



Instituto Politécnico Nacional  
Escuela Superior de Cómputo



❖ **INTEGRANTES DEL EQUIPO:**

- Barrera Perez Carlos Tonatihu
  - López Higuera Antonio
  - López Sánchez Andrés
  - Monroy Martos Elioth

❖ **UNIDAD DE APRENDIZAJE:** Application Development for Mobile Devices

❖ **PROFESOR:** Alejandro Sigfrido Cifuentes Alvarez.

❖ **PROYECTO TERCER PARCIAL:** Uso del acelerómetro para determinar la dirección de movimiento de un teléfono Android.

❖ **FECHA:** 27 de mayo de 2019.

❖ **Versión del reporte:** 1.0

❖ **GRUPO:** 3CM8.

# Índice

<b>Índice</b>	<b>2</b>
<b>Objetivo</b>	<b>3</b>
Objetivos específicos	3
<b>Conceptos</b>	<b>3</b>
<b>Desarrollo</b>	<b>4</b>
Descripción	4
Desarrollo del circuito	4
Desarrollo de la aplicación móvil	4
Software y hardware utilizados	4
Diagramas	5
Código fuente	6
Pruebas	16
<b>Conclusiones</b>	<b>18</b>
Dificultades encontradas	18
Posibles aplicaciones	18
Conclusiones individuales	19
Barrera Perez Carlos Tonatihu	19
López Higuera Antonio	19
López Sánchez Andrés	19
Monroy Martos Elioth	19
<b>Bibliografía</b>	<b>20</b>

# Objetivo

Desarrollar una aplicación móvil, la cual a través del uso de un sensor del dispositivo Android (como el acelerómetro), permita señalar, con LEDs en una tarjeta Arduino, la dirección en la cual se dirige el dispositivo (adelante, atrás, izquierda, derecha, arriba o abajo).

## Objetivos específicos

- Diseño de una aplicación móvil la cual haga uso del acelerómetro para detectar la dirección hacia la que se mueve el dispositivo.
- Implementar un medio de comunicación entre la aplicación móvil y un Arduino Uno.
- Desarrollar un circuito el cual, mediante LEDs, muestra la dirección hacia donde se mueve el dispositivo.

## Conceptos

Para el entendimiento de esta proyecto es necesario conocer los siguientes conceptos los cuales fueron clave durante el desarrollo de esta aplicación.

- **Java.** Es un lenguaje de programación orientado a objetos, compilado e interpretado por su máquina virtual, Java puede ser utilizado en prácticamente cualquier tipo de proyecto debido a su portabilidad.[1]
- **Android.** Es un sistema operativo para dispositivos móviles, basado en el núcleo de linux, diseñado principalmente para dispositivos táctiles.[2]
- **Activity Android.** Una actividades es un punto de entrada para la interacción con el usuario. Representa una pantalla con una interfaz de usuario.[2]
- **Android Manifest.** Es un archivo que describe la información esencial de la aplicación a las herramientas de construcción de Android, al sistema operativo y a Google Play. En esta se describe el nombre del paquete, los componentes de la aplicación (incluye todas las actividades, servicios, *content provides*, entre otros), los permisos que requiere la aplicación, entre otras declaraciones esenciales. [2]
- **Sensor acelerómetro Android.** Es un sensor que permite conocer el movimiento de un teléfono móvil, a través de tres dimensiones. [3]
- **Bluetooth.** Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) creado por Bluetooth Special Interest Group, Inc. que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. [4]

- **Arduino.** Es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. [5]

## Desarrollo

### Descripción

El desarrollo del proyecto se dividió en dos partes, una parte para la programación de la aplicación móvil, y otra para la implementación del circuito y la codificación del programa para el Arduino.

### Desarrollo del circuito

Para el circuito, se usaron el Arduino Uno, el modulo Bluetooth, 9 leds y cable para hacer las conexiones, en primer instancia se procedió a realizar la conexión del módulo Bluetooth ya que se sabía como conectarlo gracias a proyectos anteriores, en segunda instancia se conectaron todos los leds en la protoboard, y de ahí se conectaron a distintos pines de entrada digital al Arduino.

Para la programación del Arduino Uno, se implementó la comunicación serial para poder trabajar con el módulo Bluetooth y posteriormente se realizó una pequeña lógica la cual se encarga de comparar valores que son recibidos mediante el serial, para saber si hay que prender algún led, y en caso positivo cual (o cuales) tiene que ser prendido.

