetachEvent onAttachEvent) {
                    AttachDetachRunnable handler = new AttachDetachRunnable(
                            onAttachEvent.getSource(), false);
                    synchronized(handler)
                    {
                        runOnUiThread(handler);
                        try {
                            handler.wait();
                        } catch (InterruptedException exception) {
                            exception.printStackTrace();
                            System.out.println("ERRO main activity
addDetachListener tryPostResume");
```

```
                        }
                    }
                }
            });
            spatial.openAny();
        }
        catch (PhidgetException phidgetException){
            phidgetException.printStackTrace();
            System.out.println("ERRO main activity tryPostResume");
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        try{spatial.close();} catch (PhidgetException exception)
        {exception.printStackTrace();System.out.println("ERRO main activity
onDestroy");}
        com.phidgets.usb.Manager.Uninitialize();
    }

    @Override
    protected void onPause() {
        super.onPause();
        try{spatial.close();} catch (PhidgetException exception)
        {exception.printStackTrace();
com.phidgets.usb.Manager.Uninitialize();
        System.out.println("main activity onPause");
    }

    @Override
    protected void onResume() {
        super.onResume();

        pref_general = PreferenceManager.getDefaultSharedPreferences(this);
        String op1 = pref_general.getString("max_freq_list","125");
        String op2 = pref_general.getString("time_text","0.8");

        T = Double.valueOf(op2);Fmax = Double.valueOf(op1);
        N = (int)(T*2.56*Fmax);Fa=Fmax*2;
        DR = (1/(Fa))*1000;

        TextView frequencyView = (TextView) findViewById(R.id.max_freq_camp);
        frequencyView.setText(op1 + " Hz");
        TextView timeView = (TextView) findViewById(R.id.reading_time_camp);
        timeView.setText(op2 + getString(R.string.main_screen_time));
        TextView dataView = (TextView) findViewById(R.id.gap_time_camp);
        dataView.setText(DR + getString(R.string.main_screen_timemili));

        try{spatial.close();} catch (PhidgetException exception)
        {exception.printStackTrace
com.phidgets.usb.Manager.Uninitialize();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        int id = item.getItemId();
        if (id == R.id.action_settings) {
            Intent intent = new Intent(this, SettingsActivity.class);
            this.startActivity(intent);
            return true;
```

```
        }
        if (id == R.id.action_about){
            Intent intent = new Intent(this, SensorInfoActivity.class);
            this.startActivity(intent);
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
    public void StartButton(View view) {

        Intent SecondIntent = new Intent(this, SecondActivity.class);
        startActivity(SecondIntent);
    }

    public void StartLiveButton(View view) {

        Intent ThirdIntent = new Intent(this, ThirdActivity.class);
        startActivity(ThirdIntent);
    }

    public void StartStandardButton(View view) {

        Intent FourthIntent = new Intent(this, StandardsActivity.class);
        startActivity(FourthIntent);
    }

    class AttachDetachRunnable implements Runnable {
        Phidget source;
        boolean sensorStatus;
        public AttachDetachRunnable(Phidget source, boolean sensorStatus) {
            this.source = source;
            this.sensorStatus = sensorStatus;                    }
        public void run(){
            TextView onAttachText = (TextView)
findViewById(R.id.sensor_status_camp);
            if(sensorStatus){
                onAttachText.setText(getString(R.string.status_attached));
                onAttachText.setTextColor(Color.parseColor("#00FF00"));
                try{

                    TextView nameTxt = (TextView)
findViewById(R.id.sensor_name_camp);
                    TextView versionTxt = (TextView)
findViewById(R.id.sensor_version_camp);

                    nameTxt.setText(source.getDeviceName());

versionTxt.setText(Integer.toString(source.getDeviceVersion()));

                } catch (PhidgetException exception) {
                    exception.printStackTrace();
                    System.out.println("ERRO main activity
AttachDetachRunnable");
                }
            }else{
                onAttachText.setText(getString(R.string.status_detached));
                onAttachText.setTextColor(Color.parseColor("#FF0000"));
            }
            synchronized (this){
                this.notify();
            }
        }
    }
}
```

B-2. Segunda Atividade

