**Manual Técnico**

**Grupo 5**

**Carro de carga operado mediante una App móvil para personas que tienen incapacidad para transportar objetos domésticos pesados**

**Integrantes:**

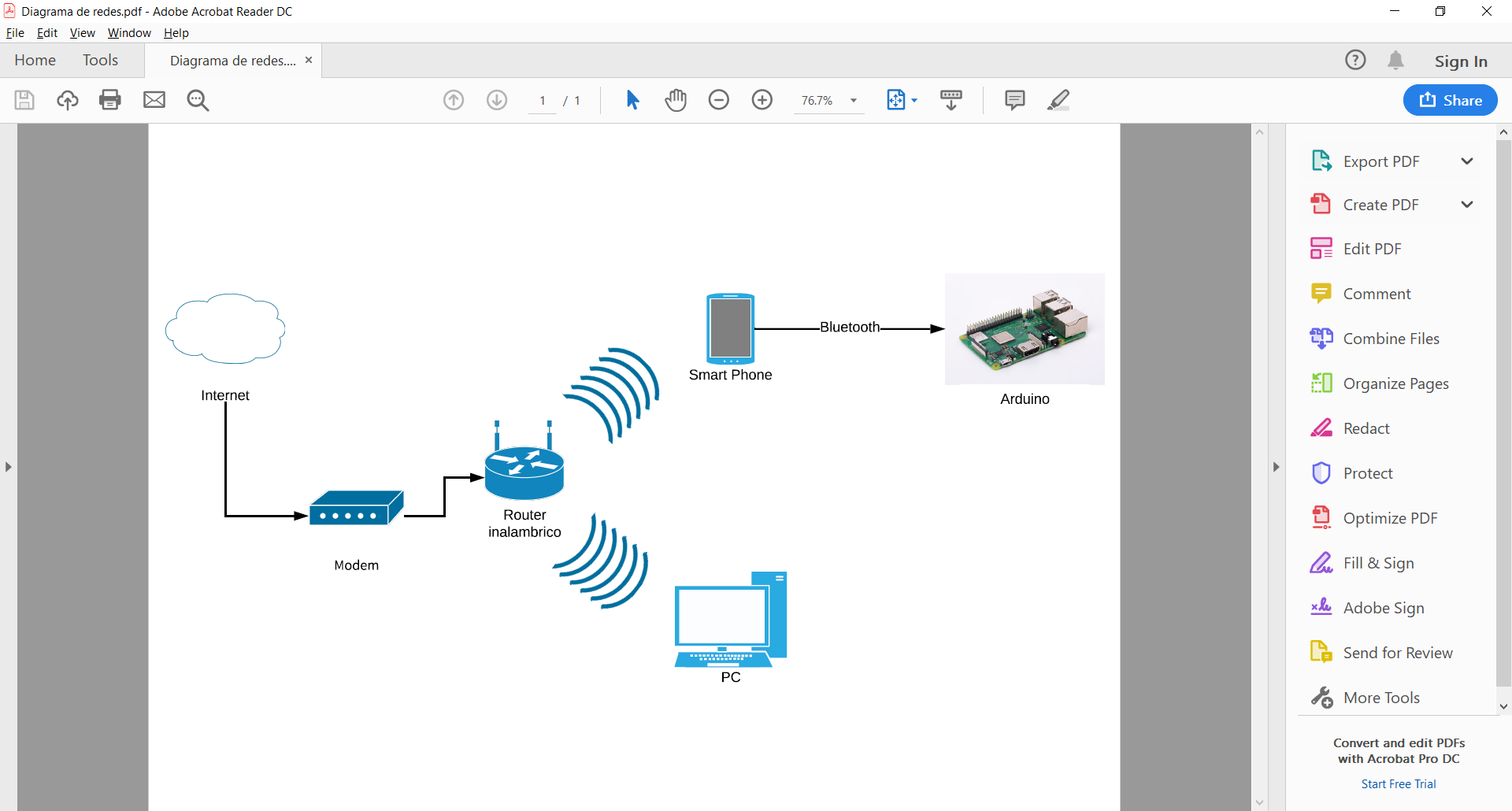
**Gabriel Murillo**

**Charly González**

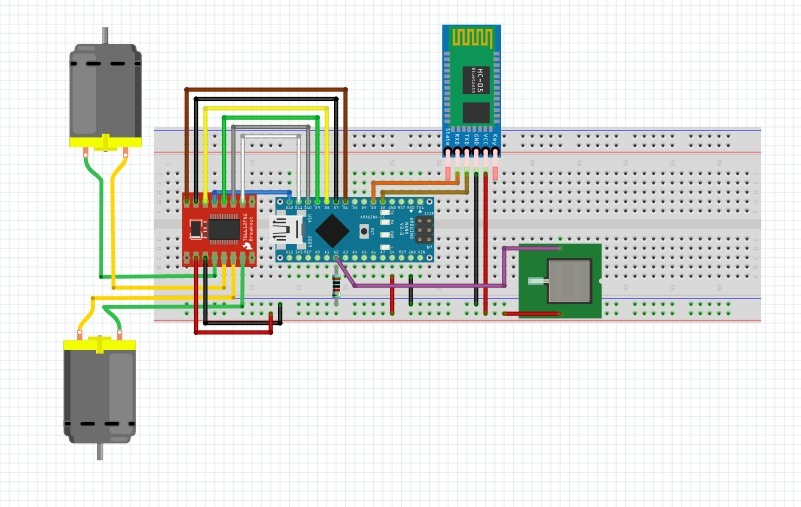
**Ivan Buendia**

**Wilman Cuero**

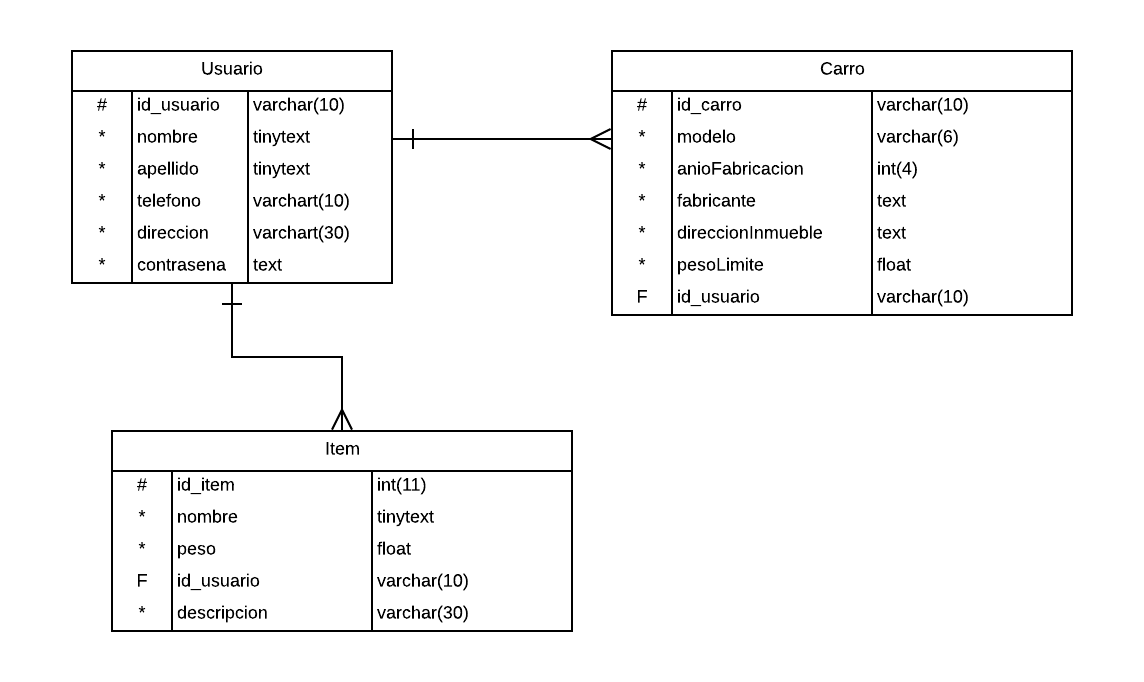
**Diagrama de red**



**Diagrama de conexion**



**Diagrama de entidad-relación**



**Código Arduino**

#include <SoftwareSerial.h>

SoftwareSerial ModBluetooth(2, 3); // RX | TX

int AnalogPin = 2; // Sensor conectado a Analog 0

int LEDpin = 5; // LED conectado a Pin 6 (PWM)

int ResRead; // La Lectura de la Resistencia por División de Tensión

int BrilloLED;

const int pinPWMA = 6;

const int pinAIN2 = 7;

const int pinAIN1 = 8;

const int pinBIN1 = 9;

const int pinBIN2 = 10;

const int pinPWMB = 11;

const int pinSTBY = 12;

const int pinLed = 4;

const int pinMotorA[3] = { pinPWMA, pinAIN2, pinAIN1 };

const int pinMotorB[3] = { pinPWMB, pinBIN1, pinBIN2 };