### Desarrollo de la aplicación móvil

Para la aplicación móvil, primero se implementaron las clases necesarias para enviar información mediante Bluetooth al Arduino, ya que esto se había realizado en otros proyectos anteriores por lo cual fue posible reutilizar el código.

### Software y hardware utilizados

A continuación, se presenta una lista del software y hardware usado para la elaboración del proyecto:

- **Android Studio 3.4.** Android IDE para el desarrollo de aplicaciones para el sistema operativo Android.
- **Teléfono con Android 5+.** Usado para realizar la pruebas correspondientes.
- **Java.** Lenguaje de programación multipropósito con el cual fue programada la aplicación móvil y el servidor web.

- **Java Development Kit 8.** Necesario para el desarrollo de aplicaciones Java.
- **Arduino UNO.** Es una placa con un microcontrolador que permite su programación para el desarrollo de aplicaciones electrónicas.
- **Arduino IDE.** Entorno de desarrollo para arduino que permite crear programas y además permite cargar los mismos a la placa de Arduino.
- **Modulo Bluetooth HC-05.** Es el módulo que se utilizó en la placa arduino para la comunicación entre dicho microcontrolador y un teléfono celular, debido a que es un módulo que se ajustaba a las necesidades del proyecto, ya que no es costoso, y su uso es sencillo.
- **Otros componentes.** Leds de distintos colores.

## Diagramas

Los diagramas siguientes muestran el circuito implementado para mostrar la funcionalidad de la aplicación móvil, es general se compone de la conexión entre el Arduino Uno y el módulo Bluetooth, y los leds de distintos colores que señalan hacia que dirección se mueve el teléfono móvil.

En la Figura 1 se puede observar, cómo es que se realizó la conexión entre el Arduino Uno y el módulo Bluetooth HC-05.

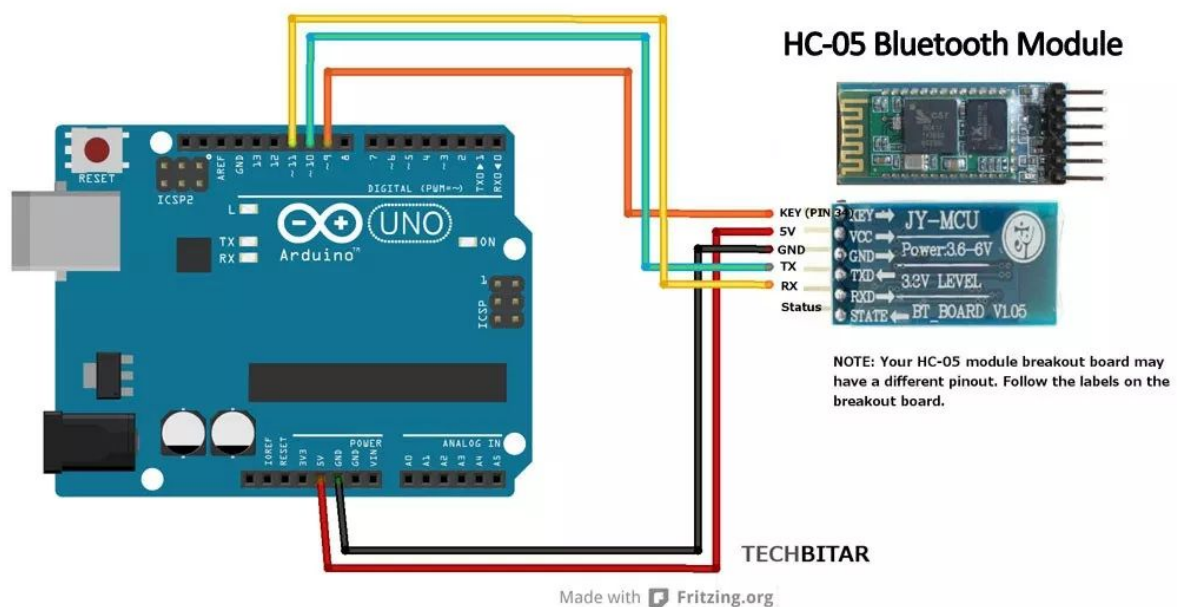


Figura 1. Conexión Arduino Uno con módulo Bluetooth HC-05.

En la Figura 2, se muestra el diagrama correspondiente a como se realiza la conexión entre el Arduino Uno y los leds.