```
package com.example.joaosousa.myapplicationdemo;

import android.app.ProgressDialog;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.Paint;

import android.hardware.usb.UsbRequest;
import android.preference.PreferenceManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
import android.widget.ToggleButton;
import com.jjoe64.graphview.GraphView;
import com.jjoe64.graphview.GridLabelRenderer;
import com.jjoe64.graphview.LegendRenderer;
import com.jjoe64.graphview.series.DataPoint;
import com.jjoe64.graphview.series.DataPointInterface;
import com.jjoe64.graphview.series.LineGraphSeries;
import com.jjoe64.graphview.series.OnDataPointTapListener;
import com.jjoe64.graphview.series.Series;
import com.phidgets.PhidgetException;
import com.phidgets.SpatialPhidget;
import com.phidgets.event.AttachEvent;
import com.phidgets.event.AttachListener;
import com.phidgets.event.DetachEvent;
import com.phidgets.event.DetachListener;
import com.phidgets.event.SpatialDataEvent;
import com.phidgets.event.SpatialDataListener;
import com.phidgets.usb.Manager;


public class SecondActivity extends ActionBarActivity {


    SharedPreferences pref_general;
    SpatialPhidget spatial;

    private LineGraphSeries<DataPoint> series;
    private LineGraphSeries<DataPoint> seriesFFT;
    private static float[] dataArray, dataArrayFFT;
    private static boolean FILTER,WINDOW;
    private static int axesID,dimension,functions;
    double T,Fmax,DR,Fa;
    int dataIndex;int doIt;int N,filterLow,filterHigh;
    DataPoint[] dataGraph;

