

Estación Meteorológica: Practica 1 (20 agosto 2021)

Grupo 6

201700355 Diego Alejandro Martinez Garcia
 201807100 Daniel Alejandro Barillas Soberanis
 201800984 Alex Fernando Méndez López
 201806838 Elian Saúl Estrada Urbina
 201504220 José Andres Rodas Arrecis

Resumen— La estación meteorológica tiene las siguientes funciones: lectura de la velocidad del viento, lector de humedad lector de temperatura, lector de direcciones en donde aplica NORTE, SUR, ESTE, OESTEM; y medir la cantidad de luz, teniendo esas funciones se puede registrar en una base de datos utilizando un Arduino MEGA que será quien haga la lectura de todos esos dispositivos que conforman la estación meteorológica, la base de datos utilizada es MySQL la cual estará presentando un papel importante para el proyecto, así mismo el Arduino estará enviando información por medio de una API a la base de datos y esta a su vez las enviara a un cliente web en donde se presentara de forma ordenada los datos obtenidos en el suceso actual del aparato en turno.

I. INTRODUCCIÓN

En la actualidad, el tener conocimiento acerca de los estados climáticos es muy útil, tanto como para saber si conviene realizar ciertas actividades al aire libre como también para saber si conviene mejor utilizar un espacio cerrado, por ello la implementación de una estación meteorológica nos ayuda a solventar esta necesidad, tanto a nivel personal como a un nivel mayor. Dicha estación recopilará información (temperatura, humedad, velocidad, cantidad de luz y dirección del viento) del ambiente en tiempo real y lo almacenará, para, posteriormente y con la ayuda de una aplicación web poder realizar graficas para representar dicha información, sacar conclusiones y tomar decisiones en base a ellas.

II. DESARROLLO DE LA PRÁCTICA

Los datos que son necesarios medir son los siguientes: temperatura, humedad, dirección del viento, cantidad de la luz en el entorno y velocidad del viento, para después ser analizados y representados gráficamente.

A. Infraestructura del Producto

Listado de Materiales

- Listado de Materiales físicos
 - o Arduino Mega
 - o Bluetooth
 - o Cables
 - o Esferas navideñas
 - o Plástico
 - o Silicon
 - o Madera
 - o Imán
- Listado de Materiales digitales
 - o AngularJS – cliente de la aplicación.
 - o Mqtt – protocolo de comunicación.
 - o NodeJS – servidor para almacenar la información.
 - o MySQL – DBMS.

Boceto del Prototipo

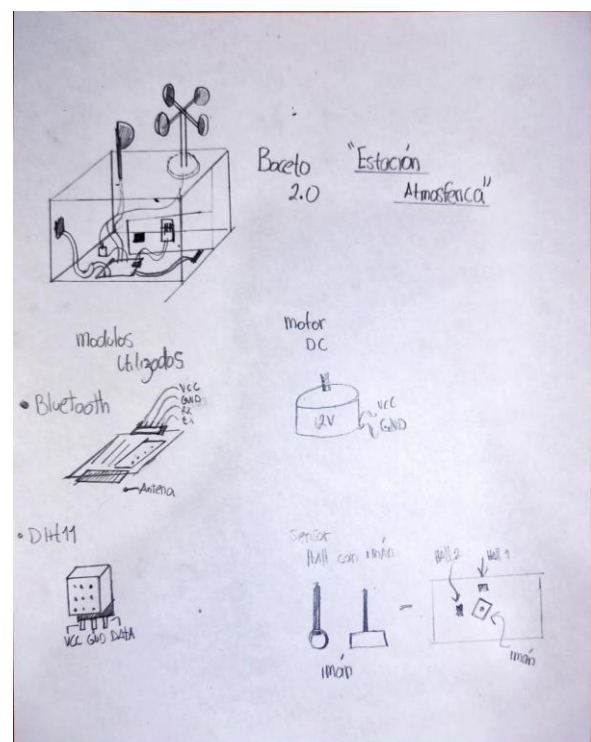


Fig. 1. Boceto de la estación meteorológica.

Esta práctica recibió el apoyo económico de:

- Diego Alejandro Martínez García
- Daniel Alejandro Barillas Soberanis
- Alex Fernando Méndez López
- Elian Saúl Estrada Urbina
- José Andrés Rodas Arrecis

B. Sensores

- Dirección del Viento:

Sensor efecto Hall Digital OH3144

TABLA I
CARACTERISTICAS DEL COMPONENTE

| Corriente de Salida | Voltaje | Tipo de Sensor | Polo censado |
|---------------------|---------|----------------------|--------------|
| 25 mA | 4~24V | Digital, tipo switch | Polo sur |

- Velocidad del Viento

Motor DC

TABLA II
CARACTERISTICAS DEL COMPONENTE

| Tamaño | Voltaje | Velocidad sin carga | Corriente sin carga |
|---------------|---------|---------------------|---------------------|
| 24.5 x 12.5mm | DC3V-6V | 3500-4500 rpm/min | 25mA |