enum moveDirection {forward,backward};

void setup(){

pinMode(pinAIN2, OUTPUT);

pinMode(pinAIN1, OUTPUT);

pinMode(pinPWMA, OUTPUT);

pinMode(pinBIN1, OUTPUT);

pinMode(pinBIN2, OUTPUT);

pinMode(pinPWMB, OUTPUT);

pinMode(pinLed, OUTPUT);

digitalWrite(pinLed, LOW);

ModBluetooth.begin(9600);

Serial.begin(9600);

ModBluetooth.println("MODULO CONECTADO");

ModBluetooth.print("#");

}

void loop(){

enableMotors();

//Sensor de peso

ResRead = analogRead(AnalogPin); // La Resistencia es igual a la lectura del sensor (Analog 0)

Serial.print("Lectura Analogica = ");

Serial.println(ResRead);

BrilloLED = map(ResRead, 0, 1023, 0, 255);

// Cambiar el rango de la lectura analógica (0-1023)

// Utilizamos en analogWrite 8 bits (0-255) configurados en el map

analogWrite(LEDpin, BrilloLED);

ModBluetooth.print("Peso Actual"+String(" ")+String(ResRead)+String(" ")+"gramos");

ModBluetooth.print("#");

delay(500); //Cien “ms” de espera en cada lectura

//Bluethoot

while (ModBluetooth.available()){

char VarChar;

VarChar = ModBluetooth.read();