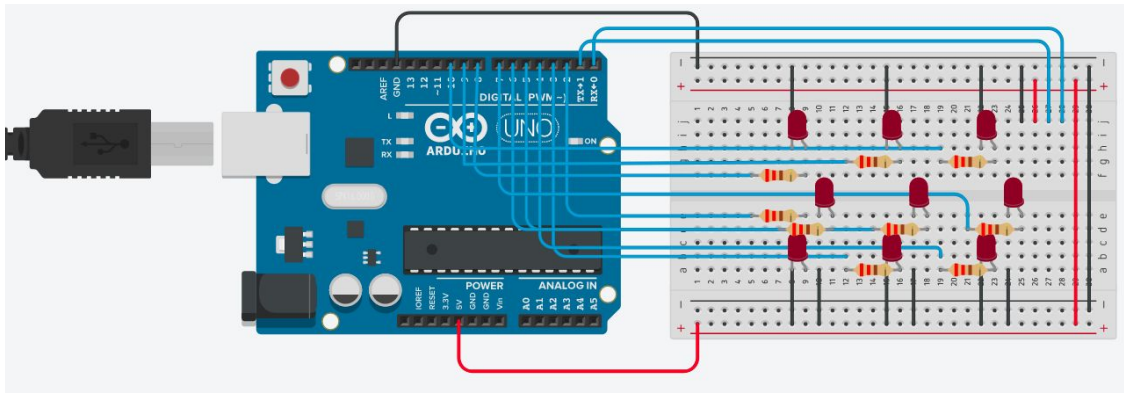


Figura 2. Conexión del Arduino Uno con los 9 LEDs.

Finalmente, en la Figura 3, se muestra el circuito ensamblado en la protoboard con el Arduino Uno.

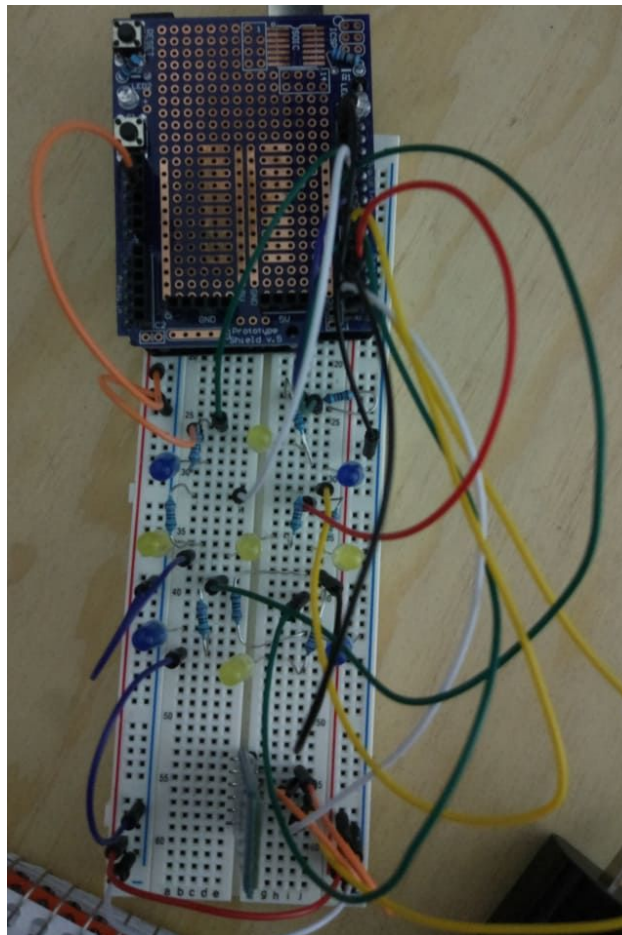


Figura 3. Circuito completo ensamblado.

## Código fuente

A continuación, se presenta el código más importante elaborado, para el desarrollo del proyecto.

El siguiente código, muestra el archivo cargado al Arduino Uno.

