

# Sistema de banco de dados e página HTML para estação meteorológica utilizando a plataforma Arduino

## Database system and HTML page for weather station using the Arduino platform

Hudson Taylor Perrut Cassim IFPR – Campus Goioerê <a href="mailto:HUDSONPERRUT18@GMAIL.COM">HUDSONPERRUT18@GMAIL.COM</a>	Nathan Sanches IFPR – Campus Goioerê <a href="mailto:nathansanches890@gmail.com">nathansanches890@gmail.com</a>	Roberto César Soltoski IFPR – Campus Goioerê <a href="mailto:roberto.soltoski@ifpr.edu.br">roberto.soltoski@ifpr.edu.br</a>	Caroline Peres Martins IFPR – Campus Goioerê <a href="mailto:caroline.martins@ifpr.edu.br">caroline.martins@ifpr.edu.br</a>
---	---	---	---

### Resumo

*A análise dos dados climáticos é de suma importância para o desenvolvimento da economia brasileira, visto que esta é subsidiada pela agricultura. De tal modo, este trabalho objetiva desenvolver uma mini estação meteorológica para coleta de dados climáticos situada no Campus do Instituto Federal de Goioerê. Além da coleta dos dados será feita a inserção destas informações em um banco de dados e a criação de uma página HTML para visualização destes. A estação é produzida com baixos custos e é capaz de coletar dados a respeito da umidade, temperatura, pressão, direção e velocidade do vento, da região de Goioerê (Paraná). Os elementos são coletados por uma placa controladora Arduino Mega e são transmitidos por meio de um módulo Ethernet. O que se espera é que as informações coletadas nesse estudo e a ferramenta desenvolvida possam auxiliar na análise climática do município.*

**Palavras-Chave:** Arduino, Banco de Dados, Estação Meteorológica.

### Abstract

*The analysis of climate data is of paramount importance for the development of the Brazilian economy, since it is subsidized by agriculture. In this way, this work aims to develop a mini meteorological station for climate data collection located on the Campus of the Federal Institute of Goioerê. In addition to data collection, this information will be inserted into a database and an HTML page will be created for viewing. The station is produced at low cost and is capable of collecting data on humidity, temperature, pressure, direction and wind speed in the region of Goioerê (Paraná). The elements are collected by an Arduino Mega controller board and are transmitted through an Ethernet module. What is expected is that the information collected in this study and the tool developed can help in the climate analysis of the municipality.*

**Keywords:** Arduino, Database, Meteorological Station.

## 1 Introdução

A meteorologia, segundo Ynoue (2017), é o estudo de processos que ocorrem na atmosfera terrestre e o resultado desses processos interferem no campo das Ciências ambientais. É necessário também pontuar que a meteorologia não é somente dados acerca da previsão do tempo, ainda de acordo com o autor, a partir do momento em que se é observado os dados climáticos em um determinado período de tempo de uma região pode-se obter informações importantes da localidade como o aumento de temperatura, diminuição de chuvas entre outros. Com essas informações, novos estudos podem surgir, a fim de ser analisado as consequências desses fatos. Além disso, a meteorologia é utilizada para análises climáticas no âmbito da agricultura.

Diante desse fato, a região de Goioerê, assim como em grande parte dos interiores dos estados do Brasil, tem sua economia baseada na agricultura. De acordo com o estudo do CEPEA (Centro de Estudos Avançados em Economia), (2020), o agronegócio, no ano de 2020, possuía a maior porcentagem no PIB nacional, 26.6% o que corresponde a cerca de R\$ 2 trilhões. Esses dados demonstram o quão importante o agronegócio é para o desenvolvimento do país, porém ainda se faz necessário o desenvolvimento de estações meteorológicas, para corroborar com o desenvolvimento regional dos interiores dos estados.

Segundo o Instituto Nacional de Meteorologia (INMET), (2020), existem somente cerca de 572 estações meteorológicas catalogadas em todo o território nacional. Portanto é necessário a instalação de mais dispositivos meteorológicos para tornar as interpretações dos dados mais precisas, se adequando a cada região.

Ainda com relação à Goioerê, é válido pontuar que a estação meteorológica mais próxima se encontra a 60km, na cidade de Umuarama, dessa forma, os dados obtidos sofrem variações de um determinado local para outro. Isso dificulta a vida dos produtores rurais e habitantes da localidade, os quais buscam os dados a respeito da temperatura, umidade, pressão e, por fim, medidas anemométricas em fontes distantes de nossa região, podendo ocorrer a variação para o local de onde é acessado a informação, dessa forma, com uma estação mais próxima poderia receber os dados com maior precisão, devido a pouca distância.

Com base nisso, o presente trabalho teve por objetivo geral o desenvolvimento de uma mini estação meteorológica instalada no Campus do Instituto Federal de Goioerê. A mini

estação meteorológica coleta informações de forma automática sobre a temperatura, umidade, pressão, direção do vento e velocidade do vento. Depois de coletado os dados dos sensores são enviados para um banco de dados.

## 1.1 Objetivos específicos

- Desenvolver uma página HTML de exibição dos dados;
- Fazer testes que comprovam a eficiência da estação;

## 2 Área de instalação da estação meteorológica

A instalação da estação meteorológica será realizada na cidade de Goioerê, na região noroeste do estado do Paraná. Este município se encontra no terceiro planalto paranaense, conforme divisão feita por Maack (2002).