if(VarChar == '0'){

digitalWrite(pinLed, HIGH);

delay(100);

ModBluetooth.print("RETROCEDIENDO");

Serial.print("LED ENCENDIDO");

ModBluetooth.print("#");

move(backward, 80);

}

if(VarChar == '1') {

digitalWrite(pinLed, HIGH);

delay(100);

ModBluetooth.print("AVANZANDO");

Serial.print("LED ENCENDIDO");

ModBluetooth.print("#");

move(forward, 80);

}

if(VarChar == '2') {

digitalWrite(pinLed, LOW);

delay(100);

ModBluetooth.print("DETENIDO");

Serial.print("LED APAGADO");

ModBluetooth.print("#");

fullStop();

}

if(VarChar == '3') {

digitalWrite(pinLed, LOW);

delay(100);

ModBluetooth.print("DERECHA");

Serial.print("LED ENCENDIDO");

ModBluetooth.print("#");

turn(forward,40);

}

if(VarChar == '4') {

digitalWrite(pinLed, LOW);

delay(100);

ModBluetooth.print("IZQUIERDA");

Serial.print("LED ENCENDIDO");

ModBluetooth.print("#");

turn(backward,40);

}

}

}

//Funciones que controlan el vehiculo

void move(int direction, int speed)

{

if (direction == forward)

{

moveMotorForward(pinMotorA, speed);

moveMotorForward(pinMotorB, speed);

