DOCUMENTACION:
Practica 1: Estación Meteorológica

Gustavo Omar Perez, Luis Roberto Rivera, Jose Diego Perez, Marlon Abraham Fuentes, Fabio André Sánchez integrantes del grupo 8 del laboratorio de Arquitectura de Computadores y Ensambladores 2.

I. INDRODUCCION

Muchas aplicaciones de IoT nos permiten automatizar recursos y procesos en empresas como actividades domésticas en casa y en otras situaciones. Esto está directamente relacionado con una serie de avances tecnológicos que da la posibilidad de transformar objetos cotidianos en dispositivos electrónicos es su característica fundamental y Guatemala no es la excepción ya que día con día se pretende tener una mejor proyección de aplicación de IoT. En el siguiente documento se explica sobre el desarrollo de un dispositivo de estación meteorológica con la capacidad de medir y reportar las distintas magnitudes relacionadas al análisis del tiempo que soporte las condiciones a la intemperie del variante clima del país.

II. BOCETOS DEL PROTOTIPO

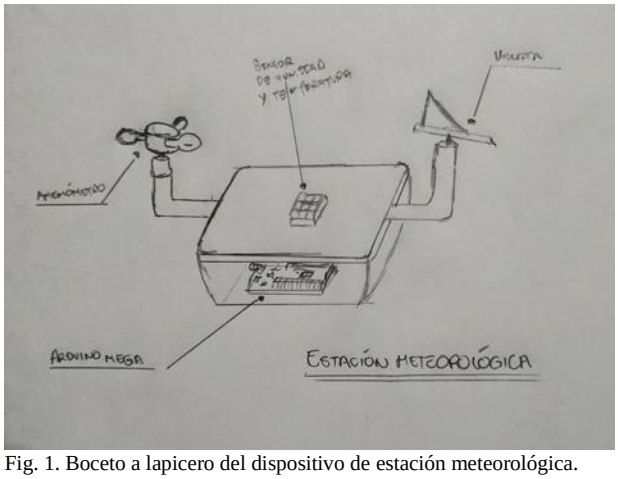

Fig. 1. Boceto a lapicero del dispositivo de estación meteorológica.

Para el dispositivo de estación meteorológica se pretende que el diseño cuente con un anemómetro para la medición de la velocidad del viento en un punto y en un instante determinado, este contara con una hélice que rota sobre una “b” y se enlaza en la parte superior del dispositivo por un cableado largo y del otro lado del dispositivo, para determinar la dirección del viento, se usara una veleta que consta de una placa que gira libremente eje sobre el que está colocado un señalador que indicara la velocidad del viento. Que tendrá la indicación sobre los puntos cardinales en una cruz horizontal que se sitúa un poco más abajo del señalador.

Contará con una abertura en la parte superior donde se ubicará el sensor de humedad y temperatura hechos, el dispositivo se pretende realizar con materiales resistentes, ya que estarán expuestos a situaciones meteorológicas de alto riesgo por lo que debe garantizar su óptimo funcionamiento.

Finalmente, toda la información recopila a través de los sensores se enviará a través del puerto serial y un cableado desde la placa de Arduino Mega hacia la computadora para su posterior análisis.

III. SENSORES

Para la obtención de los datos de temperatura y humedad se utilizó el sensor con digital DHT11 con alta fiabilidad y estabilidad que utiliza un pin digital para enviar la información a través de Arduino.

Los pines del sensor DHT11 son:

- GND: conexión con tierra
- DATA: transmisión de datos
- VCC: alimentación

Para la lectura del sensor es necesario instalar e importar la librería DHT y definir el pin donde se conecta el sensor, la ventaja de la librería es de que nos proporciona datos en grados centígrados y grados Fahrenheit, y únicamente llamando a las funciones correspondientes en este caso se utilizaron readHumidity y readTemperature.

```
DHT11
// Incluimos libreria
#include <DHT.h>

// Definimos el pin digital donde se conecta el sensor
#define DHTPIN 2
// Dependiendo del tipo de sensor
#define DHTTYPE DHT11

// Inicializamos el sensor DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // Inicializamos comunicación serie
  Serial.begin(9600);

  // Comenzamos el sensor DHT
  dht.begin();

}

void loop() {
    // Esperamos 5 segundos entre medidas
  delay(3000);

  // Leemos la humedad relativa
  float h = dht.readHumidity();
  // Leemos la temperatura en grados centigrados (por defecto)
  float t = dht.readTemperature();
  // Leemos la temperatura en grados Fahreheit
  float f = dht.readTemperature(true);

  // Comprobamos si ha habido algún error en la lectura
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Error obteniendo los datos del sensor DHT11");
    return;
  }
```

Fig. 2. Screenshot de código utilizado para la lectura del sensor de humedad y temperatura escrito en C y con el IDE de Arduino.

IV. BASE DE DATOS

Para el almacenamiento de datos de manera persistente se utilizó el sistema de base de datos no SQL mongoDB, que permite adaptarlo a nuestras necesidades por su versatilidad y sobre todo por su velocidad, rendimiento y funcionalidad, siendo esto de las mejores opciones para el procesamiento de datos en tiempo real.