Figura 1 - Paraná Divisão Climática



Fonte: IAPAR, Instituto de Desenvolvimento Rural do Paraná<sup>1</sup>

A respeito da questão climática, Goioerê está na divisão climática Cfa de Koeppen (1900). Ainda segundo o autor, a divisão Cfa é entendida como sendo um clima subtropical úmido que não possui uma estação seca definida, há também a ocorrência de chuva durante todo

---

<sup>1</sup> Disponível em: < <http://www.geografia.seed.pr.gov.br/modules/galeria/detalhe.php?foto=367&evento>>. Acesso em: 15 jan. 2022

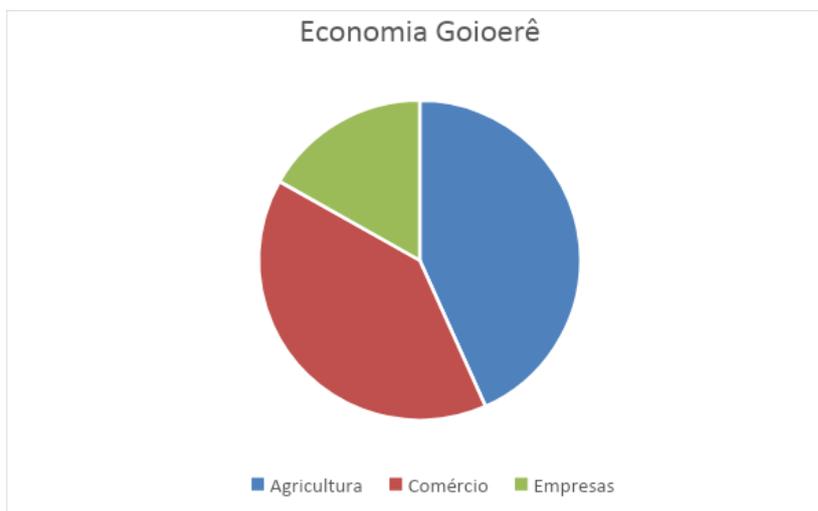
o ano e as temperaturas são acima de 22° C no verão e abaixo de 18° C no inverno. Há ocorrência desse clima nas regiões Centro-Norte, Oeste e Sudeste do Estado.

A respeito da umidade, conforme divisão de Koeppen, Francisco (2022) explica: “a umidade é a medida do teor de vapor d’água no ar pelo teor máximo que poderia conter”. Dito isso, possui a média anual de 67,6% em Goioerê e adjacências; uma umidade considerada próxima da ideal pela OMS (Organização Mundial da Saúde).

Conforme já citado, a região possui meses chuvosos durante todo o ano, isso se deve a presença de ventos que trazem as massas polares e passam pela área. Quanto à temperatura, o município possui temperaturas médias mensais mais elevadas durante o mês de janeiro e fevereiro, de 30.7° C. Já a menor temperatura média mensal, ocorre no mês de julho, com 13.8° C IAPAR (2016).

A respeito da economia do município, os dados da Secretaria de Estado da Fazenda do Paraná em 2016 mostram que 45.2% da receita gerada pelo município era proveniente do setor da agricultura. Lavoura esta que é constituída basicamente pelas culturas de soja e milho, Revisão do Plano Diretor Municipal – Goioerê (2018).

**Figura 2** - Gráfico da Economia do município de Goioerê



Fonte: Revisão do plano diretor municipal - Goioerê – Pr – 2018<sup>2</sup>

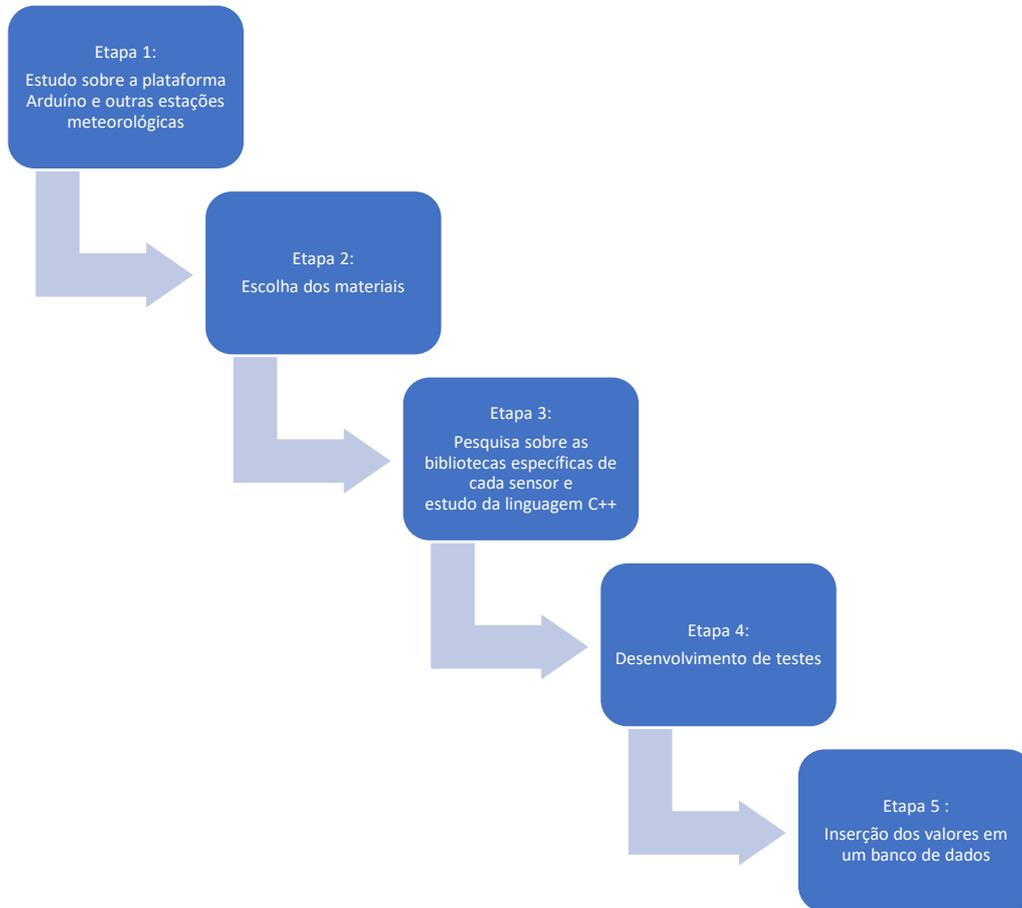
Com 45,2% da receita baseada na agricultura, nota-se a importância de estações meteorológicas que corroborem com o desenvolvimento da economia do município, afim de trazer cada vez mais o desenvolvimento para as áreas dos interiores dos estados brasileiros.