}

else

{

moveMotorBackward(pinMotorA, speed);

moveMotorBackward(pinMotorB, speed);

}

}

void turn(int direction, int speed)

{

if (direction == forward)

{

moveMotorForward(pinMotorA, speed);

moveMotorBackward(pinMotorB, speed);

}

else

{

moveMotorBackward(pinMotorA, speed);

moveMotorForward(pinMotorB, speed);

}

}

void fullStop()

{

disableMotors();

stopMotor(pinMotorA);

stopMotor(pinMotorB);

}

//Funciones que controlan los motores

void moveMotorForward(const int pinMotor[3], int speed)

{

digitalWrite(pinMotor[1], LOW);

digitalWrite(pinMotor[2], HIGH);

analogWrite(pinMotor[0], speed);

}

void moveMotorBackward(const int pinMotor[3], int speed)

{

digitalWrite(pinMotor[1], HIGH);

digitalWrite(pinMotor[2], LOW);

analogWrite(pinMotor[0], speed);

}

void stopMotor(const int pinMotor[3]){

digitalWrite(pinMotor[1], LOW);

digitalWrite(pinMotor[2], LOW);

analogWrite(pinMotor[0], 0);

}

void enableMotors(){

digitalWrite(pinSTBY, HIGH);

}

void disableMotors(){

digitalWrite(pinSTBY, LOW);

}

**Código Android Studio**

**Clase Bluetooth (Java)**

package com.example.pst\_carro;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import java.util.Set;

public class bluetooth extends AppCompatActivity {

//1)

// Depuración de LOGCAT

private static final String TAG = "dispositivos\_bt"; //<-<- PARTE A MODIFICAR >->->

// Declaracion de ListView

ListView IdLista;

// String que se enviara a la actividad principal, mainactivity

public static String EXTRA\_DEVICE\_ADDRESS = "device\_address";

// Declaracion de campos

private BluetoothAdapter mBtAdapter;

private ArrayAdapter mPairedDevicesArrayAdapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bluetooth);

}

public void onResume() {

super.onResume();

//---------------------------------

VerificarEstadoBT();

// Inicializa la array que contendra la lista de los dispositivos bluetooth vinculados

mPairedDevicesArrayAdapter = new ArrayAdapter(this, R.layout.textviewlayout);//<-<- PARTE A MODIFICAR >->->

// Presenta los disposisitivos vinculados en el ListView

IdLista = (ListView) findViewById(R.id.IdList);

IdLista.setAdapter(mPairedDevicesArrayAdapter);

IdLista.setOnItemClickListener(mDeviceClickListener);

// Obtiene el adaptador local Bluetooth adapter

mBtAdapter = BluetoothAdapter.getDefaultAdapter();

//------------------- EN CASO DE ERROR -------------------------------------

//SI OBTIENES UN ERROR EN LA LINEA (BluetoothDevice device : pairedDevices)

//CAMBIA LA SIGUIENTE LINEA POR

//Set <BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();

//------------------------------------------------------------------------------

// Obtiene un conjunto de dispositivos actualmente emparejados y agregua a 'pairedDevices'

Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();

// Adiciona un dispositivos previo emparejado al array

if (pairedDevices.size() > 0)

{

for (BluetoothDevice device : pairedDevices) { //EN CASO DE ERROR LEER LA ANTERIOR EXPLICACION

mPairedDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());

}

}

}

// Configura un (on-click) para la lista

private AdapterView.OnItemClickListener mDeviceClickListener = new AdapterView.OnItemClickListener() {

public void onItemClick(AdapterView av, View v, int arg2, long arg3) {

// Obtener la dirección MAC del dispositivo, que son los últimos 17 caracteres en la vista

String info = ((TextView) v).getText().toString();

String address = info.substring(info.length() - 17);

// Realiza un intent para iniciar la siguiente actividad

// mientras toma un EXTRA\_DEVICE\_ADDRESS que es la dirección MAC.