## BlueLed.ino

```
int pinLED[] = {2,3,4,5,6,7,8,9,10};
int values[] = {1,2,4,8,16,32,64,128,256};
String data;
char dataArray[50];
int value, value2;
void setup() {
    Serial.begin(9600);
    for(int i = 0; i < 9; i++){
        pinMode(pinLED[i], OUTPUT);
        digitalWrite(pinLED[i], HIGH);
        delay(110);
        digitalWrite(pinLED[i], LOW);
    }
}
void loop(){
    if(Serial.available() > 0) // Si la conexión serial esta disponible
    activaremos o no el LED
    {
        char buffer[60]=" ";
        data = Serial.readString(); //Leemos los datos
        value = data.substring(1).toInt();
        value2 = data.substring(0,1).toInt();
        if(value2 > 0){
            digitalWrite(pinLED[4], HIGH);
        }else{
            digitalWrite(pinLED[4], LOW);
        }
        for(int i = 0; i < 9; i++){
            check(value, i, values[i]);
        }
        // Lineas de impresión de datos en la consola
        Serial.print("1 - "+data+" - "+(value2 > 0));
        Serial.print("\n");
        Serial.flush();
    }
}
void check(int value, int i, int expected){
    if(i != 4){
        if(value & expected){
            digitalWrite(pinLED[i], HIGH);
        }else{
            digitalWrite(pinLED[i], LOW);
        }
    }
}
```

A continuación, se muestra el archivo manifest realizado para que la aplicación tuviera los permisos necesarios para funcionar.

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:dist="http://schemas.android.com/apk/distribution"
    package="com.durazno.accelerometer">
```

```

<dist:module dist:instant="true" />
<uses-feature
    android:name="android.hardware.sensor.accelerometer"
    android:required="true" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".DiscoverActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <action android:name="android.intent.action.VIEW" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>

```

## ActivityMain.java:

```

package com.durazno.accelerometer;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import androidx.appcompat.app.AppCompatActivity;
import com.durazno.accelerometer.entities.ValueSender;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.text.DecimalFormat;
import java.util.UUID;
import butterknife.BindView;
import butterknife.ButterKnife;

```



```

public class MainActivity extends AppCompatActivity implements
SensorEventListener {
    private static final String TAG = "MainActivity";
    private static final UUID mUUID =
UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    private boolean rightAxis = false;
    private boolean leftAxis = false;
    private boolean topAxis = false;
    private boolean bottomAxis = false;
    private boolean topLeftAxis = false;
    private boolean bottomLeftAxis = false;
    private boolean topRightAxis = false;
    private boolean bottomRightAxis = false;
    private boolean isBtConnected = false;
    private boolean centerAxis = false;
    private ValueSender valueSender;
    private String address;
    BluetoothAdapter myBluetooth = null;
    BluetoothSocket btSocket = null;
    InputStream socketInputStream;
    OutputStream socketOutputStream;
    //View Objects
    @BindView(R.id.btn_bottom)
    ToggleButton bottom;
    @BindView(R.id.btn_top)
    ToggleButton top;
    @BindView(R.id.btn_left)
    ToggleButton left;
    @BindView(R.id.btn_right)
    ToggleButton right;
    @BindView(R.id.btn_center)
    ToggleButton center;
    @BindView(R.id.btn_bottomRight)
    ToggleButton bottomRight;
    @BindView(R.id.btn_topRight)
    ToggleButton topRight;
    @BindView(R.id.btn_bottomLeft)
    ToggleButton bottomLeft;
    @BindView(R.id.btn_topLeft)
    ToggleButton topLeft;
    @BindView(R.id.tv_datosX)
    TextView tvDatosX;
    @BindView(R.id.tv_datosY)
    TextView tvDatosY;
    @BindView(R.id.tv_datosZ)
    TextView tvDatosZ;
    @BindView(R.id.tv_datosBt)
    TextView tvDatosBt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.bind(this);
        SensorManager mSensorManager;
        Sensor sensor;
        address =
getIntent().getStringExtra(DiscoverActivity.EXTRA_ADDRESS);

```