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

```
        pref_general =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        String FFmax = pref_general.getString("max_freq_list","125");
        String TT = pref_general.getString("time_text","0.8");
        String AXES = pref_general.getString("axes_list", "2");
        String DIMENSION = pref_general.getString("dimension_text","0");
        FILTER = pref_general.getBoolean("filter", false);
        WINDOW= pref_general.getBoolean("window", false);
        String windowFunction = pref_general.getString("window_list","0");
        String FILTERLow = pref_general.getString("filter_low", "500");
        String FILTERHigh = pref_general.getString("filter_high", "10");
        System.out.println("freq max "+FFmax+" freq max");
        T = Double.valueOf(TT); Fmax = Double.valueOf(FFmax );axesID =
Integer.valueOf(AXES);

filterLow=Integer.valueOf(FILTERLow);filterHigh=Integer.valueOf(FILTERHigh);

dimension=Integer.valueOf(DIMENSION);functions=Integer.valueOf(windowFunction)
;

        N = (int)(T*2.56*Fmax);Fa=Fmax*2;
        DR = (1/Fa*1000);


        dataArray = new float[N];dataGraph=new DataPoint[N];dataIndex = 0;


        try {
            com.phidgets.usb.Manager.Initialize(getApplicationContext());
            spatial = new SpatialPhidget();
            spatial.addAttachListener(new AttachListener() {
                @Override
                public void attached(AttachEvent attachEvent) {
                    try {
                        spatial.setDataRate((int)DR);
                    } catch (PhidgetException e) {
                        e.printStackTrace();
                        System.out.println("ERRO second activity
addAttachListener");
                    }
                }
            });
            spatial.addDetachListener(new DetachListener() {
                @Override
                public void detached(DetachEvent detachEvent) {
                    //qq coisa
                }
            });
            spatial.addSpatialDataListener(new SpatialDataListener() {
                public void data(SpatialDataEvent sde) {
                    runOnUiThread(new SpatialDataRunnable(
                            sde.getData()[0].getAcceleration()[0],
                            sde.getData()[0].getAcceleration()[1],
                            sde.getData()[0].getAcceleration()[2]));
                }
            });
            spatial.openAny();
        }catch (PhidgetException e) {
            e.printStackTrace();
        }
    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_second, menu);
        return true;
    }
```

```
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
    private DataPoint[] generateData() {
        System.out.println("ESTOU AQUI generateData");
        DataPoint[] values = new DataPoint[dataArray.length];
        for (int i=0; i<dataArray.length; i++) {
            double x = i*(T/N);
            double y = dataArray[i];
            System.out.println("ESTOU AQUI generateData" + "t " + x + "pt " +
y);
            DataPoint v = new DataPoint(x, y);
            values[i] = v;
        }
        return values;
    }
    private DataPoint[] generateDataFFT() {
        System.out.println("ESTOU AQUI generateDataFFT");
        DataPoint[] valuesFFT = new DataPoint[dataArrayFFT.length];
        for (int i=0; i<dataArrayFFT.length; i++) {
            double x = i*(1/T);
            double y = dataArrayFFT[i];
            System.out.println("ESTOU AQUI generateDataFFT" + " t " + x + " pt
" + y);
            DataPoint v = new DataPoint(x, y);
            valuesFFT[i] = v;
        }
        return valuesFFT;
    }
    private class SpatialDataRunnable implements Runnable {
        double ax, ay, az,data;
        public SpatialDataRunnable(double ax, double ay, double az) {
            this.ax = (double)Math.round(ax*1000)/1000;
            this.ay = (double)Math.round(ay*1000)/1000;
            this.az = (double)Math.round(az*1000)/1000;
            if (axesID==0){data=ax;}else if
(axesID==1){data=ay;}else{data=az;}
            if (dimension==1||dimension==2){data=9.81*data;}else if
(dimension==3||dimension==5){
                data=data*9.81*1000;}else if
(dimension==4){data=data*9.81*100;}else{}
        }
        public void run() {
            dataArray[dataIndex] = (float)data;
            dataIndex +=1;

        }
    }
    private void doGraph() {
        if (dimension==2||dimension==3){dataArray=
FFTutils.fftV(N,(float)Fa,T, dataArray);
        }else if (dimension==4||dimension==5){dataArray= FFTutils.fftD(N,
(float)Fa,T,dataArray);}else{}
        GraphView graph = (GraphView) findViewById(R.id.graph);
        series = new LineGraphSeries<DataPoint>(generateData());

        graph.getGridLabelRenderer().setGridColor(Color.WHITE);
        graph.getGridLabelRenderer().setHighlightZeroLines(false);
        graph.getGridLabelRenderer().setHorizontalLabelsColor(Color.BLACK);
        graph.getGridLabelRenderer().setVerticalLabelsColor(Color.BLACK);
        graph.getGridLabelRenderer().setVerticalLabelsAlign(Paint.Align.LEFT);
        graph.getGridLabelRenderer().setTextSize(30);
```

```
graph.getGridLabelRenderer().setGridStyle(GridLabelRenderer.GridStyle.HORIZONT
AL);
        graph.getGridLabelRenderer().reloadStyles();