---

<sup>2</sup> Disponível em: <https://www.goioere.pr.gov.br/uploads/legislacao/132-Aspectos-Socio-Economico-Demografico-Goioere.pdf>. Acesso em: 15 de jan. de 2022.

### 3 Metodologia

**Figura 3** - Organograma das Etapas trabalhadas



Fonte: Os autores

O desenvolvimento foi feito em cinco etapas: na etapa 1, – foi realizado um estudo sobre a plataforma Arduíno e outras estações meteorológicas feitas com a placa controladora. Em que se observou os sensores e a plataforma Arduíno escolhida, bem como os módulos que permitem enviar os dados para o banco de dados. A etapa 2 – foi marcada pela escolha dos materiais. Foram escolhidos o modelo do Arduíno (Mega), o sensor de temperatura, umidade e pressão (BME280), o Módulo Ethernet (ENC28J60) que permite a conexão cabeada com o servidor, o anemômetro (SV 10) pode captar ventos que variam de 1 a 120 km/h, indicador de direção do vento (DV 10). Na etapa 3, foi feita uma pesquisa sobre as bibliotecas específicas de cada componente, verificando as que melhor funcionam, além de um estudo da linguagem e a elaboração do programa. A penúltima etapa, a etapa 4 – foi voltada ao desenvolvimento de testes; testes, a princípio, com a passagem dos dados colhidos nos sensores para uma página

base HTML e comparações com valores de outras estações. Por fim, a etapa 5 – foi destinada a inserção dos dados em um banco de dados; nessa etapa foi criado o diagrama relacional do banco de dados que será usado e finalizamos com os códigos necessários para criação e inserções de dados na base.

Dessa forma, o desenvolvimento tem por objetivo a inserção via banco de dados e acesso em um servidor local na própria rede.

### 3.1 Equipamentos Utilizados

Os equipamentos que fazem parte da estação meteorológica presente no Campus Goioerê, e que terão seus dados enviados, foram adquiridos por meio do projeto IFTECH<sup>3</sup> que visa o desenvolvimento técnico nos Institutos Federais. E são estes:

**Placa Arduino Mega:** o Arduíno foi desenvolvido pelos italianos Massimo Banzi e Gianluca Martino, e pelo sueco David Cuartielles. Ele surge como uma opção barata e fácil de ser programada para os alunos de uma universidade na Itália (EVANS et al, 2013). Dessa forma, ele pode ser explicado por Mcroberts, como: “(...) um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos” (MCROBERTS, 2011, p. 22) ele ainda o define como sendo hardware livre, ou seja, podendo ser programado para diversos fins, neste estudo, usou-se a placa para enviar dados colhidos dos sensores para um site na Internet.

O Arduíno Mega, modelo adotado na pesquisa, é um dos vários modelos existentes. O Mega possibilita mais memória e pinos, se comparado com os demais modelos existentes no mercado; (TECH SPECS ARDUÍNO, 2021). A seguir, seguem as especificações do dispositivo coletados do site oficial Arduíno.

- Microcontrolador: ATmega2560 (Arquitetura AVR de 8bits).
- Clock: 16 MHz.
- Memória flash: 256 KB(dos quais 8 KB usados pelo gerenciador de inicialização).
- SRAM: 8 KB.
- EEPROM: 4 KB.
- Tensão operacional: 5V.
- Tensão de entrada (recomendado) 7-12V.
- Tensão de entrada (limite): 6-20V.

---

<sup>3</sup> Segundo o próprio site do IFPR, o IFTECH é uma feira de inovação tecnológica que busca através de investimentos financeiros desenvolver projetos desenvolvidos por alunos dos Institutos Federais, afim do desenvolvimento de novas tecnologias inovadoras. Disponível em: <<https://reitoria.ifpr.edu.br/institucional/pro-reitorias/proepi-2/agif/inovacao-2/programas-institucionais-de-inovacao/iftech/>>

- 54 Pinos de E/S digitais.
- 16 Pinos de entrada analógicas.
- portas seriais.
- pinos para interrupção externa.

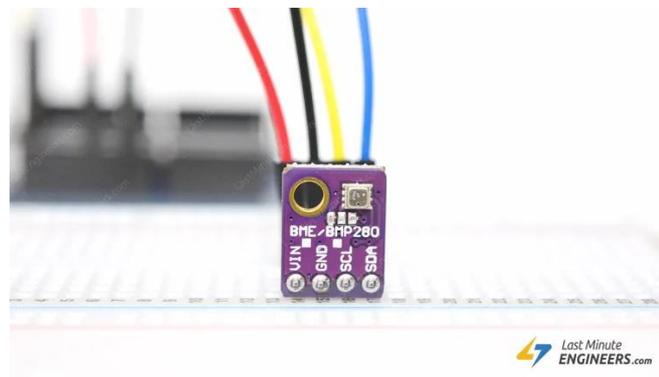
Figura 4 - Arduíno Mega



Fonte: Embarcados, 2022<sup>4</sup>

**Sensor BME280:** o módulo sensor do BME280 lê pressão barométrica, temperatura e umidade. Como a pressão muda com a altitude, é possível estimar a altitude. BME280 usa o protocolo de comunicação I2C ou SPI para trocar dados com um microcontrolador. Mede a temperatura de -40 a 85 graus Celsius, 0 a 100% de umidade, pressão de 300 hpa a 1100 hpa, altitude de 0 a 30.000ft (GLENN, 2021).

Figura 5 - Sensor BME280



Fonte: Eletrogate, 2022<sup>5</sup>

**Controlador de Internet ENC28J60:** este controlador de Internet se comunica com a placa Arduíno, estabelecendo conexão com o servidor por meio de uma rede *Ethernet*. Possui uma entrada para conector RJ45, com uma velocidade de transmissão de dados de 20MHz, dessa forma, conectando o controlador à internet (MICROCHIP DATA SHEET, 2008). Esta é uma

---

<sup>4</sup> Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 20 de dez. de 2021

<sup>5</sup> Disponível em: <https://lastminuteengineers.com/bme280-arduino-tutorial/>. Acesso em: 21 de dez. de 2021

das principais peças do sistema tratado no trabalho, tendo em vista que por meio deste *Ethernet Shield* pode-se conectar nossa estação com a rede global de computadores. (MICROCHIP ENC28J60 DATASHEET, 2008).