```

        valueSender = new ValueSender("d0");
        Log.d(TAG, "onCreate: Initializing Sensor Services");
        mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        sensor =
mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        mSensorManager.registerListener(this, sensor,
SensorManager.SENSOR_DELAY_NORMAL);
        Log.d(TAG, "onCreate: Registered accelerometer");
        new ConnectBT(valueSender).execute();
    }
    @Override
    public void onSensorChanged(SensorEvent event) {
        int value = 0;
        String s = "";
        float x = event.values[0];
        float y = event.values[1];
        float z = event.values[2];
        if (x > 0.85) {
            leftAxis = true;
            rightAxis = false;
            value += 8;
        } else if (x < -.85) {
            rightAxis = true;
            leftAxis = false;
            value += 32;
        } else {
            rightAxis = false;
            leftAxis = false;
        }
        if (y > 0.85) {
            bottomAxis = true;
            topAxis = false;
            value += 2;
        } else if (y < -.85) {
            topAxis = true;
            bottomAxis = false;
            value += 128;
        } else {
            topAxis = bottomAxis= false;
        }
        if (topAxis) {
            if (leftAxis) {
                value -= 72;
                topLeftAxis = true;
                topAxis = leftAxis = false;
            } else {
                topLeftAxis = false;
            }
            if (rightAxis) {
                value += 96;
                topRightAxis = true;
                topAxis = rightAxis = false;
            } else {
                topRightAxis = false;
            }
        } else {

```

```

        topLeftAxis = topRightAxis = false;
    }
    if (bottomAxis) {
        if (leftAxis) {
            value -= 9;
            bottomLeftAxis = true;
            leftAxis = false;
        }
        if (rightAxis) {
            value -= 30;
            bottomRightAxis = true;
            rightAxis = false;
        }
    }
    else {
        bottomRightAxis = false;
        bottomLeftAxis = false;
    }
    if (z < 9 || z > 10) {
        centerAxis = true;
        s = "1" + value;
    } else {
        centerAxis = false;
        s = "0" + value;
    }
    valueSender.setValue(s);
    updateToggleButton();
    DecimalFormat decimalFormat = new DecimalFormat();
    decimalFormat.setMaximumFractionDigits(2);
    Log.d(TAG, "onSensorChanged: Z: " + event.values[2]);
    tvDatosX.setText(getString(R.string.X, decimalFormat.format(x)));
    tvDatosY.setText(getString(R.string.Y, decimalFormat.format(y)));
    tvDatosZ.setText(getString(R.string.Z, decimalFormat.format(z)));
    tvDatosBt.setText(getString(R.string.Bt, valueSender.getValue()));
}
private void updateToggleButton() {
    top.setChecked(topAxis);
    bottom.setChecked(bottomAxis);
    right.setChecked(rightAxis);
    left.setChecked(leftAxis);
    topLeft.setChecked(topLeftAxis);
    topRight.setChecked(topRightAxis);
    bottomLeft.setChecked(bottomLeftAxis);
    bottomRight.setChecked(bottomRightAxis);
    center.setChecked(centerAxis);
}
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    Log.d(TAG, "onAccuracyChanged: ");
}
private class ConnectBT extends AsyncTask<Void, Void, Void> {
    private boolean connectSuccess = true; //if it's here, it's almost
connected
    private ValueSender valueSender;
    String readMessage = "1";
    public ConnectBT(ValueSender valueSender) {
        this.valueSender = valueSender;
    }
}

```

```

@Override
protected Void doInBackground(Void... devices) //while the
progress dialog is shown, the connection is done in background
{
    try {
        if (btSocket == null || !isBtConnected) {
            myBluetooth =
BluetoothAdapter.getDefaultAdapter();//get the mobile bluetooth device
BluetoothDevice device =
myBluetooth.getRemoteDevice(address);//connects to the device's address
and checks if it's available
            btSocket =
device.createInsecureRfcommSocketToServiceRecord(mUUID);//create a RFCOMM
(SPP) connection
            BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
            btSocket.connect();//start connection
        }
    } catch (IOException e) {
        Log.d(TAG, "doInBackground: "+ e.toString());
        connectSuccess = false;//if the try failed, you can check
the exception here
    }
    return null;
}

@Override
protected void onPostExecute(Void result) {
    super.onPostExecute(result);
    if (!connectSuccess) {
        msg("Connection Failed. Is it a SPP Bluetooth? Try again.");
        finish();
    } else {
        msg("Connected.");
        isBtConnected = true;
        try {
            socketOutputStream = btSocket.getOutputStream();
            socketInputStream = btSocket.getInputStream();
            socketOutputStream.flush();
        } catch (Exception e) {
            Log.d(TAG, "onPostExecute: "+e.toString());
        }
        final Handler handler = new Handler();
        final Runnable[] runnables = new Runnable[1];
        runnables[0] = () -> {
            try {
                byte[] buffer2 = new byte[60];
                byte[] buffer = valueSender.getValue().getBytes();
                int bytes;
                if (readMessage.charAt(0) == '1') {
                    socketOutputStream.write(buffer);
                    Log.d(TAG, "onPostExecute: " + readMessage);
                    socketOutputStream.flush();
                }
                bytes = socketInputStream.read(buffer2);
                readMessage = new String(buffer2, 0, bytes);
            } catch (IOException e) {
                Log.d(TAG, "onPostExecute: "+ e.toString());
            }
        }
    }
}

```