              graph.getViewport().setBackgroundColor(Color.argb(255, 222, 222,
222));

        series.setTitle(getString(R.string.graph_time));
        series.setColor(Color.BLUE);
        series.setDrawDataPoints(true);
        series.setDataPointsRadius(8);
        series.setThickness(6);


        graph.getLegendRenderer().setVisible(true);
        graph.getLegendRenderer().setTextSize(20);
        graph.getLegendRenderer().setBackgroundColor(Color.argb(150, 50, 0,
0));
        graph.getLegendRenderer().setTextColor(Color.WHITE);
        graph.getLegendRenderer().setAlign(LegendRenderer.LegendAlign.TOP);

        series.setOnDataPointTapListener(new OnDataPointTapListener() {
            @Override
            public void onTap(Series series, DataPointInterface dataPoint) {
                Toast.makeText(getApplicationContext(), getString(
                                R.string.graph_ondatapoint) + dataPoint,
                        Toast.LENGTH_SHORT).show();
            }
        });

        //enable scaling
        graph.getViewport().setScalable(true);

        String yAxis = new String();
        if (dimension==0){yAxis=(getString(R.string.string_graph_yaxis_g));
        }else if
(dimension==1){yAxis=(getString(R.string.string_graph_yaxis_a));
        }else if
(dimension==2){yAxis=(getString(R.string.string_graph_yaxis_v));
        }else if
(dimension==3){yAxis=(getString(R.string.string_graph_yaxis_vv));
        }else if
(dimension==4){yAxis=(getString(R.string.string_graph_yaxis_c));
        }else{ yAxis=(getString(R.string.string_graph_yaxis_d));}


        graph.getGridLabelRenderer().setHorizontalAxisTitle(getString(
                R.string.graph_horizontal_time));
        graph.getGridLabelRenderer().setVerticalAxisTitle(yAxis);

        graph.addSeries(series);
    }
    private void runData() {

        if (WINDOW&&functions==0){
        }else if (WINDOW&&functions==1){
            for (int i=0; i<dataArray.length; i++) {
                double t = i*(T/N);
                dataArray[i]=dataArray[i]*(float)(1-Math.cos(2*Math.PI*t)/T);
            }
        }else if (WINDOW&&functions==2){
            double tau = 1.5;
            for (int i=0; i<dataArray.length; i++) {
            double t = i*(T/N);
            dataArray[i]=dataArray[i]*(float)(Math.exp(t/tau));
                System.out.println("ESTOU AQUI generateDataWindow" +
dataArray[i] + " t " +t+ " exp " + (float)(Math.exp(t/tau)));
```

```
            }
        }else{System.out.println("ESTOU AQUI NOwindow " +WINDOW);}
        dataArrayFFT = FFTutils.fft(N,(float)Fa,dataArray);
        doGraphFFT();
    }
    private void doGraphFFT() {
        if (dimension==2||dimension==3){dataArray=
FFTutils.fftV(N,(float)Fa,T,dataArray);
        }else if (dimension==4||dimension==5){dataArray=
FFTutils.fftD(N,(float)Fa,T,dataArray);}else{}
        GraphView graph = (GraphView) findViewById(R.id.graph2);
        seriesFFT = new LineGraphSeries<DataPoint>(generateDataFFT());

        graph.getGridLabelRenderer().setGridColor(Color.WHITE);
        graph.getGridLabelRenderer().setHighlightZeroLines(false);
        graph.getGridLabelRenderer().setHorizontalLabelsColor(Color.BLACK);
        graph.getGridLabelRenderer().setVerticalLabelsColor(Color.BLACK);
        graph.getGridLabelRenderer().setVerticalLabelsAlign(Paint.Align.LEFT);
        graph.getGridLabelRenderer().setTextSize(30);

graph.getGridLabelRenderer().setGridStyle(GridLabelRenderer.GridStyle.HORIZONT
AL);
        graph.getGridLabelRenderer().reloadStyles();

        graph.getViewport().setBackgroundColor(Color.argb(255, 222, 222,
222));

        seriesFFT.setTitle(getString(R.string.graph_frequency));
        seriesFFT.setColor(Color.BLUE);
        seriesFFT.setDrawDataPoints(true);
        seriesFFT.setDataPointsRadius(8);
        seriesFFT.setThickness(6);

        graph.getLegendRenderer().setVisible(true);
        graph.getLegendRenderer().setTextSize(20);
        graph.getLegendRenderer().setBackgroundColor(Color.argb(150, 50, 0,
0));
        graph.getLegendRenderer().setTextColor(Color.WHITE);
        graph.getLegendRenderer().setAlign(LegendRenderer.LegendAlign.TOP);


        graph.getViewport().setScalable(true);

        seriesFFT.setOnDataPointTapListener(new OnDataPointTapListener() {
            @Override
            public void onTap(Series series, DataPointInterface dataPoint) {
                Toast.makeText(getApplicationContext(), getString(
                                R.string.graph_ondatapoint) + dataPoint,
                        Toast.LENGTH_SHORT).show();
            }
        });
        String yAxis = new String();
        if (dimension==0){yAxis=(getString(R.string.string_graph_yaxis_g));
        }else if
(dimension==1){yAxis=(getString(R.string.string_graph_yaxis_a));
        }else if
(dimension==2){yAxis=(getString(R.string.string_graph_yaxis_v));
        }else if
(dimension==3){yAxis=(getString(R.string.string_graph_yaxis_vv));
        }else if
(dimension==4){yAxis=(getString(R.string.string_graph_yaxis_c));
        }else{ yAxis=(getString(R.string.string_graph_yaxis_d));}

        graph.getGridLabelRenderer().setHorizontalAxisTitle(getString
                (R.string.graph_horizontal_frequency));
        graph.getGridLabelRenderer().setVerticalAxisTitle(yAxis);

        graph.addSeries(seriesFFT);
```

```
    }
}
```