Intent i = new Intent(bluetooth.this, control.class);//<-<- PARTE A MODIFICAR >->->

//Intent i2 = new Intent(dispositivos\_bt.this, Principal.class);

i.putExtra(EXTRA\_DEVICE\_ADDRESS, address);

startActivity(i);

}

};

private void VerificarEstadoBT() {

// Comprueba que el dispositivo tiene Bluetooth y que está encendido.

mBtAdapter= BluetoothAdapter.getDefaultAdapter();

if(mBtAdapter==null) {

Toast.makeText(getBaseContext(), "El dispositivo no soporta Bluetooth", Toast.LENGTH\_SHORT).show();

} else {

if (mBtAdapter.isEnabled()) {

Log.d(TAG, "...Bluetooth Activado...");

} else {

//Solicita al usuario que active Bluetooth

Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(enableBtIntent, 1);

}

}

}

}

**Clase Bluetooth (XML)**

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

<android.support.constraint.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=".bluetooth">

<TextView

android:id="@+id/IdTitulo"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="50dp"

android:layout\_marginTop="16dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<ListView

android:id="@+id/IdList"

android:layout\_width="368dp"

android:layout\_height="460dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="38dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

**Clase Carga (Java)**

package com.example.pst\_carro;

import android.content.Intent;

import android.os.AsyncTask;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.JsonReader;

import android.util.Log;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.Reader;

import java.io.UnsupportedEncodingException;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.ArrayList;

public class carga extends AppCompatActivity {

ArrayAdapter mArrayAdapter;

ListView listView;

TextView textViewPeso, textViewLista;

Button buttonReiniciarLista, buttonCargar;

double pesoLimite = 120;

double pesoCarga = 0;

String id\_usuario;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_carga);

listView = (ListView) findViewById(R.id.listView);

textViewPeso = (TextView) findViewById(R.id.textViewPeso);

textViewLista = (TextView) findViewById(R.id.textViewLista);

buttonCargar = (Button) findViewById(R.id.buttonCargar);

buttonCargar.setEnabled(true);

buttonReiniciarLista = (Button) findViewById(R.id.buttonReiniciarLista);

mArrayAdapter = new ArrayAdapter(this,R.layout.textviewlayout);

listView.setAdapter(mArrayAdapter);

Bundle bundle = getIntent().getExtras();

id\_usuario = bundle.getString("id\_usuario");

new consultarItems().execute("http://172.20.141.9/consultar\_item.php?id\_usuario="+id\_usuario);

buttonCargar.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(carga.this,bluetooth.class);

startActivity(i);

}

});

buttonReiniciarLista.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Toast.makeText(carga.this,"Lista Reiniciada",Toast.LENGTH\_SHORT).show();

}

});

}

private class consultarItems extends AsyncTask<String, Void, String> {

@Override

protected String doInBackground(String... urls) {

// params comes from the execute() call: params[0] is the url.

try {

return downloadUrl(urls[0]);

} catch (IOException e) {

return "Unable to retrieve web page. URL may be invalid.";

}

}

// onPostExecute displays the results of the AsyncTask.

@Override

protected void onPostExecute(String result) {

JSONArray ja = null;

String[] token;

String[] token2;

int length = result.length();

int i;

token=result.split("]");

try {

for (i=0;i<length;i++){

token2=token[i].replace("[","").replace("\"","").split(",");

ja = new JSONArray(token2);

mArrayAdapter.add(ja.getString(0) + "\n" + ja.getString(1));

}

} catch (JSONException e) {

e.printStackTrace();

}

}

}

private String downloadUrl(String myurl) throws IOException {

Log.i("URL",""+myurl);

myurl = myurl.replace(" ","%20");

InputStream is = null;

// Only display the first 500 characters of the retrieved

// web page content.

int len = 500;

try {

URL url = new URL(myurl);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setReadTimeout(10000 /\* milliseconds \*/);

conn.setConnectTimeout(15000 /\* milliseconds \*/);

conn.setRequestMethod("GET");

conn.setDoInput(true);

// Starts the query

conn.connect();

int response = conn.getResponseCode();

Log.d("respuesta", "The response is: " + response);

is = conn.getInputStream();

// Convert the InputStream into a string

String contentAsString = readIt(is, len);

return contentAsString;

// Makes sure that the InputStream is closed after the app is

// finished using it.