```

        // Repeat every 1 seconds
        handler.postDelayed(runnables[0], 100);
    };
    handler.post(runnables[0]);
}

}

private void msg(String s) {
    Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_LONG).show();
}
}
}

```

activity\_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
    <ToggleButton
        android:id="@+id/btn_bottom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginBottom="160dp"/>
    <TextView
        android:id="@+id/tv_datosY"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>
    <TextView
        android:id="@+id/tv_datosBt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:layout_marginTop="10dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>
    <TextView
        android:id="@+id/tv_datosZ"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"

```

```

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>
<TextView
    android:id="@+id/tv_datosX"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>
<ToggleButton
    android:id="@+id/btn_right"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginEnd="10dp"/>
<ToggleButton
    android:id="@+id/btn_left"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginStart="10dp"/>
<ToggleButton
    android:id="@+id/btn_top"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="160dp"/>
<ToggleButton
    android:id="@+id/btn_topLeft"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="160dp"
    android:layout_marginStart="10dp"/>
<ToggleButton
    android:id="@+id/btn_topRight"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="160dp"
    android:layout_marginEnd="10dp"/>
<ToggleButton

```

```

        android:id="@+id/btn_center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>
<ToggleButton
    android:id="@+id/btn_bottomLeft"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="160dp"
    android:layout_marginStart="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"/>
<ToggleButton
    android:id="@+id/btn_bottomRight"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="160dp"
    android:layout_marginEnd="10dp"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

## Pruebas

A continuación se muestran algunas capturas de pantalla de la aplicación móvil.

En la Figura 4, se muestra la pantalla de la aplicación que permite seleccionar el dispositivo Bluetooth al que se requiere conectarse.

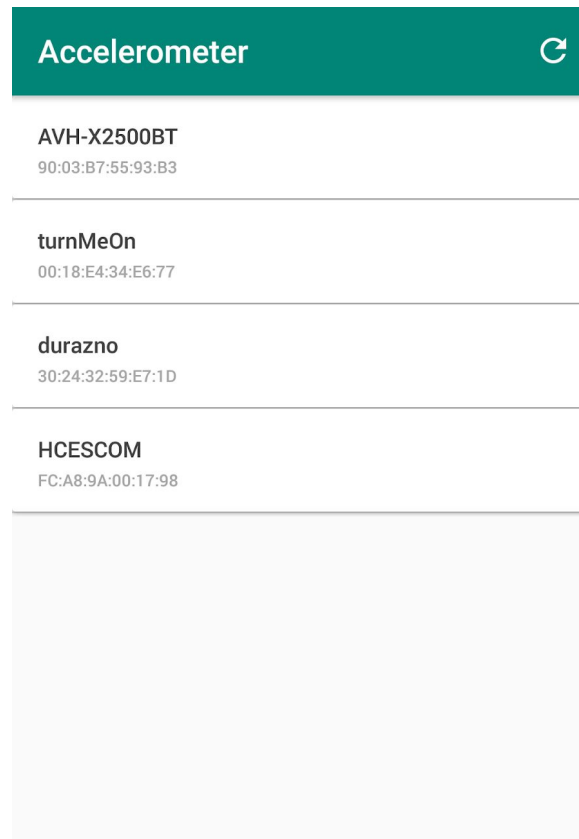


Figura 4. Selección del dispositivo Bluetooth.

Una vez que se conecta al dispositivo Bluetooth, se muestra la pantalla de la Figura 5, la cual es la pantalla principal y emula a los leds que se encuentran en la protoboard, ya que muestra nueve botones los cuales, pueden estar prendido u apagados según sea el movimiento del teléfono, al igual que lo hacen los leds. De igual forma, se muestra en la parte inferior de la pantalla, la posición del teléfono, en los tres ejes (x,y,z).