B-3. Terceira Atividade

```
package com.example.joaosousa.myapplicationdemo;

import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.util.Log;
import android.view.Menu;
import android.os.Handler;

import com.jjoe64.graphview.GraphView;
import com.jjoe64.graphview.LegendRenderer;
import com.jjoe64.graphview.series.DataPoint;
import com.jjoe64.graphview.series.LineGraphSeries;
import com.phidgets.PhidgetException;
import com.phidgets.SpatialPhidget;
import com.phidgets.event.AttachEvent;
import com.phidgets.event.AttachListener;
import com.phidgets.event.DetachEvent;
import com.phidgets.event.DetachListener;
import com.phidgets.event.SpatialDataEvent;
import com.phidgets.event.SpatialDataListener;

public class ThirdActivity extends ActionBarActivity {

    SpatialPhidget spatial;
    SharedPreferences pref_general;

    private final Handler mHandler = new Handler();
    private Runnable mTimer1;
    private LineGraphSeries<DataPoint> mSeries;
    private LineGraphSeries<DataPoint> mSeriesFFT;
    private GraphView graphFFT,graph;
    private static float[] dataArray, dataArrayFFT;
    private static boolean FILTER,WINDOW;
    private static int axesID,dimension,functions;
    double T,Fmax,Fa,DR;
    int dataIndex;int doIt;int N,filterLow,filterHigh;
    private static DataPoint[] values,dataGraphaf;

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

        pref_general =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        String FFmax = pref_general.getString("max_freq_list","125");
        String TT = pref_general.getString("time_text","0.8");
        String AXES = pref_general.getString("axes_list", "2");
        String DIMENSION = pref_general.getString("dimension_text","0");
        FILTER = pref_general.getBoolean("filter", false);
        WINDOW= pref_general.getBoolean("window", false);
        String windowFunction = pref_general.getString("window_list","0");
        String FILTERLow = pref_general.getString("filter_low", "500");
        String FILTERHigh = pref_general.getString("filter_high", "10");
        System.out.println("freq max "+FFmax+" freq max");
```

```
        T = Double.valueOf(TT); Fmax = Double.valueOf(FFmax );axesID =
Integer.valueOf(AXES);

filterLow=Integer.valueOf(FILTERLow);filterHigh=Integer.valueOf(FILTERHigh);

dimension=Integer.valueOf(DIMENSION);functions=Integer.valueOf(windowFunction)
;

        N = (int)(T*2.56*Fmax);Fa=Fmax*2;
        DR = (1/Fa*1000);

        N=1024;DR=1;
        dataArray = new float[N];dataArrayFFT = new float[N];dataIndex = 0;
        dataGraphaf=new DataPoint[(int)(N/2.56)];

        String yAxis = new String();
        if (dimension==0){yAxis=(getString(R.string.string_graph_yaxis_g));
        }else if
(dimension==1){yAxis=(getString(R.string.string_graph_yaxis_a));
        }else if
(dimension==2){yAxis=(getString(R.string.string_graph_yaxis_v));
        }else if
(dimension==3){yAxis=(getString(R.string.string_graph_yaxis_vv));
        }else if
(dimension==4){yAxis=(getString(R.string.string_graph_yaxis_c));
        }else{ yAxis=(getString(R.string.string_graph_yaxis_d));}

        graph = (GraphView) findViewById(R.id.graph);
        mSeries = new LineGraphSeries<DataPoint>(generateData());
        graph.getGridLabelRenderer().setHorizontalAxisTitle(getString(
                R.string.graph_horizontal_time));
        graph.getGridLabelRenderer().setVerticalAxisTitle(yAxis);
        graph.addSeries(mSeries);

        graphFFT = (GraphView) findViewById(R.id.graph2);
        graphFFT.getGridLabelRenderer().setHorizontalAxisTitle(getString
                (R.string.graph_horizontal_frequency));
        graphFFT.getGridLabelRenderer().setVerticalAxisTitle(yAxis);
    }


    @Override
    protected void onResume() {
        super.onResume();
        mTimer1 = new Runnable() {
            @Override
            public void run() {
                System.out.println("ESTOU AQUI third on resume");
                mSeries.resetData(generateData());
                mHandler.postDelayed(this, 256);}
        };
        mHandler.postDelayed(mTimer1, 256);
    }
    @Override
    protected void onPause() {
        mHandler.removeCallbacks(mTimer1);
        super.onPause();
        try{spatial.close();} catch (PhidgetException exception)
        {exception.printStackTrace();System.out.println("ERRO third activity
onPause");}
        com.phidgets.usb.Manager.Uninitialize();
    }
    @Override
    protected void onStop() {
        super.onStop();