- Temperatura y Humedad

Módulo DHT11

TABLA III
CARACTERISTICAS DEL COMPONENTE

| Voltaje | Corriente | Rango Temperatura | Rango Humedad |
|----------|-----------|-------------------|---------------|
| 3.5~5VDC | 2.5mA | 0~50° ± 2° C | 20~90% RH |

- Cantidad de Luz

Foto-resistor KY-018

TABLA IV
CARACTERISTICAS DEL COMPONENTE

| Tamaño | Voltaje | Peso | Tipo de salida |
|----------------|---------|------|----------------|
| 20 x 17 X 8 mm | 3.3~5V | 1 gr | Analógica |

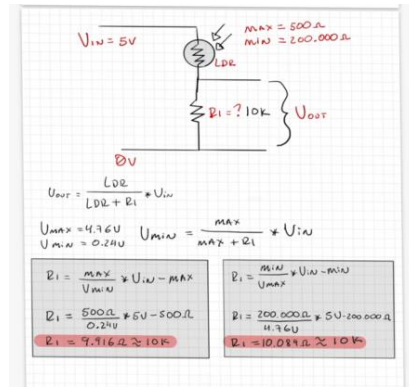


Fig. 2. Cálculo de la resistencia para el sensor de luz.

C. Conectividad

- Entorno del Objeto

INTemperie

La estación meteorológica debe permanecer en la intemperie para poder recolectar información del ambiente.

- Tamaño del Objeto

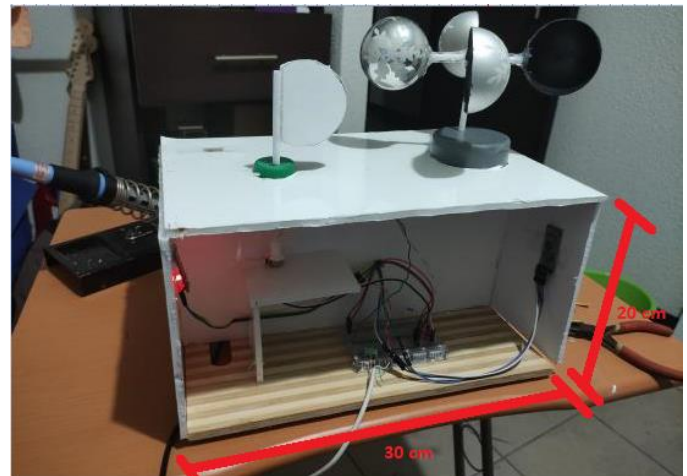


Fig. 3. Tamaño de la estación

- Comunicación Arduino – Servidor

○ Enviar datos al serial

```

if (miBT.available()) // si hay informacion disponible desde modulo
  Serial.write(miBT.read()); // lee Bluetooth y envia a monitor serial de Arduino

if (Serial.available()) // si hay informacion disponible desde el monitor serial
  miBT.write(Serial.read()); // lee monitor serial y envia a Bluetooth
  miBT.println(getJSON( humidity, temperature, temperatureF, windDirection, windSpeed, illum));
  delay(3000);
}

```

Fig. 4. Datos al serial

○ Leer datos del serial

```
const mqtt = require('mqtt');
const serialPort = require('serialport');
// const readline = require('@serialport/parser-readline');

const publisher = mqtt.connect("wss://my-broker-acye2.herokuapp.com");

const usb_port = 'COM5';
const bauds = 9600;

const port = new serialPort(usb_port, { baudRate: bauds });
const parser = port.pipe(new serialPort.parsers.Readline({ delimiter: "\n" }));

const onConnect = () =>{
  parser.on('data', (data) => {
    console.log(data);
    publisher.publish("ACYE2_G6/data", data);
  });
}

port.on('open', () => { console.log("Se abrió la comunicación"); });
publisher.on('connect', onConnect);
```

Fig. 5. Leyendo datos del serial.

○ Almacenar Datos

```
const mqtt = require('mqtt');
const sub = mqtt.connect("wss://my-broker-acye2.herokuapp.com");
const DB = require('.../config/configdb');

const onMessage = (topic, message) => {
  if (message.toString().includes('temperature')) {
    let data = JSON.parse(message.toString());
    const sql = `
      INSERT INTO status
      (temperature, temperaturef, humidity, windspeed, winddirection, brightness, date_time)
      VALUES
      (${data.temperature}, ${data.temperaturef}, ${data.humidity}, ${data.windspeed}, \"${data.winddirection}\", ${data.luminosidad}, NOW());
    `;
    DB.query(sql, function(error, results) {
      if (error) console.log(error);
      console.log("Id ", results.insertId, " inserted");
    });
  } else return;
}

sub.on('connect', () => { sub.subscribe("ACYE2_G6/data"); });
sub.on('message', onMessage);

module.exports = onMessage
```