**Figura 6** - Controlador de Internet ENC28J60



Fonte: Baú da eletrônica, 2022<sup>6</sup>

**Indicador de Direção do Vento:** este instrumento é composto por um conjunto de resistores que se localizam em graus específicos em sua pá, cada resistor tem seu ponto específico para realizar a transmissão de um sinal, ao ser acionado o resistor emite um sinal para a placa que capta a posição emissora do valor e converte em graus (DATASHEET, 2022).

**Figura 7** - Indicador de Direção do Vento



Fonte Usinainfo, 2022<sup>7</sup>

**Sensor de Vento para Estação Meteorológica - SV10:** o sensor SV 10 é formado por conchas de alumínio conforme a figura abaixo. Em seu interior, possui um rolamento que realiza o movimento giratório deste. Dessa forma, são enviados sinais para a placa controladora que os

---

<sup>6</sup> Disponível em: <https://www.baudaeletronica.com.br/modulo-ethernet-enc28j60.html>. Acesso em 21 de dez. de 2022.

<sup>7</sup> Disponível em: <https://www.usinainfo.com.br/blog/indicador-de-direcao-do-vento-com-arduino-melhorando-sua-estacao-meteorologica/>. Acesso em 21 de dez. de 2022.

capta e os converte para M/s e Km/h, STRAUB (2017). O sensor SV 10 pode captar ventos que variam de 1 a 120 km/h (DATASHEET, 2022).

**Figura 8** – Anemômetro SV10



Fonte: UsinaInfo, 2022<sup>8</sup>

## 4 Código

O software foi desenvolvido usando a linguagem C++ na própria plataforma Arduino, conforme estipulado no Datasheet do produto, ou seja, as especificações dos produtos. A princípio foram realizadas pesquisas das bibliotecas mais eficientes e atualizadas para cada componente. A partir disso foi elaborado o código (Anexo 1). Com base em diversos autores e bibliotecas específicas para cada sensor, os autores trabalhados foram estes:

- Prof. Afonso Miguel (PUCPR - Conexão *Ethernet Shield* ENC28J60 e HTML)
- Matheus Gebert Straub (Mecânico de Manutenção com Experiência em Sistemas Embarcados, Automação e Desenvolvimento de Projetos; graduado nos Cursos de Matemática - URI e Física - UFFS. – Sensores de velocidade e direção do Vento)
- Adilson Thomsen (Formado em Ciência da Computação pela Universidade de Mogi das Cruzes).
- Nicollas (Aluno formado em Técnico de Informática pelo Instituto Federal de Brasília, IFB)

---

<sup>8</sup> Disponível em: <https://www.usinainfo.com.br/blog/anemometro-arduino-um-sensor-de-vento-para-estacao-meteorologica/>. Acesso em 22 de dez. de 2021

Todo o código comentado nos tópicos abaixo está demonstrado no anexo 1 do trabalho, para que assim possa ser visualizado e entendido de forma integra.

#### **4.1 Definição das Bibliotecas que serão utilizadas e variáveis.**

É observado que da linha 1 a 8 do código do anexo 1, são realizadas as inclusões de bibliotecas que serão utilizadas no projeto, sendo elas a biblioteca do módulo de Internet ENC28J60 (linhas 1 a 5), biblioteca do sensor BME280 (linhas 6 e 7) e a dos sensores anemômetros (linha 8) (ARDUÍNO LIBRARIES, 2021). A partir dessas bibliotecas vários métodos tornam-se disponíveis para definir-se por meio do módulo Ethernet, são exemplos desses métodos a definição do endereço da placa controladora na rede (IP) e o método de definição do endereço MAC, além de criar comunicações com servidor, banco de dados e a criação de páginas HTML (THOMSEN, 2014). Nas linhas 6 e 7, averígua-se a inclusão da biblioteca do sensor BME280 (GLENN, 2018). Ela contém métodos que realizarão a definição de qual sensor está sendo utilizado e métodos de leitura de temperatura e pressão. Do mesmo modo, na linha 8 é realizada a inclusão da biblioteca dos sensores anemômetros criados pela plataforma Arduino, juntamente com o desenvolvedor Nicholas Zambetti, ela possui todos os métodos e funções que estaremos usando nesse programa.

#### **4.2 Inicialização de variáveis**

Nessa etapa do código, é realizada a definição das variáveis, definem as variáveis que vão receber os valores coletados pelo sensor de temperatura, umidade, pressão. Também é inicializada uma variável “pin” que será a pinagem utilizada para receber sinais dos sensores no pino “0”. Da mesma forma, as variáveis que serão utilizadas nas medições anemométricas são inicializadas com o valor 0 (STRAUB, 2017), também são inicializadas as variáveis úteis para medição da velocidade do vento, que será visa no decorrer do código. Em seguida é declarado o objeto do sensor BME.

Por fim, são inicializadas as variáveis que servirão para conexão do sistema com a Internet. A função do tipo byte *mymac []* serve para definir o *MAC* (este dado é um código único na rede, que serve para identificar a placa controladora), do controlador de Internet, a função também do tipo byte *MYIP []* tem a função de definir o *IP* da placa na rede interna possibilitando, então, uma conexão com a rede externa. O IP foi definido no roteador como sendo estático para que não houvesse conflitos com o servidor DHCP (definição de IP's automáticos), ainda se define um buffer para alocar um espaço de memória para a conexão (MIGUEL, 2014).

### 4.3 Medição da Velocidade do Vento

A partir da biblioteca oficial do Arduíno Wire.h e códigos adaptados de Straub (2017), coletou-se os dados emitidos pelo sensor anemômetro podendo, então, receber os valores sobre a velocidade do vento, o qual será demonstrado em km/h para melhor compreensão.