        try{spatial.close();} catch (PhidgetException exception)
        {exception.printStackTrace();System.out.println("ERRO third activity
onStop");}
```

```
        com.phidgets.usb.Manager.Uninitialize();
    }
    private DataPoint[] generateData(){

        try {
            com.phidgets.usb.Manager.Initialize(getApplicationContext());
            spatial = new SpatialPhidget();
            spatial.addAttachListener(new AttachListener() {
                @Override
                public void attached(AttachEvent attachEvent) {
                    try {
                        spatial.setDataRate((int)DR);
                    } catch (PhidgetException e) {
                        e.printStackTrace();
                        System.out.println("ERRO second activity
addAttachListener");
                    }
                }
            });
            spatial.addDetachListener(new DetachListener() {
                @Override
                public void detached(DetachEvent detachEvent) {
                    //qq coisa
                }
            });
            spatial.addSpatialDataListener(new SpatialDataListener() {
                public void data(SpatialDataEvent sde) {
                    runOnUiThread(new SpatialDataRunnable(
                            sde.getData()[0].getAcceleration()[0],
                            sde.getData()[0].getAcceleration()[1],
                            sde.getData()[0].getAcceleration()[2]));
                }
            });
            spatial.openAny();
        }catch (PhidgetException e) {
            e.printStackTrace();
        }
        //int count = 32;
        int count = N;
        DataPoint[] values = new DataPoint[count];
        for (int i=0; i<count; i++) {
            double x = i*(T/N);
            double y = dataArray[i];
            DataPoint v = new DataPoint(x, y);
            values[i] = v;
        }
        return values;
    }
    private DataPoint[] generateDataFFT() {
        for (int i = 0; i < dataArrayFFT.length; i++){
            dataGraphaf[i] = new DataPoint(i*(1/T),dataArrayFFT[i]);
        }
        return dataGraphaf;
    }
    private class SpatialDataRunnable implements Runnable {
        double ax, ay, az,data;
        public SpatialDataRunnable(double ax, double ay, double az) {
            this.ax = (double)Math.round(ax*1000)/1000;
            this.ay = (double)Math.round(ay*1000)/1000;
            this.az = (double)Math.round(az*1000)/1000;
            if (axesID==0){data=ax;}else if
(axesID==1){data=ay;}else{data=az;}
            if (dimension==1||dimension==2){data=9.81*data;}else if
(dimension==3||dimension==5){
                data=data*9.81*1000;}else if
(dimension==4){data=data*9.81*100;}else{System.out.println("ESTOU AQUI
dimension: " + dimension + " 0");}
        }
```

```
        public void run() {
            dataArray[dataIndex] = (float)data;
            System.out.println("ESTOU AQUI SpatialDataRunnable: "+" " +
dataIndex +" " + dataArray[dataIndex]);
            dataIndex +=1;
            if (dataIndex==1024){//tava 32
                dataIndex=0;
                if (dimension==2||dimension==3){dataArray=
FFTutils.fftV(N,(float)Fa,T,dataArray);
                }else if (dimension==4||dimension==5){dataArray=
FFTutils.fftD(N,(float)Fa,T,dataArray);}else{System.out.println("ESTOU AQUI
dimension2: " + dimension + " 0");}
                if (FILTER){
                    System.out.println("ESTOU AQUI run FILTER: " + FILTER);
                    new FILTERutils.Filter(filterHigh,Fa,"high",1,dataArray);
                    new FILTERutils.Filter(filterLow,Fa,"low",1,dataArray);
                }
            }
        }
    }
    private void runData() {
        if (WINDOW&&functions==0){System.out.println("ESTOU AQUI window0 "
+WINDOW+ " " +functions);
        }else if (WINDOW&&functions==1){
            for (int i=0; i<dataArray.length; i++) {
                double t = i*(1/T);
                dataArray[i]=dataArray[i]*(float)(1-Math.cos(2*Math.PI*t)/T);
            }
        }else if (WINDOW&&functions==2){
            double tau = 1.5;
            for (int i=0; i<dataArray.length; i++) {
                double t = i*(1/T);
                dataArray[i]=dataArray[i]*(float)(Math.exp(t/tau));
            }
        }else{ }
        dataArrayFFT = FFTutils.fft(N, (float)Fa, dataArray);
        for (int i = 0; i < dataArrayFFT.length; i++){
            dataGraphaf[i] = new DataPoint(i,dataArrayFFT[i]);
        }
    }
}
```