} finally {

if (is != null) {

is.close();

}

}

}

public String readIt(InputStream stream, int len) throws IOException, UnsupportedEncodingException {

Reader reader = null;

reader = new InputStreamReader(stream, "UTF-8");

char[] buffer = new char[len];

reader.read(buffer);

return new String(buffer);

}

}

**Clase Carga (XML)**

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

<android.support.constraint.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=".carga">

<ListView

android:id="@+id/listView"

android:layout\_width="360dp"

android:layout\_height="357dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.486"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.019" />

<TextView

android:id="@+id/textViewPeso"

android:layout\_width="373dp"

android:layout\_height="45dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.636"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.791" />

<Button

android:id="@+id/buttonReiniciarLista"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="Reiniciar lista"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.237"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.898" />

<Button

android:id="@+id/buttonCargar"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="Cargar"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.739"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.896" />

<TextView

android:id="@+id/textViewLista"

android:layout\_width="362dp"

android:layout\_height="116dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.545"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.646" />

</android.support.constraint.ConstraintLayout>

**Clase Control (Java)**

package com.example.pst\_carro;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.bluetooth.BluetoothSocket;

import android.content.Intent;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.MotionEvent;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.UUID;

import com.example.pst\_carro.R;

public class control extends AppCompatActivity {

//1)

Button avanzar, retroceder, derecha, izquierda;

TextView IdBufferIn;

//-------------------------------------------

Handler bluetoothIn;

final int handlerState = 0;

private BluetoothAdapter btAdapter = null;

private BluetoothSocket btSocket = null;

private StringBuilder DataStringIN = new StringBuilder();

private ConnectedThread MyConexionBT;

// Identificador unico de servicio - SPP UUID

private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

// String para la direccion MAC

private static String address = null;

public double pesoLimite = 120;

//-------------------------------------------

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_control);

//2)

//Enlaza los controles con sus respectivas vistas

IdBufferIn = (TextView) findViewById(R.id.datos);

avanzar = (Button) findViewById(R.id.avanzar);

retroceder = (Button) findViewById(R.id.retroceder);

derecha = (Button) findViewById(R.id.derecha);

izquierda = (Button) findViewById(R.id.izquierda);

bluetoothIn = new Handler() {

public void handleMessage(android.os.Message msg) {

//Double pesoActual = 0.00;

if (msg.what == handlerState) {

String readMessage = (String) msg.obj;

DataStringIN.append(readMessage);

int endOfLineIndex = DataStringIN.indexOf("#");

if (endOfLineIndex > 0) {

String dataInPrint = DataStringIN.substring(0, endOfLineIndex);

IdBufferIn.setText("Peso de carga actual: " + dataInPrint);//<-<- PARTE A MODIFICAR >->->

//pesoActual=Double.parseDouble(dataInPrint);

//if (pesoActual>pesoLimite){

// Toast.makeText(control.this,"Peso limite superado, descargue el carro",Toast.LENGTH\_LONG).show();

//}

DataStringIN.delete(0, DataStringIN.length());

}

}

}

};

btAdapter = BluetoothAdapter.getDefaultAdapter(); // get Bluetooth adapter

VerificarEstadoBT();

// Configuracion onClick listeners para los botones

// para indicar que se realizara cuando se detecte

// el evento de Click

avanzar.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

if (event.getAction()==MotionEvent.ACTION\_DOWN){

MyConexionBT.write("1");

return true;

}

if (event.getAction()==MotionEvent.ACTION\_UP){

MyConexionBT.write("2");

return true;

}

return false;

}

});

retroceder.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

if (event.getAction()==MotionEvent.ACTION\_DOWN){

MyConexionBT.write("0");

return true;