A princípio, realizam a definição dos pinos da entrada dos valores emitidos pelo sensor SV10 no anemômetro; a partir disso, é criada uma função do tipo *void* chamada *windvelocity*, em inglês, velocidade do vento. Essa função, como o próprio nome diz, dará início ao processo de cálculo da velocidade, enquanto o anemômetro estiver girando a função estará calculando a velocidade, consoante se pode observar que enquanto a do tempo de início mais o período for menor que o tempo decorrido do sistema ativo ele continua na função (STRAUB, 2017).

Em seguida, é demonstrado o cálculo de rotações por minutos, o qual será utilizado no decorrer do código, em que para cálculo da velocidade do vento é realizada uma função matemática de velocidade angular. Por meio desse resultado, calcula-se a velocidade linear em km/h, de acordo com o que fora estabelecido no projeto. Por fim, há um contador de voltas das pás do anemômetro (STRAUB, 2017).

### 4.4 Medição da Direção do Vento

É possível analisar a leitura de dados emitidos pelo sensor de indicação do vento, como já supracitado os resistores emitirão sinais elétricos sempre que pararem em um determinado ponto. Assim sendo, o Arduíno reconhece esses sinais cada um em uma determinada direção. É realizado a leitura do pino ligado ao sensor e transformado o valor de maneira que a placa Arduíno reconhece a direção da biruta, informando para qual direção o vento está dirigindo-se (STRAUB, 2019).

### 4.5 Leitura Dados Sensor de Temperatura, Umidade e Pressão

O código do sensor de temperatura, umidade e pressão baseou-se nas bibliotecas Adafruit\_BME280.h e Adafruit\_Sensor.h, desenvolvidas pela empresa Adafruit (2020). Ela realiza a chamada das funções para coleta de pressão, temperatura e umidade e, posteriormente, é criada a função de leitura da temperatura por meio do comando `readTemperatura()`, retornando um valor *Float*, seguindo essa linha de programação é criada as funções de leitura da umidade e leitura da pressão.

### 4.6 Conexão Ethernet Shield

O Código foi adaptado do professor Afonso Miguel (2014). Ele demonstra a função de inicialização do módulo Ethernet, por meio deste equipamento é possível realizarmos a inserção em um banco de dados (MIGUEL, 2014).

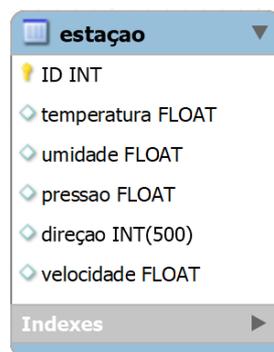
É especificado o tempo em segundos de quando será tratada essa função, verifica-se o tamanho do endereço MAC coincidente com o tamanho padrão caso não ocorra apresenta mensagem de erro. Posteriormente, caso tudo ocorra de forma correta, ele inicia os procedimentos de passagem dos IP's na rede, ou seja, o endereço de cada componente. São coletados o IP do controlador, o IP com o gateway padrão e o IP do servidor DNS, o servidor DNS segundo site oficial da AWS (2022), é definido como um servidor que irá traduzir os endereços IP (Ex: 192.168.0.1), para endereços de rede com fácil acesso para os usuários (Ex: [www.teste.com.br](http://www.teste.com.br)). É finalizado por um teste para verificação se existe ou não um servidor que realiza esse processo no projeto (MIGUEL, 2014).

#### 4.7 Banco de Dados

A inserção dos valores ocorrerá em um banco de dados criado na própria máquina local simulada como um servidor dedicado pelo aplicativo Xampp (2002). Os valores inseridos serão: temperatura, umidade, pressão, direção e velocidade do vento. Estes serão inseridos no formato *Float*, ou seja, serão acessados como números reais, exceto a direção que será declarada como *Int*, que são números inteiros.

##### 4.7.1 Criação do Banco de Dados

Figura 9 – Diagrama Relacional da Tabela “estação”



Fonte: Os autores

Para criação de um banco de dados utiliza-se a linguagem sql. Conforme explica Beaulieu (2019), a linguagem sql é uma linguagem relacional em que os dados estão relacionados entre si através de tabelas, estes dados são armazenados por tempo determinado ou não, evitando que sejam perdidos e, dessa maneira, sejam armazenados em servidores para

serem acessados sempre que necessário. Abaixo observa-se os comandos utilizados durante o processo de criação e inserções no banco de dados pela linguagem sql.

#### Quadro 4.7.1

```
1 create database estacao;
```

Fonte: Os autores

No quadro 4.7.1 temos o comando “create database” que fará a criação de uma nova base de dados para armazenamento das informações climáticas (Beaulieu, 2019). Esta base recebe o nome de “projeto\_estacao”. Dentro da nova base de dados executa-se o próximo comando:

#### Quadro 4.7.1.2

```
1 create table dados_climaticos(
2     ID INT not null primary key
3     autoincrement,
5     temperatura Float,
6     umidade Float,
7     pressao Float,
8     direcao Int,
9     velocidade Float,
10 )
```

Fonte: Os autores

O comando do quadro 4.7.1.2 visa demonstrar a criação de uma tabela dentro da base criada anteriormente. A tabela criada, contará com a coluna do ID de cada registro, além das colunas fundamentais, que armazenarão os valores climáticos. Cada coluna de dados foi definida com o tipo Float ou Int, ou seja, armazenam variáveis numerais. Assim se obtém o banco de dados que será utilizado no estudo, o próximo passo é a inserção de valores na tabela criada.