B-4. Quarta Atividade

```
package com.example.joaosousa.myapplicationdemo;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.preference.PreferenceManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
```

```
import com.phidgets.Phidget;
import com.phidgets.PhidgetException;
import com.phidgets.SpatialPhidget;
import com.phidgets.event.AttachEvent;
import com.phidgets.event.AttachListener;
import com.phidgets.event.DetachEvent;
import com.phidgets.event.DetachListener;
import com.phidgets.event.SpatialDataEvent;
import com.phidgets.event.SpatialDataListener;
import com.phidgets.usb.Manager;

import java.io.IOException;
import java.util.Objects;


public class FourthActivity extends ActionBarActivity {

    SpatialPhidget spatial;
    SharedPreferences pref_general;
    TextView[] valueText,standardText;

    private static double value,noRepeat;
    private static float[] dataArray1,dataArray2,dataArray3;
    int type;
    double T,Fmax,DR,Fa;
    int dataIndex;int doIt;int N;

    private float[] data;
    private float valueLimit;
    private String whatToDo,estrutura,Standard;

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

        pref_general =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        String FFmax = pref_general.getString("max_freq_list","125");
        String TT = pref_general.getString("time_text","0.8");
        T = Double.valueOf(TT); Fmax = Double.valueOf(FFmax );
        N = (int)(T*2.56*Fmax);Fa=Fmax*2;
        DR = (1/Fa*1000);

        //na barra de ferramentas por aviso que os alarmes esta armados

        //N=32;

        View background = getWindow().getDecorView();
        background.setBackgroundColor(Color.WHITE);
        Intent intent = getIntent();
        Standard = intent.getExtras().getString("standard");
        System.out.println(Standard);
        if (Standard.equals("ISO")){
            generateDataISO();
        }else if (Standard.equals("NP")){
            generateDataNP();
        }else if (Standard.equals("BS")){
            runBS();
        }else if (Standard.equals("Custom")){
            runCustom();
        }else {runDin();}

        dataArray1 = new float[N];dataArray2 = new float[N];dataArray3 = new
float[N];

        valueText = new TextView[8];standardText = new TextView[1];
```

```
        valueText[0] = (TextView) findViewById(R.id.valueLive);
        standardText[0] = (TextView) findViewById(R.id.standardText);

    }

    @Override
    protected void onPause() {
        super.onPause();
        try{spatial.close();} catch (PhidgetException exception)
        {exception.printStackTrace();System.out.println("ERRO fourth activity
onPause");}
        com.phidgets.usb.Manager.Uninitialize();
    }

    @Override
    protected void onResume() {
        noRepeat=0;
        super.onResume();
    }


    private void generateDataISO() {
        try {
            com.phidgets.usb.Manager.Initialize(getApplicationContext());
            spatial = new SpatialPhidget();
            spatial.addAttachListener(new AttachListener() {
                @Override
                public void attached(AttachEvent attachEvent) {
                    try {
                        spatial.setDataRate((int)DR);
                    } catch (PhidgetException e) {
                        e.printStackTrace();
                        System.out.println("ERRO second activity
addAttachListener");
                    }
                }
            });
            spatial.addDetachListener(new DetachListener() {
                @Override
                public void detached(DetachEvent detachEvent) {
                    //qq coisa
                }
            });
            spatial.addSpatialDataListener(new SpatialDataListener() {
                public void data(SpatialDataEvent sde) {
                    runOnUiThread(new SpatialDataRunnableISO(
                            sde.getData()[0].getAcceleration()[0],
                            sde.getData()[0].getAcceleration()[1],
                            sde.getData()[0].getAcceleration()[2]));
                }
            });
            spatial.openAny();
        }catch (PhidgetException e) {
            e.printStackTrace();
        }
    }
    private void generateDataNP() {
        try {
            com.phidgets.usb.Manager.Initialize(getApplicationContext());
            spatial = new SpatialPhidget();
            spatial.addAttachListener(new AttachListener() {
                @Override
                public void attached(AttachEvent attachEvent) {
                    try {
                        spatial.setDataRate((int)DR);
                    } catch (PhidgetException e) {
                        e.printStackTrace();
                    }
```

```
                }
            });
            spatial.addDetachListener(new DetachListener() {
                @Override
                public void detached(DetachEvent detachEvent) {
                    //qq coisa
                }
            });
            spatial.addSpatialDataListener(new SpatialDataListener() {
                public void data(SpatialDataEvent sde) {
                    runOnUiThread(new SpatialDataRunnable(
                            sde.getData()[0].getAcceleration()[0],
                            sde.getData()[0].getAcceleration()[1],
                            sde.getData()[0].getAcceleration()[2]));
                }
            });
            spatial.openAny();
        }catch (PhidgetException e) {
            e.printStackTrace();
            System.out.println("erro openany");
        }
    }

    private void runDin() {

        TextView nameText = (TextView) findViewById(R.id.standardText);
        nameText.setText("DIN standard 4150");
    }
    private void runBS() {

        TextView nameText = (TextView) findViewById(R.id.standardText);
        nameText.setText("BS standard 7385");
    }
    private void runNP() {
        System.out.println("ESTIVE AQUI runNP");
        pref_general =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        String TYPE = pref_general.getString("standardNP_list","1");
        type=Integer.valueOf(TYPE);

        data = STANDARDSutils.npStandard(N,(float)Fa,T,dataArray1, dataArray2,
dataArray3);
        //o utilizador tem de ter escolhido o tipo de estruturas
        //estrutura = ("sensiveis");
        if (type==0){estrutura = ("sensiveis");}else if (type==1){estrutura =
("correntes");
        }else{estrutura = ("reforcadas");}

        TextView valueText = (TextView) findViewById(R.id.valueLive);
        valueText.setText(" " + data[0] + " ");
        TextView standardText = (TextView) findViewById(R.id.standardText);
        standardText.setText(" NP 27074 ");

        if (estrutura.equals("sensiveis") && data[1] <= 10) {

            valueLimit =1.5f;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("sensiveis") && (data[1]
>=10)&&(data[1]<=40)) {
```

```
            valueLimit =3;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("sensiveis") && (data[1] >= 40)) {

            valueLimit =6;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("correntes") && (data[1] <= 10)) {

