UNIVERSIDAD POLITÉCNICA DE TULANCINGO

ORTIZ OSORIO CRISTIAN URIEL
FRANCISCO MELO MARIO ETZAEL

U P T

Ingeniería en Sistemas Computacionales

Conexión de arduino para consumir Servicios RestFul

Contents

Código Arduino

```
/*********
Hecho por Mario Etzael, Cristian Uriel, Jose Juan, Gerardo
Ingenieria en Sistemas Computacionales
Grupo 72
Mtro. Uriel Edgardo Escobar
*********/

// Import required libraries
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
// Replace with your network credentials
//String ip = "http://192.168.1.145:8080/restfull/public/api/";
String ip = "http://192.168.114.246/Arduino/public/api/";
HTTPClient http;

//Remplazar con la ssid de su red y contraseña
const char* ssid = "UPT";
const char* password = "12345678";
```

```
#define DHTPIN 5     // Digital pin connected to the DHT sensor
#define DHTTYPE    DHT11     // DHT 11

DHT dht(DHTPIN, DHTTYPE);

// current temperature & humidity, updated in loop()
float t = 0.0;
float h = 0.0;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;    // will store last time DHT was updated

// Updates DHT readings every 10 seconds
const long interval = 3000;

void setup() {
  // Serial port for debugging purposes
  Serial.begin(115200);
  dht.begin();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  pinMode(DHTPIN, OUTPUT);
  digitalWrite(DHTPIN, LOW);
  Serial.println("Connecting to WiFi");
```

```
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }
  pinMode(BUILTIN_LED, OUTPUT);
  digitalWrite(BUILTIN_LED, LOW);

  // Print ESP8266 Local IP Address
  Serial.println(WiFi.localIP());
}

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    // save the last time you updated the DHT values
    previousMillis = currentMillis;
    // Read temperature as Celsius (the default)
    float newT = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    //float newT = dht.readTemperature(true);
    // if temperature read failed, don't change t value
    if (isnan(newT)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      t = newT;
      Serial.println(t);
    }
    // Read Humidity
```

```
  float newH = dht.readHumidity();
  // if humidity read failed, don't change h value
  if (isnan(newH)) {
    Serial.println("Failed to read from DHT sensor!");
  }
  else {
    h = newH;
    Serial.println(h);
  }

  String postData = "Temperatura=" + String(t) + "&Humedad=" + String(h);
  http.begin(ip + "editT/1?");         //Specify request destination
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");    //Specify content-type header
  int httpCode = http.PUT(postData);   //Send the request
  String payload = http.getString();    //Get the response payload
  Serial.println(payload);

  //Consulta de Motor

  http.begin(ip + "motor"); //Specify request destination
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");    //Specify content-type header
  httpCode = http.GET();   //Send the request
  String estado = http.getString();    //Get the response payload
  if (estado == "1")
  {
    digitalWrite(BUILTIN_LED, LOW);
  } else
```

```
  {
    digitalWrite(BUILTIN_LED, HIGH);
  }

 }
}
```