### 4.7.2 Comandos de inserção no Banco de Dados

```
1 <?php
2
3     $conn= new mysqli("localhost" , "root", "", "arduino");
4     if($conn-> connect_errno){
5         echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
6         exit();
7     }
8
9     $temperatura = $_GET['temperatura'];
10    $pressao = $_GET['pressao'];
11    $umidade = $_GET['umidade'];
12    $velocidade = $_GET['velocidade'];
13    $direcao = $_GET['direcao'];
14
15    if($temperatura == null || $umidade == null|| $pressao == null|| $direcao ==
null){
16        echo "valor nullo";
17    }
18    else{
19        $sql = "INSERT INTO estacao (temperatura,pressao,umidade,velocidade,direcao)
20        VALUES ('$temperatura', '$pressao', '$umidade', '$velocidade','$direcao)";
21
22        if ($conn->query($sql) === TRUE) {
```

```

23     session_start();
24     $last_id = $conn->insert_id;
25     $_SESSION['id'] = $last_id;
26     echo "New record created successfully" . $_SESSION['id'] ;
27 } else {
28     echo "Error: " . $sql . "<br>" . $conn->error;
29 }
30 }
31 header("Location: arduino.php");
32 ?>

```

Fonte: Os autores

Por fim, o quadro 4.7.2 demonstra o código necessário para inserções no banco de dados. Todo o código do quadro ... foi adaptado a partir de análises de sites e do vídeo do canal “Nicollas Code” (2020), ex-aluno do Instituto Federal, isso se deve a fato do módulo Ethernet ENCJ2860 não possuir bibliotecas oficiais. A linguagem utilizada é o php uma linguagem de *script open source* para desenvolvimento de códigos em HTML, porém esse desenvolvimento é realizado do lado do servidor (Manual PHP Brasil, 2022).

Em relação ao código as linhas 3 a 7 são utilizadas para uma tentativa de conexão com o banco de dados criados. Na linha 3 são passados o endereço do banco de dados, o usuário e o nome do banco. Caso ocorra erro, verificação realizada na linha 4, é enviada uma mensagem de erro e sai do script (linha 6). As linhas 9 a 13 são declaradas as variáveis que serão inseridas no banco. Os valores dessas variáveis são recebidos por meio do método Get que recebe os dados dentro do loop do código Arduino (Wippler, 2018).

Na linha 15 é realizado uma verificação para checagem se os valores recebidos no método Get não são nulos, caso não sejam, é criado uma variável “\$sql” que possui a função de armazenar o comando “INSERT INTO”, o nome da tabela, os nomes das colunas e os valores que cada coluna irá receber. O comando “INSERT INTO” é um comando da linguagem sql que irá inserir os valores em si na tabela “estação”, conforme criado anteriormente (Beaulieu, 2019). Por fim, é passado esse comando da variável “\$sql” para a conexão criada na linha 3, caso os valores sejam inseridos, é retornado o último “ID” do registro e enviada uma mensagem de sucesso (linha 26), dessa forma, na linha 31, é chamado um novo script “arduino.php” que criará a página HTML (Wippler, 2018).

### 4.7.3 Select no Banco de Dados e Página HTML

Quadro 4.7.3

```

1 <?php
2     $conn= new mysqli("localhost" , "root", "", "arduino");
3     if($conn-> connect_errno){
4         echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
5         exit();
6     }

```

```

7     $sql = "SELECT MAX(id) AS id FROM estacao";
8     $result = mysqli_query($conn, $sql);
9     $row = mysqli_fetch_object($result) ;
10    //echo "Maximum Working Hours: " . $row->id;
11    $r = $conn -> query("SELECT * FROM estacao WHERE id=( $row->id)");
12 ?>
13 <!DOCTYPE html>
14 <html>
15   <head>
16     <meta charset="UTF-8">
17     <title>Estação</title>
18     <style>
19     .position{
20       height: 200px;
21       float: right;
22       float: left;
23     }
24     h1{
25       padding-top: 12px;
26     }
27     .t{
28       width:15%;
29     }
30     .tt{
31       width:75%;
32       padding-top: -30px;
33       padding-right: -10px;
34     }
35     table{
36       padding-top: 80px;
37       width: 100%;
38       padding-left: 180px;
39     }
40   </style>
41 </head>
42 <body>
43   <div class="row">
44     <div class="position" id="t">
45       
46     </div>
47     <div class="position" id="tt">
48       <h1>Estação Meteorológica Campus Avançado Goioerê</h1>
49     </div>
50   </div>
51   <div class="row-2">
52     <table border-collapse>
53       <tr> <?php $dado = $r->fetch_array() ?>
54         <td width="100" height="40">Temperatura:<?php echo $dado['temperatura'];?> C</td>
55         <td width="100" height="40">Pressão:<?php echo $dado['pressao'];?> hPa</td>
56         <td width="100" height="40">Umidade: <?php echo $dado['umidade'];?>%</td>
57         <td width="100" height="40">Velocidade Do Vento:<?php echo $dado['velocidade'];?> K/h</td>
58         <td width="100" height="40">Direção:<?php echo $dado['direcao'];?>Graus</td>
59       </tr>
60     </tr>
61   </table>
62 </div>
63 </body>
64 </html>

```

Fonte: Os autores

O código acima, do quadro 4.7.3, demonstra nas linhas 1 a 12 a criação de um *script* php. Este script tem a função de criar uma nova conexão com o banco de dados (linhas 2 a 6), e posteriormente realiza um “SELECT”, que segundo Beaulieu (2019), realiza uma seleção no

banco de dados e retorna o valor desejado. No caso do “SELECT” da linha 7 o retorno é o “ID” do último registro inserido no banco. Por meio desse registro, é feita uma nova chamada na linha 11, essa chamada trará todos os valores da tabela que estão ligados com o ID colhido na linha 7, ou seja, serão recebidos todos os dados armazenados na última inserção do banco de dados. Estes dados ficarão armazenados no *array* “\$r”, criado na linha 11.

Posteriormente é demonstrado a criação de uma página HTML (linhas 13 a 64) que ao ser chamada durante o código irá coletar a temperatura, umidade, pressão, direção e velocidade do vento (Linhas 54 a 58), que estavam armazenadas no *array* “\$r”, e passado para um novo *array* de dados que terá acesso a cada dado climático, anteriormente armazenados no banco de dados.