Fig. 6. Almacenando datos en la DB

- Comunicación Servidor – Base de Datos

wss://my-broker-acye2.herokuapp.com/ACYE2_G6/data

- Diagrama de comunicación

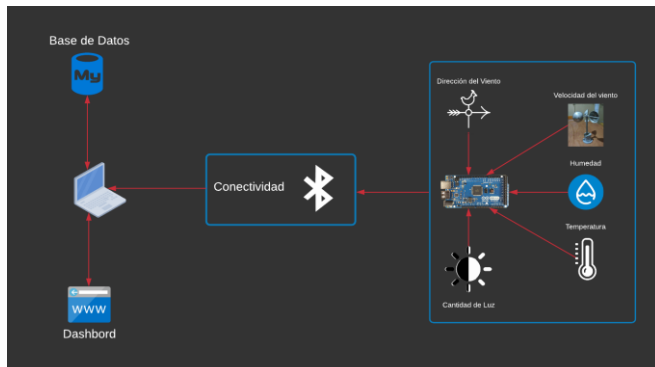


Fig. 7. Diagrama de Comunicación

D. Analíticas

Base de Datos

TABLA V
ESPECIFICACIÓN DE LOS ATRIBUTOS DE LA BASE DE DATOS

| Atributo | Tipo de Dato |
|------------------|--------------|
| Id_status | Int |
| Temperatura | Float |
| Temperaturaf | Float |
| Humedad | Float |
| Velocidad_viento | Float |
| Direccion_viento | Varchar(5) |
| Luminosidad | Int |
| Fecha | datetime |

Reportes

○ Temperatura



Fig. 8. Muestra la tendencia de la temperatura por día del mes.

○ Velocidad del Viento

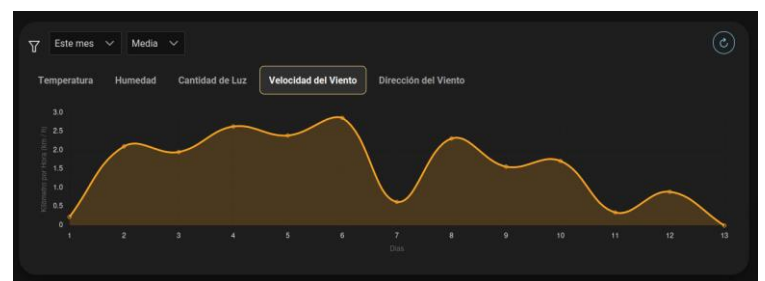


Fig. 9. Indica la velocidad del viento por día del mes.

○ Humedad

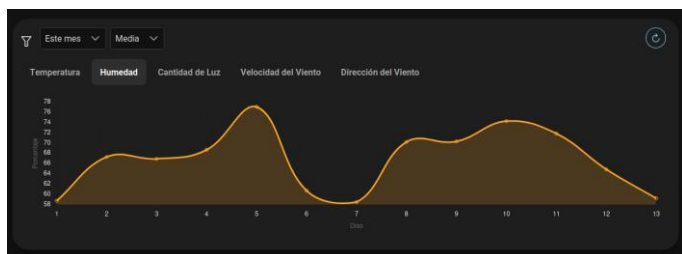


Fig. 10. Indica la humedad por día del mes.

○ Dirección del Viento

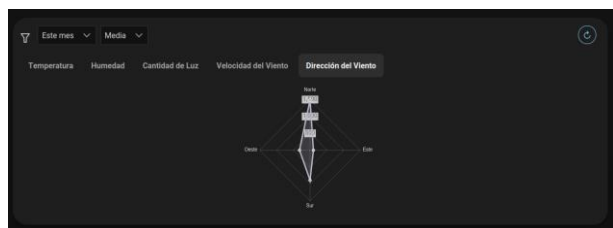


Fig. 11. Muestra la tendencia de la dirección del viento.

○ Cantidad de Luz

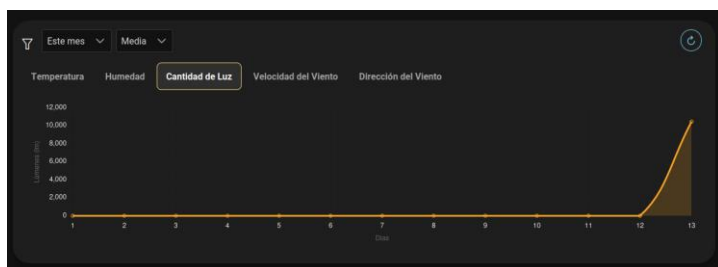


Fig. 12. Luminocidad del mes.

E. Aplicación Inteligente



Fig. 13. Dashbord de la aplicación que analiza el clima.

F. Repositorio

https://github.com/Eq-Maravilla-Dinamita-Lobo-1/ACE2_2S21_G6_Practica2

REFERENCIAS

Web Sites:

- [1] Arduino. Documentación de Arduino. <https://www.arduino.cc/en/Guide>
- [2] Bitwise Ar. (1 de Junio 2019). Magnetómetro 3 ejes Brújula digital QMC5883L bus I2C, <https://www.youtube.com/watch?v=Q31mkKY1iM8&t=1124s>.
- [3] Nick Hehr, reconbot. (Junio 2021). SerialPort. <https://www.npmjs.com/package/serialport>.
- [4] Processing. Documentación de Processing. <https://processing.org>.