Código de Aplicación móvil (MainActivity.java)

```
package com.example.arduino;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {
    RequestQueue requestQueue;
    Button btnOn;
    Handler h=new Handler();
    TextView txtTemp, txtHume;
    String hume="0", temp="0";
```

```
    Thread m, b;
    private static int p=0;
    Boolean tr=true;

    private static String
ip="http://187.141.55.141:10580/ProyectosAlumnos/Arduino/";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtTemp=(TextView)findViewById(R.id.txtTemp);
        txtHume =(TextView)findViewById(R.id.txtHume);
        btnOn=(Button)findViewById(R.id.btnOn);
        btnOn.setEnabled(false);
        m=new Thread(new Runnable() {
            @Override
            public void run() {
                while (true) {
                    h.post(new Runnable() {
                        @Override
                        public void run() {
                            HistorialT(ip+"TempHume.php");
                           //
HistorialT("http://"+ip+"/Android/TempHume.php");
                        }
                    });
                    try {
                        Thread.sleep(1000);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                    if (Float.parseFloat(temp)>=30.00){
                        tr=false;
                        p=0;
                    }else{
                        tr=true;
                    }
                }
            }
        });
        b=new Thread(new Runnable() {
            @Override
            public void run() {
                while (true) {
                    h.post(new Runnable() {
                        @Override
                        public void run() {
                            if (tr==false){
                                btnOn.setEnabled(true);
                            }else{
                                btnOn.setEnabled(false);
                                if (p==0) {
                                    Actualizar(ip+"Motor.php", "0");
                                    p=1;
                                }
                            }
                        }
```

```
                    });
                    try {
                        Thread.sleep(1000);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        });
        m.start();
        b.start();

        btnOn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Actualizar(ip+"Motor.php", "1");
            }
        });
    }
    public void HistorialT (String URL){

        JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(URL, new
Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                JSONObject jsonObject = null;
                for (int i = 0; i < response.length(); i++) {
                    try {
                        jsonObject = response.getJSONObject(i);
                        temp= jsonObject.getString("Temperatura");
                        hume=jsonObject.getString("Humedad");
                        txtHume.setText(hume+" %");
                        txtTemp.setText(temp+" °C");
                    } catch (JSONException e) {
                        Toast.makeText(getApplicationContext(),
e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(getApplicationContext(), error.toString(),
Toast.LENGTH_SHORT).show();
            }
        }
        );
        requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(jsonArrayRequest);
    }
    private void Actualizar(String URL, final String state){
        StringRequest stringRequest=new StringRequest(Request.Method.POST,
URL, new Response.Listener<String>() {
            @Override
            public void onResponse(String s) {
                Toast.makeText(getApplicationContext(), "Actualizado
Correctamente", Toast.LENGTH_SHORT).show();
```

```
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError volleyError) {
                Toast.makeText(getApplicationContext(),
volleyError.toString(),Toast.LENGTH_LONG).show();
            }
        }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError
{
                Map<String,String> par=new HashMap<String, String>();
                par.put("estado", state);
                return par;
            }
        };
        RequestQueue requestQueue= Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }
}
```

Código de Aplicación móvil (activity-main.xml)

```
<RelativeLayout android:layout_width="match_parent"
android:layout_height="match_parent" android:orientation="vertical"
    android:id="@+id/relativeL"
    xmlns:android="http://schemas.android.com/apk/res/android"
android:background="#A79DA4">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Lector de Temperatura"
        android:layout_margin="30dp"
        android:textSize="30dp"
        android:textAlignment="center"
        android:textColor="#000"
        />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Temperatura"
        android:textSize="30dp"
        android:textAlignment="center"
        android:textColor="#000"
        android:layout_marginTop="120dp"
        />
    <ImageView
        android:layout_width="100dp"
        android:layout_height="150dp"
        android:src="@drawable/temperatura"
        android:layout_marginTop="190dp"
```

```
        android:layout_marginLeft="40dp"
        />
    <TextView
        android:id="@+id/txtTemp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="20 °C"
        android:textColor="#000"
        android:layout_marginLeft="200dp"
        android:layout_marginTop="240dp"
        />
    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/humedad"
        android:layout_marginTop="360dp"
        android:layout_marginLeft="40dp"
        />

    <TextView
        android:id="@+id/txtHume"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="20 %"
        android:textColor="#000"
        android:layout_marginLeft="200dp"
        android:layout_marginTop="390dp"
        />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="500dp"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:background="#E7E5E0"
        android:id="@+id/btnOn"
        android:text="Encender Ventilador"
        />
</RelativeLayout>
```

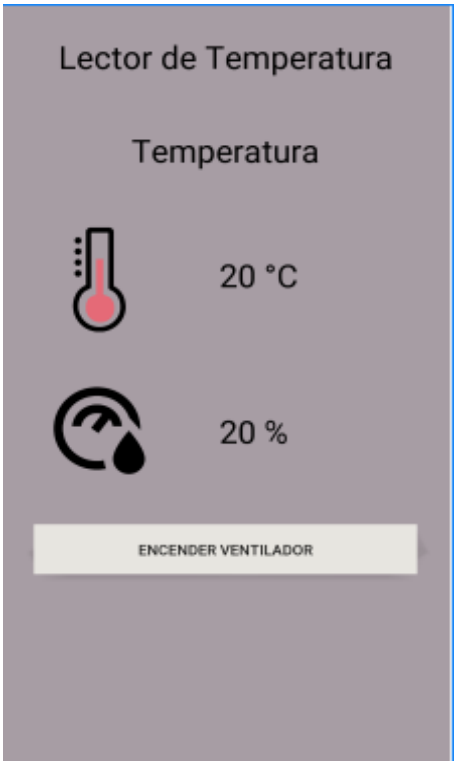
Código de Aplicación móvil (AndroidManifest.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.arduino">
    <uses-permission android:name="android.permission.INTERNET" />
    <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=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