Para Silva (2008), tratando-se do HTML ela é uma linguagem de hipertextos que darão as bases para desenvolvimento de sites *web*, eles serão executados nos navegadores e são formados basicamente por *tags*, cada *tag* determina uma função. No código, exemplos de *tags* são: <body> </body>, <h1> </h1>. A *tag* Body, por exemplo, possui a função de determinar o “corpo” da página em que será demonstrada ao usuário, já a *tag* h1 é um modelo de escrita de texto que serão emitidos na página WEB, Silva (2008). Dessa forma, o sistema poderá emitir na tela os valores coletados nas variáveis

#### **4.8 Saída Serial**

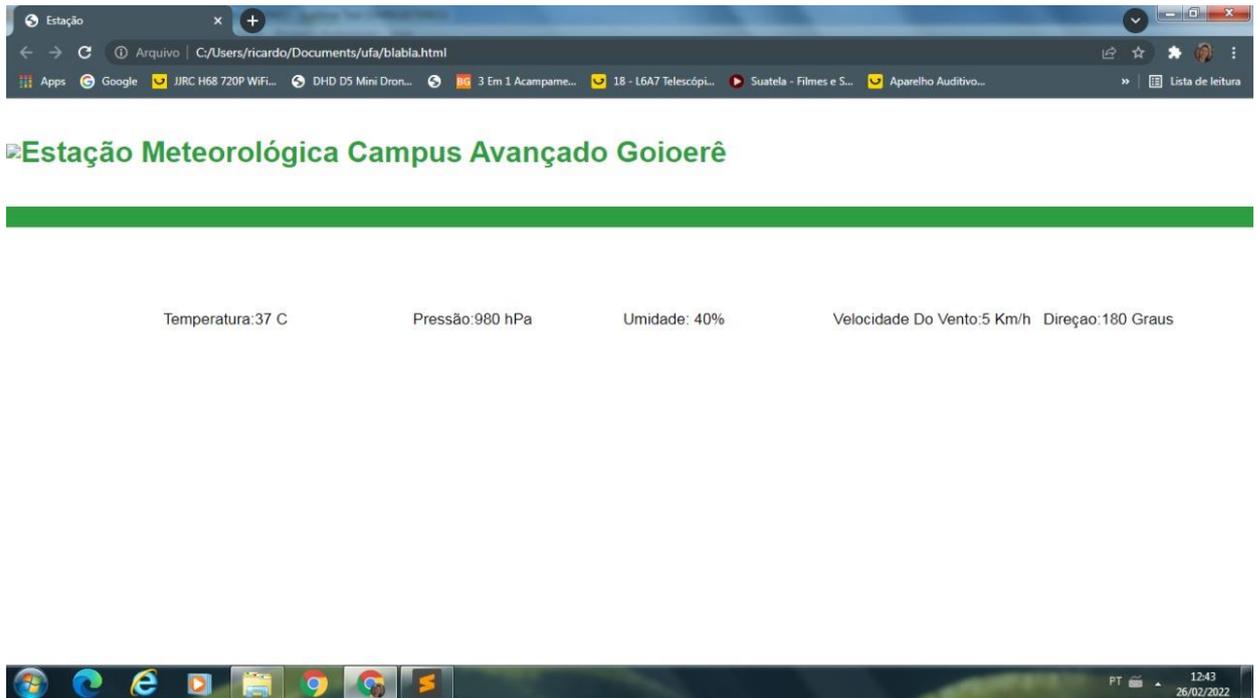
É realizada uma saída no terminal para verificar se todas as variáveis foram preenchidas corretamente e caso dê algum erro, torna-se de forma mais rápida a correção. O comando Serial.print escreve na saída serial o valor que se encontrar entre parênteses, como será demonstrado, por meio de um teste realizado em nossa estação.

Assim sendo, concluímos o cumprindo o objetivo da criação do banco de dados. Os resultados demonstrados foram colhidos em testes para verificar se todas as linhas estão corretas e o sistema funcionando, de forma íntegra.

### **5. Resultados**

Com a compilação e carregamento em nossa placa controladora, chega-se aos resultados parciais, a fim de demonstrar a funcionalidade da estação meteorológica. Partindo disso, os resultados obtidos foram os seguintes:

Figura 10 – Página HTML



Fonte: Os autores

Na figura 10, nota-se a página HTML criada pelo *script* PHP, os dados são atualizados de 4 em 4 segundos. E são obtidos diretamente do banco de dados. Na figura abaixo realizamos um teste para verificar se os valores coletados em nossa estação estão próximos com os valores da estação de Umuarama, conforme citados em seção anterior a cidade se encontra a 60km o que justifica algumas variações.

Figura 11 – Dados Climáticos, Umuarama



Fonte: Tempo.com<sup>9</sup>

Como podemos verificar por meio da figura 11, as temperaturas e umidades variam pouco, porém com uma nova estação meteorológica os dados obtidos para a região de Goioerê podem-se contribuir para melhorar ainda mais a aferição de dados climáticos. A placa Arduino vai escrever e enviar os dados para o banco de dados a cada 4 segundos e para o terminal do Arduino as informações a cada 2 segundos.

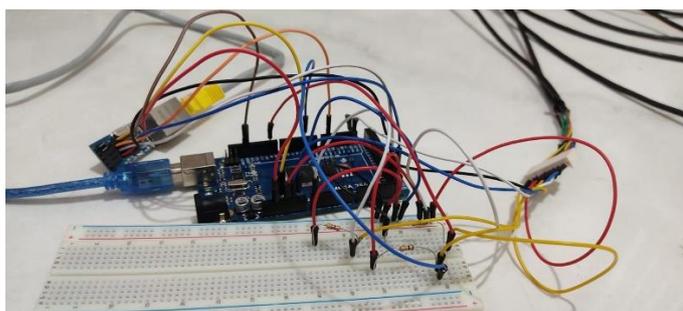
**Figura 12** – Saída Serial

```
Temperatura:26.8°C  
Umidade: 72%  
Pressao: 954.82 hPa  
Direcao: 315 graus --Veloc 5.32 Km/h
```

Fonte: Os autores

Na figura 12, vê-se a saída serial diretamente do console de programação do sistema, assim pode-se averiguar sempre que necessário se os valores colhidos irão coincidir. Até o presente momento, a estação meteorológica desenvolvida se encontra da seguinte forma:

**Figura 13** – Placa Arduino



Fonte: Os autores

Pela figura 13, nota-se o cabo Ethernet (cabo branco) conectado ao módulo Ethernet de forma a possibilitar o acesso WEB e as conexões dos sensores que ligam os fios de alimentação e de leitura de dados (cabo preto).