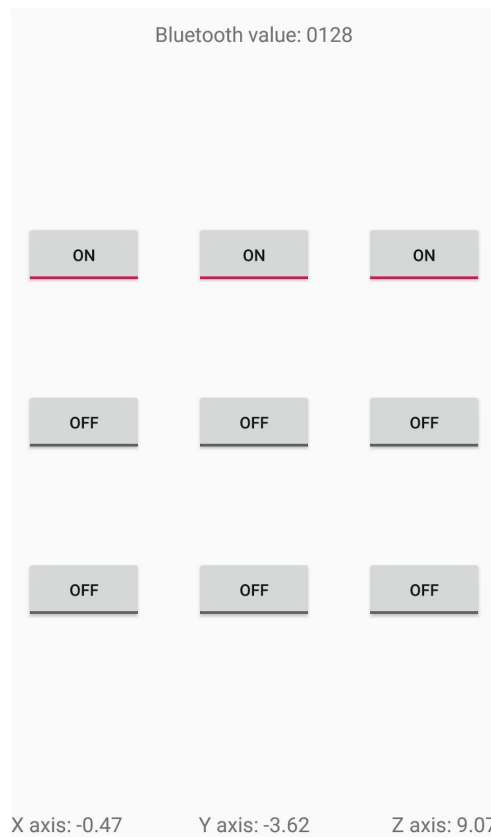


Figura 5. Pantalla principal de la aplicación.

Finalmente, en las Figuras 6 y 7, se muestran fotografías de como se ven los leds prendidos según sea el movimiento del teléfono móvil.

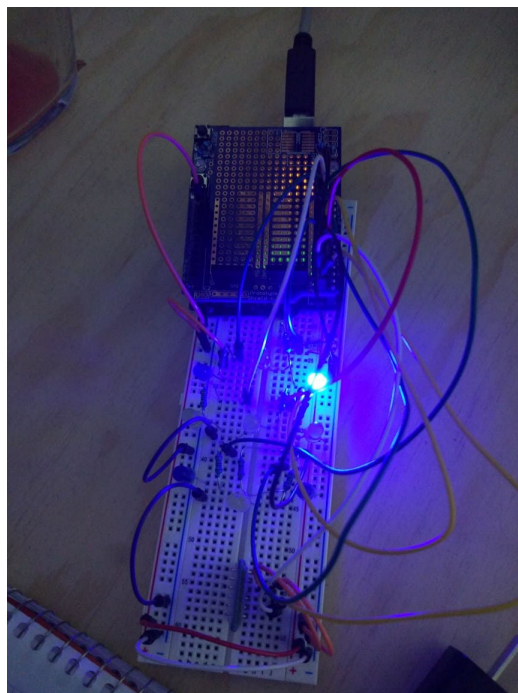


Figura 6. Led encendido debido a que se detecto movimiento en el dispositivo móvil (1).

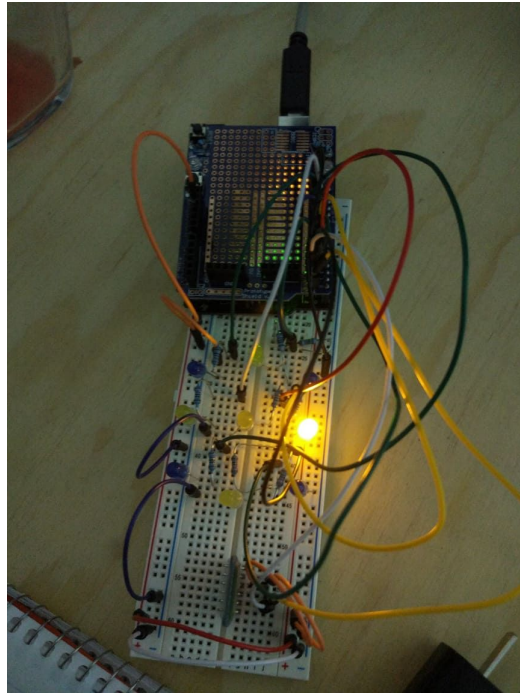


Figura 7. Led encendido debido a que se detectó movimiento en el dispositivo móvil (2).

## Conclusiones

### Dificultades encontradas

Al principio, resultó complicado el entendimiento del acelerómetro, la posición en la que tenía que permanecer. El eje Z que marca la profundidad es complejo de calcular, este no siempre funcionaba como debía funcionar, se ven alterados todos los ejes, por lo cual es difícil definir con precisión cuando baja y cuando sube, debido a que si lo mueves en algún eje este también cambia, pero cuando haces un movimiento agresivo sobre el eje Z, este valor varía demasiado.

### Posibles aplicaciones

El uso de los distintos sensores del dispositivo móvil, permite dar una mayor funcionalidad a nuestras aplicaciones Android, el uso del acelerómetro nos permite desarrollar aplicaciones las cuales hagan uso del movimiento del usuario o del dispositivo, por ejemplo, podríamos detectar la velocidad a la que se mueve el dispositivo y por ende, realizar un cálculo de la velocidad a la que se desplaza un usuario en un automóvil. Lo cual, en combinación con el uso del API de Google Maps, se podría mostrar un mapa al usuario e indicar la velocidad a la que se está moviendo, esto con el fin de indicarle si se encuentra dentro del límite de velocidad permitido o no. Otra aplicación, podría ser detectar la cantidad de actividad física que realiza el usuario, a través del movimiento que se detecta en el dispositivo. Esto

con la finalidad de indicarle, por ejemplo, la cantidad de calorías que quemó durante una sesión de ejercicio.

## Conclusiones individuales

### Barrera Perez Carlos Tonatihu

Este proyecto es solo una muestra de lo versátil que pueden llegar a ser los proyectos con Android, ya que a diferencia de los midlets, esta plataforma nos ofrece más herramientas con que trabajar. En este caso fueron los sensores, en concreto el acelerómetro con el cual se pueden hacer cosas interesantes ya que nos permite conocer el movimiento del teléfono. Debido a que no se había trabajado con este sensor de tuvo que investigar cómo se utiliza ya que en un inicio resultó ser la parte más complicada del proyecto ya que el resto de este ya se había trabajado con anterioridad en los proyectos de los primeros dos parciales.