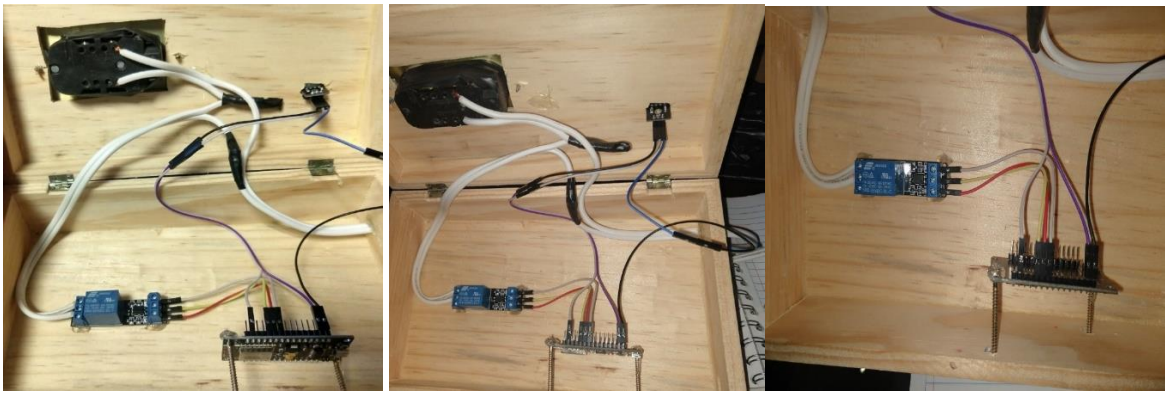
Interfaz de la aplicación móvil.

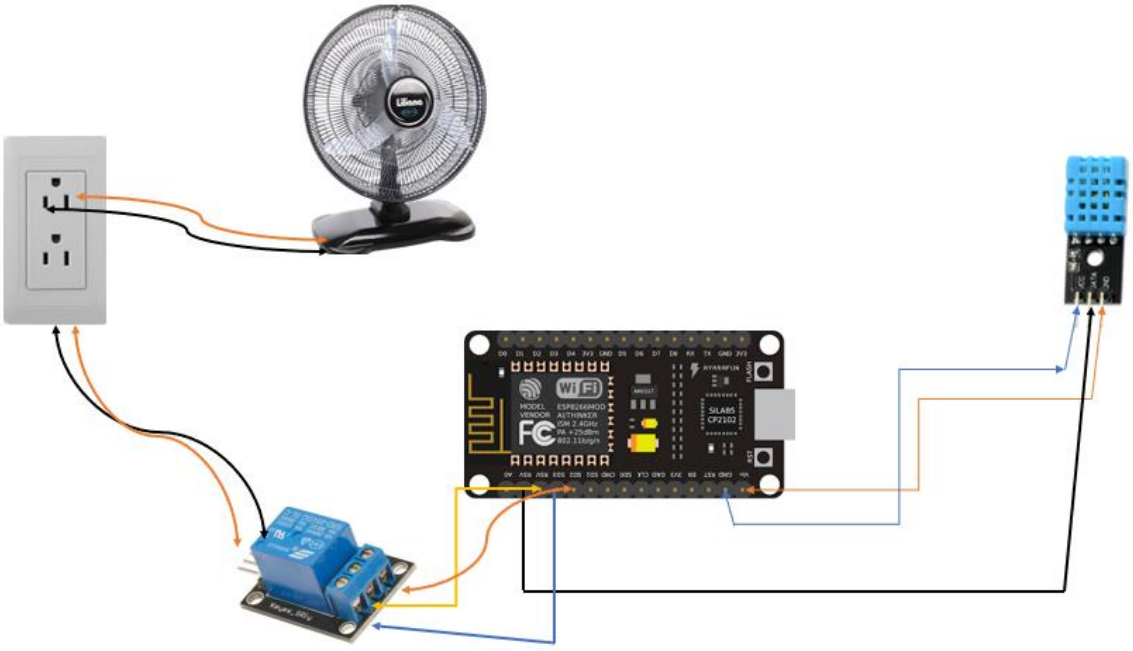


Función de la aplicación móvil.

En la aplicación se actualiza cada segundo la temperatura la cual se va ah mostrar en el respectivo espacio, cuando la temperatura llega ah 30° se activará el botón que realizará la activación del paso de corriente hacia el sistema de ventilación mediante el Arduino.

Diagrama de conexión de Arduino.





Materiales.

- Arduino
- Sensor DHT11
- Clavija
- Socket
- 2 metros de cable
- 10 Jumpers
- Silicon
- Cinta de aislar