Para el almacenamiento en la base de datos se utilizo documentos, colecciones y a su vez clave-valor, por ejemplo, para la representación de la dirección el viento se utilizó la siguiente estructura

{
“temperatura”: “Valor de la temperatura obtenida por el sensor”,
“humedad”: “Valor de la humedad obtenida por el sensor”,
“velocidad”: “Velocidad del viento obtenida por el sensor”,
“dirección”:” Dirección de los puntos cardinales(N-S-E-O)”
}

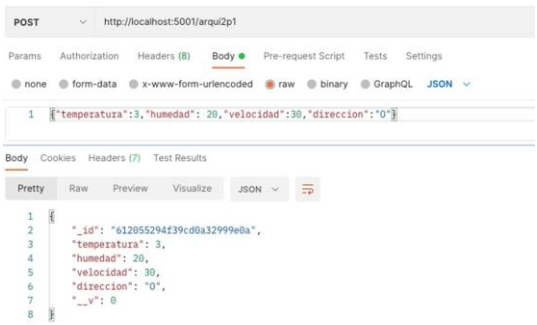

Fig. 3. Ejemplo de colección de humedad y temperatura.

V. BACKEND

Para el lado del servidor se utilizó Node con el framework express por su facilidad de crear servidores en tiempo record y con la facilidad de comunicarse con la base de datos mongoDB, esta base de datos se utilizó para la escritura y lectura de datos recopilados a través de Arduino y el dispositivo de estación meteorológica.

```
server.js
require('dotenv').config()

const express = require('express')
const SerialPort = require('serialport')
const Readline = require('@serialport/parser-readline')
const app = express()
const mongoose = require('mongoose')
const Subscriber = require('./models/arqui2p1')

//mongoose.connect("mongodb://sopes:sopes123@localhost:27017/subscribers", { useNewUrlParser: true })
//mongoose.connect("mongodb://192.168.1.8:27017/iotgl", { useNewUrlParser: true })

const mongoUserName = "DBuser";
const mongoPassword = "DBpassword";
const mongoHost = "localhost";
//const mongoHost = "35.83.237.94";
const mongoPort = "27017";
const mongoDatabase = "arqui2p1";



var uri = "mongodb://" + mongoUserName + ":" + mongoPassword + "@" + mongoHost + ":" + mongoPort + "/" + mongoDatabase;
```

Fig. 4. Screenshot de código realizado para el servidor que muestra la creación del servidor con express y la conexión con la base de datos mongoDB, escrito en JavaScript.

VI. FRONTEND

Con el objetivo de poder visualizar y evaluar la medición de las magnitudes en tiempo real, se desarrolló una aplicación en Processing, con el fin de representar de forma gráfica todos los datos que sean recolectados.

Para la representación de la dirección del viento se un círculo que se mueve en dirección hacia los puntos cardinales (N-S-E-O) según sea la lectura obtenida por el dispositivo de estación meteorológica.

Fig. 5. Ejemplo sobre representación de la dirección del viento, en este caso del oeste.

Para la representación de la temperatura del ambiente en un instante se utilizó un círculo que varía de color según la tabla de referencia de las temperaturas y el color que ofrecen.

Temperatura	Color
2.000 °C	Rojo
2.800 °C	Anaranjado
3.200 °C	Amarillo
4.000 °C	Amarillo claro
5.000 °C	Marfil
5.500 °C	Blanco
6.000 °C	Verdoso
6.500 °C	Azulado
7.300 °C	Azul
9.000 °C	Azul intenso

Fig. 6. Tabla de referencia de las temperaturas y el color que ofrecen.

La representación quedaría de la siguiente manera:

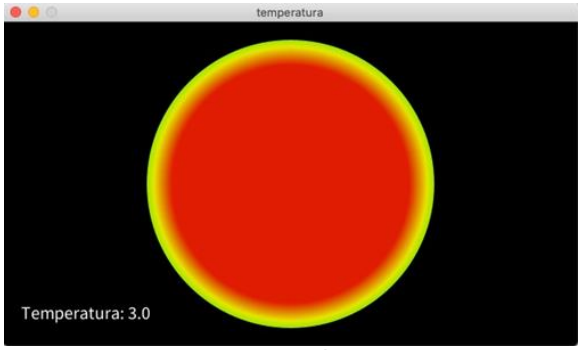

Fig. 7. Ejemplo sobre la representación de la lectura de la temperatura de 3.0

Para la representación de la humedad se utilizaron usa serie de partículas rojizas que representa el porcentaje de humedad del entorno en un instante determinado.

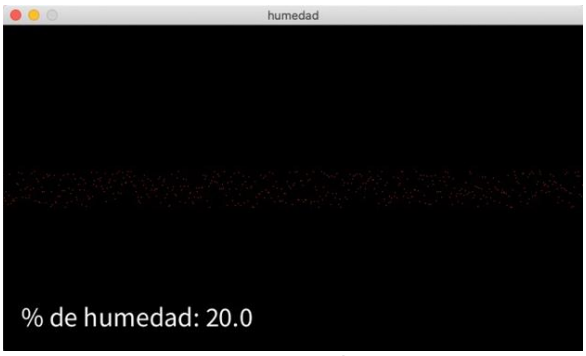

Fig. 8. Ejemplo sobre la representación de la lectura de la humedad en el ambiente del 20%.

VII. CAPAS DEL FRAMEWORK DE IOT

VIII. REPOSITORIO

Link: https://github.com/gopl1495/ACE2_2S21_G8.git