### López Higuera Antonio

Los dispositivos móviles actualmente te permiten realizar todo tipo de proyectos. realmente la creatividad o la factibilidad de proyecto es lo único que te limitan, puesto que los dispositivos móviles cuentan con tantas herramientas en sensores y que están disponibles para el uso dentro de aplicaciones. Quedé muy satisfecho en la realización del proyecto.

### López Sánchez Andrés

La realización del proyecto resultó interesante pues el uso del acelerómetro del dispositivo móvil para poder captar el movimiento dio como resultado un circuito capaz de responder al cambio de posición encendiendo LEDs, que sí quizás parece algo simple, brinda la oportunidad de poder aplicar este conocimiento en algún proyecto un poco más desarrollado o enfocarlo a alguna solución simple. También agregar que durante el desarrollo aplicamos y reforzamos conocimientos visto en clase y durante realización de los demás proyectos.

### Monroy Martos Elioth

Con este proyecto aplicamos los conocimientos que fuimos adquiriendo durante el transcurso de la materia, como el uso del módulo Bluetooth HC-05 para la comunicación inalámbrica entre el dispositivo móvil y el Arduino, además de que aprendimos a usar un sensor del teléfono, lo cual es interesante ya que así como se puede hacer uso de este sensor, se pueden hacer uso de otros sensores del teléfono como el giroscopio. Finalmente, el trabajar con un sensor, nos permite extender la funcionalidad de nuestras aplicaciones móviles.

## Bibliografía

- [1] M. Lozano Ortega and B. Bonev, "Introducción a Android", *Curso de Android y Java para Dispositivos Móviles*, 2010. [Online]. Disponible en: <http://www.jtech.ua.es/apuntes/ajdm2010/sesiones/sesion09-apuntes.html>. [Consultado: 10- Abril- 2019].
- [2] "Application Fundamentals", Android Developer, 2019. [Online]. Disponible en: <https://developer.android.com/guide/components/fundamentals>. [Consultado: 10-Abril-2019].
- [3] "¿Cómo funciona exactamente el acelerómetro de un smartphone?", AndroidPit, 2012. [Online]. Disponible en: <https://www.androidpit.es/funcionamiento-acelerometro-smartphones>. [Consultado: 24-Mayo-2019].
- [4] "Bluetooth", *Wikipedia*, 2019. [Online]. Disponible en: <https://es.wikipedia.org/wiki/Bluetooth>. [Consultado: 25- Feb- 2019].
- [5] Y. FM, "Qué es Arduino, cómo funciona y qué puedes hacer con uno", *Xataka.com*, 2018. [Online]. Disponible en: <https://www.xataka.com/basics/que-arduino-como-functiona-que-puedes-hacer-uno>. [Consultado: 25- Feb- 2019].