}

if (event.getAction()==MotionEvent.ACTION\_UP){

MyConexionBT.write("2");

return true;

}

return false;

}

});

derecha.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

if (event.getAction()==MotionEvent.ACTION\_DOWN){

MyConexionBT.write("3");

return true;

}

if (event.getAction()==MotionEvent.ACTION\_UP){

MyConexionBT.write("2");

return true;

}

return false;

}

});

izquierda.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

if (event.getAction()==MotionEvent.ACTION\_DOWN){

MyConexionBT.write("4");

return true;

}

if (event.getAction()==MotionEvent.ACTION\_UP){

MyConexionBT.write("2");

return true;

}

return false;

}

});

}

private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException

{

//crea un conexion de salida segura para el dispositivo

//usando el servicio UUID

return device.createRfcommSocketToServiceRecord(BTMODULEUUID);

}

@Override

public void onResume()

{

super.onResume();

//Consigue la direccion MAC desde DeviceListActivity via intent

Intent intent = getIntent();

//Consigue la direccion MAC desde DeviceListActivity via EXTRA

address = intent.getStringExtra(bluetooth.EXTRA\_DEVICE\_ADDRESS);//<-<- PARTE A MODIFICAR >->->

//Setea la direccion MAC

BluetoothDevice device = btAdapter.getRemoteDevice(address);

try

{

btSocket = createBluetoothSocket(device);

} catch (IOException e) {

Toast.makeText(getBaseContext(), "La creacción del Socket fallo", Toast.LENGTH\_LONG).show();

}

// Establece la conexión con el socket Bluetooth.

try

{

btSocket.connect();

} catch (IOException e) {

try {

btSocket.close();

} catch (IOException e2) {}

}

MyConexionBT = new ConnectedThread(btSocket);

MyConexionBT.start();

}

@Override

public void onPause()

{

super.onPause();

try

{ // Cuando se sale de la aplicación esta parte permite

// que no se deje abierto el socket

btSocket.close();

} catch (IOException e2) {}

}

//Comprueba que el dispositivo Bluetooth Bluetooth está disponible y solicita que se active si está desactivado

private void VerificarEstadoBT() {

if(btAdapter==null) {

Toast.makeText(getBaseContext(), "El dispositivo no soporta bluetooth", Toast.LENGTH\_LONG).show();

} else {

if (btAdapter.isEnabled()) {

} else {

Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(enableBtIntent, 1);

}

}

}

//Crea la clase que permite crear el evento de conexion

private class ConnectedThread extends Thread

{

private final InputStream mmInStream;

private final OutputStream mmOutStream;

public ConnectedThread(BluetoothSocket socket)

{

InputStream tmpIn = null;

OutputStream tmpOut = null;

try

{

tmpIn = socket.getInputStream();

tmpOut = socket.getOutputStream();

} catch (IOException e) { }

mmInStream = tmpIn;

mmOutStream = tmpOut;

}

public void run()

{

byte[] buffer = new byte[256];

int bytes;

// Se mantiene en modo escucha para determinar el ingreso de datos

while (true) {

try {

bytes = mmInStream.read(buffer);

String readMessage = new String(buffer, 0, bytes);

// Envia los datos obtenidos hacia el evento via handler

bluetoothIn.obtainMessage(handlerState, bytes, -1, readMessage).sendToTarget();

} catch (IOException e) {

break;

}

}

}

//Envio de trama

public void write(String input)

{

try {

mmOutStream.write(input.getBytes());

}

catch (IOException e)

{

//si no es posible enviar datos se cierra la conexión

Toast.makeText(getBaseContext(), "La Conexión fallo", Toast.LENGTH\_LONG).show();

finish();

}

}

}

}

**Clase Control (XML)**

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

<android.support.constraint.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=".control">

<TextView

android:id="@+id/datos"

android:layout\_width="221dp"

android:layout\_height="58dp"

android:layout\_marginStart="48dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="Peso de carga actual"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.379"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.738" />

<Button

android:id="@+id/avanzar"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="⇧"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.146" />