            valueLimit =3;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("correntes") && ((data[1]
>=10)&&(data[1]<=40))) {

            valueLimit =6;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("correntes") && (data[1] >= 40)) {

            valueLimit =12;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("reforcadas") && (data[1] <= 10)) {

            valueLimit =6;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
```

```
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else if (estrutura.equals("reforcadas") && ((data[1]
>=10)&&(data[1]<=40))) {

            valueLimit =12;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        } else {

            valueLimit =40;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
            }else{
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        }
    }
    private void runISO() {
        System.out.println("ESTIVE AQUI runISO");
        pref_general =
PreferenceManager.getDefaultSharedPreferences(FourthActivity.this);
        String ZONE1 = pref_general.getString("standardISO_text1","1");
        String ZONE2 = pref_general.getString("standardISO_text2","2");
        String ZONE3 = pref_general.getString("standardISO_text3","10");
        double zone1 = Double.valueOf(ZONE1);double zone2 =
Double.valueOf(ZONE2);
        double zone3 = Double.valueOf(ZONE3);

        data = STANDARDSutils.isoStandard(N,(float)Fa,T, dataArray1);


        TextView valueText = (TextView) findViewById(R.id.valueLive);
        valueText.setText(" " + data[0] + " ");
        TextView standardText = (TextView) findViewById(R.id.standardText);
        standardText.setText(" ISO 10816 ");

        if (data[0]<=zone1){
            valueLimit =(float)zone1;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] < valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        }else if (data[0]>zone1&&data[0]<=zone2){
            valueLimit =(float)zone2;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.laranja);
                value=0;runAlarm(value);
            }
        }else{
            valueLimit =(float)zone3;
```

```
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (data[0] > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
                value=1;runAlarm(value);
                View background = getWindow().getDecorView();
                background.setBackgroundColor(Color.RED);
            }
        }
        generateDataISO();
    }
    private void runCustom() {
        pref_general =
PreferenceManager.getDefaultSharedPreferences(FourthActivity.this);
        String ZONE1 = pref_general.getString("standardCustom_text1","2");
        String ZONE2 = pref_general.getString("standardCustom_text2","10");
        double zone1 = Double.valueOf(ZONE1);double zone2 =
Double.valueOf(ZONE2);
        int qq = 5;
        TextView nameText = (TextView) findViewById(R.id.standardText);
        nameText.setText("Custom");
        TextView valueText = (TextView) findViewById(R.id.valueLive);
        valueText.setText(" " + qq + " ");

        if (qq<=zone1){
            valueLimit =(float)zone1;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (qq < valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.verde);
            }
        }else if (qq>zone1&&qq<zone2){
            valueLimit =(float)zone1;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (qq > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.laranja);
                value=0;runAlarm(value);
            }
        }else{
            valueLimit =(float)zone2;
            TextView valueTripView = (TextView) findViewById(R.id.valueAlarm);
            valueTripView.setText(" " + valueLimit);
            if (qq > valueLimit){
                ImageView alarmViewR = (ImageView) findViewById(R.id.image);
                alarmViewR.setImageResource(R.drawable.vermelho);
                value=1;runAlarm(value);
                View background = getWindow().getDecorView();
                background.setBackgroundColor(Color.RED);
            }
        }
    }
    private void runAlarm(double value) {
        pref_general =
PreferenceManager.getDefaultSharedPreferences(FourthActivity.this);
        Boolean NOTIFICATION =
pref_general.getBoolean("notifications_new_message", false);
        Boolean SMS = pref_general.getBoolean("notifications_SMS", false);
        String ALARM = pref_general.getString("notifications_ringtone",
                "content://settings/system/notification_sound");
        Uri uri = Uri.parse(ALARM);

        if
(value==0&&NOTIFICATION.equals(true)&&SMS.equals(true)&&noRepeat==0){
            SmsManager sendSMS = SmsManager.getDefault();
```

```
            sendSMS.sendTextMessage("_________",null,"The alarm value has been
exceed",
                    null,null);
            noRepeat=1;         }else if
(value==1&&NOTIFICATION.equals(true)&&SMS.equals(true)&&noRepeat==0){
            SmsManager sendSMS = SmsManager.getDefault();
            sendSMS.sendTextMessage("961819235",null,"The alarm value has been
exceed",
                    null,null);
            noRepeat=1;             playSound(getApplicationContext(), uri);
        }else if (value==1&&NOTIFICATION.equals(true)&&SMS.equals(false)){
            playSound(getApplicationContext(), uri);
        }
    }

    private MediaPlayer player;
    private void playSound(Context context,Uri alarm){
        player = new MediaPlayer();
        try{
            player.setDataSource(context, alarm);
            final AudioManager audio = (AudioManager)
                    context.getSystemService(Context.AUDIO_SERVICE);

            if (audio.getStreamVolume(audio.STREAM_ALARM) !=0){
                player.setAudioStreamType(AudioManager.STREAM_ALARM);
                player.prepare();
                player.start();
            }
        }catch (IOException e){
            System.out.println("ERRO fourth activity mediaplayer");

        }
    }
    private class SpatialDataRunnable implements Runnable {
        double ax, ay, az;
        public SpatialDataRunnable(double ax, double ay, double az) {
            this.ax = (double)Math.round(ax*1000)/1000;
            this.ay = (double)Math.round(ay*1000)/1000;
            this.az = (double)Math.round(az*1000)/1000;
        }
        public void run() {

    }
    private class SpatialDataRunnableISO implements Runnable {
        double ax, ay, az;
        public SpatialDataRunnableISO(double ax, double ay, double az) {
            this.ax = (double)Math.round(ax*1000)/1000;
            this.ay = (double)Math.round(ay*1000)/1000;
            this.az = (double)Math.round(az*1000)/1000;
        }
        public void run() {
            dataArray1[dataIndex] = (float)(az*9.81*1000);
            System.out.println("ESTOU AQUI SpatialDataRunnable: "+" " +
dataIndex +" " + dataArray1[dataIndex]);
            dataIndex +=1;
            if (dataIndex==N){dataIndex=0;
                System.out.println("ESTOU AQUI dataIndex: " +  dataIndex);
                runISO();
            }
        }
    }
}
```