<Button

android:id="@+id/derecha"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="⇨"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.785"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.277" />

<Button

android:id="@+id/izquierda"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="⇦"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.211"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.277" />

<Button

android:id="@+id/retroceder"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="⇩"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.403" />

</android.support.constraint.ConstraintLayout>

**Clase Login (Java)**

package com.example.pst\_carro;

import android.content.Intent;

import android.os.AsyncTask;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import org.json.JSONArray;

import org.json.JSONException;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.Reader;

import java.io.UnsupportedEncodingException;

import java.net.HttpURLConnection;

import java.net.URL;

public class login extends AppCompatActivity {

EditText editTextUser;

EditText editTextPassword;

Button buttonLogin;

String user = "defaultuser";

String password = "defaultpassword";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

editTextUser = (EditText) findViewById(R.id.editTextUser);

editTextPassword = (EditText) findViewById(R.id.editTextPassword);

buttonLogin = (Button) findViewById(R.id.buttonLogin);

buttonLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new loginApp().execute("http://172.20.141.9/login.php?id\_usuario="+editTextUser.getText().toString());

if (editTextUser.getText().toString().equals(user)){

if (editTextPassword.getText().toString().equals(password)){

Intent i = new Intent(login.this,carga.class);

i.putExtra("id\_usuario",user);

startActivity(i);

editTextUser.setText("");

editTextPassword.setText("");

user = "defaultuser";

password = "defaultpassword";

finish();

}else{

editTextUser.setText("");

editTextPassword.setText("");

Toast.makeText(login.this,"Credenciales no validas",Toast.LENGTH\_SHORT).show();

}

}else {

editTextUser.setText("");

editTextPassword.setText("");

Toast.makeText(login.this,"Credenciales no validas",Toast.LENGTH\_SHORT).show();

}

}

});

}

private class loginApp extends AsyncTask<String, Void, String> {

@Override

protected String doInBackground(String... urls) {

// params comes from the execute() call: params[0] is the url.

try {

return downloadUrl(urls[0]);

} catch (IOException e) {

return "Unable to retrieve web page. URL may be invalid.";

}

}

// onPostExecute displays the results of the AsyncTask.

@Override

protected void onPostExecute(String result) {

JSONArray ja = null;

try {

ja = new JSONArray(result);

user = ja.getString(0);

password = ja.getString(1);

} catch (JSONException e) {

e.printStackTrace();

}

}

}

private String downloadUrl(String myurl) throws IOException {

Log.i("URL",""+myurl);

myurl = myurl.replace(" ","%20");

InputStream is = null;

// Only display the first 500 characters of the retrieved

// web page content.

int len = 500;

try {

URL url = new URL(myurl);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setReadTimeout(10000 /\* milliseconds \*/);

conn.setConnectTimeout(15000 /\* milliseconds \*/);

conn.setRequestMethod("GET");

conn.setDoInput(true);

// Starts the query

conn.connect();

int response = conn.getResponseCode();

Log.d("respuesta", "The response is: " + response);

is = conn.getInputStream();

// Convert the InputStream into a string

String contentAsString = readIt(is, len);

return contentAsString;

// Makes sure that the InputStream is closed after the app is

// finished using it.

} finally {

if (is != null) {

is.close();

}

}

}

public String readIt(InputStream stream, int len) throws IOException, UnsupportedEncodingException {

Reader reader = null;

reader = new InputStreamReader(stream, "UTF-8");

char[] buffer = new char[len];

reader.read(buffer);

return new String(buffer);

}

}

**Clase Login (XML)**

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

<android.support.constraint.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=".login">

<EditText

android:id="@+id/editTextUser"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:ems="10"

android:hint="Usuario"

android:inputType="number"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.246" />

<EditText

android:id="@+id/editTextPassword"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:ems="10"

android:hint="Contrasena"

android:inputType="numberPassword"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.391" />

<Button

android:id="@+id/buttonLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="Iniciar sesion"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.535" />

</android.support.constraint.ConstraintLayout>