---

<sup>9</sup> Disponível em: < <https://www.tempo.com/hoje/umuarama.htm> >. Acesso em: 18/02/2022

**Figura 14** – Estação Meteorológica



Fonte: Os autores

Os sensores foram instalados em um ambiente externo, a fim de captar os valores climáticos de forma correta. A estação conta com um suporte em que são instaladas as bases dos sensores.

## **6. Considerações Finais**

Conclui-se então que o trabalho foi desenvolvido baseado em três linguagens diferentes: C++, PHP e SQL, além da linguagem de hipertextos HTML. Os dados colhidos ainda necessitam de passarem por mais testes, para que se comprove fielmente a veracidade da estação, porém conforme notamos no teste realizado os valores não estão foram da média, se comparado com outra estação já existente. Notou-se que a temperatura, umidade e pressão se encontram de forma muito semelhantes. O que significa que o sensor não está recebendo interferências externas, que sirvam para inviabilizar o projeto. Porém com mais ciclos de testes, a estação contará com uma veracidade muito maior.

Vale ressaltar, a possibilidade de, em trabalhos futuros, a inserção dos valores climáticos em uma página web no site do IFPR – Campus Avançado Goioerê, podendo ser acessado via rede global de computadores em qualquer local, não somente via banco de dados e servidor local na própria rede.

Assim sendo, espera-se que o projeto possa ter continuidade para que o objetivo de ajudar no desenvolvimento do município de Goioerê e região, possa ser alcançado com sucesso.

## 7. Referências

ADAFRUIT (USA). Adafruit Unified Sensor. Adafruit Unified Sensor. [S. l.]: Arduino.cc, 2020. Disponível em: <https://www.arduino.cc/reference/en/libraries/adafruit-unified-sensor/>. Acesso em: 16 jan. 2022.

ARDUINO. *Tech Specs Arduino mega 2560*. Disponível em: <https://store-usa.arduino.cc/products/arduino-mega-2560-rev3>. Acesso em: 15 de jan. de 2022.

ARDUINO | Módulo Ethernet Shield ENC28J60 | Comunicação com Servidor PHP + MySQL |. Direção: Nicollas Code. Brasil: Youtube, 2020. Disponível em: <https://www.youtube.com/watch?v=ufWfV1poePQ&t=192s>. Acesso em: 18 fev. 2022.

BEAULIEU, Alan. *Aprendendo SQL: Dominando os Fundamentos de SQL*. 2. ed. São Paulo: Novatec, 2009. 368 p. Disponível em: [https://www.google.com.br/books/edition/Aprendendo\\_SQL/1Cp4DQAAQBAJ?hl=pt-BR&gbpv=1](https://www.google.com.br/books/edition/Aprendendo_SQL/1Cp4DQAAQBAJ?hl=pt-BR&gbpv=1). Acesso em: 18 fev. 2022.

CEPEA - Centro de Estudos Avançados em Economia. *Pib Do Agronegócio Alcança Participação De 26,6% No Pib Brasileiro Em 2020*. Piracicaba, SP, 2021. Disponível em: <https://cepea.esalq.usp.br/br/releases/pib-agro-cepea-com-avanco-de-24-3-no-ano-pib-agro-alcanca-participacao-de-26-6-no-pib-brasileiro-em-2020.aspx>. Acesso em: 22 de dez de 2021.

EVANS, M., Noble, J.,Hochenbaum, J. (2013). *Arduino in action*. Manning Publications, p. 368

FRANCISCO, Wagner de Cerqueira e. "Umidade do Ar"; *Brasil Escola*. Disponível em: <https://brasilecola.uol.com.br/geografia/umidade-ar.htm>. Acesso em 27 de janeiro de 2022.

GLENN, Tyler. BME280. In: BME280. [S. l.], 26 fev. 2018. Disponível em: <https://www.arduino.cc/reference/en/libraries/bme280/>. Acesso em: 27 jan. 2022.

INMET – Instituto Nacional de Meteorologia. *SOBRE METEOROLOGIA*. Brasília, DF, 2015. Disponível em: <https://portal.inmet.gov.br/sobre-meteorologia>

MAACK, R. *Geografia física do Estado do Paraná*. 3. ed. Curitiba: Imprensa Oficial, 2002.

MICROCHIP (USA). ENC28J60 Data Sheet: Stand-Alone Ethernet Controller with SPI Interface. USA: [s. n.], 2008. 96 p. Disponível em: <https://storage.googleapis.com/baudaeletronicadatasheet/ENC28J60.pdf>. Acesso em: 27 jan. 2022.

MIGUEL, Afonso. *Arduino EthernetShield ENC28J60 Configuration*. Youtube, 17 março de 2014. Disponível em: <https://www.youtube.com/watch?v=d2toibPesS4>. Acesso em: 15 de janeiro de 2022.

PHP (BRASIL). *PHP BRASIL. Manual PHP*. In: *PHP BRASIL. Manual PHP*. [S. l.], 2022. Disponível em: [https://www.php.net/manual/pt\\_BR/intro-what-is.php](https://www.php.net/manual/pt_BR/intro-what-is.php). Acesso em: 18 fev. 2022.

PREFEITURA DE GOIOERÊ. *Revisão do plano diretor municipal cap 1.2 Aspectos Ambientais*. Goioerê, PR, 2018. Disponível em: <https://goioere.pr.gov.br/uploads/legislacao/12-Aspectos-Ambientais-Goioere-finalizado.pdf>

ROLIM, Glauco de Souza et al. *Classificação climática de Köppen e de Thornthwaite e sua aplicabilidade na determinação de zonas agroclimáticas para o estado de São Paulo*. *Bragantia*, v. 66, p. 711-720, 2007.

SILVA, Maurício Samy. Criando sites com HTML: Sites de alta qualidade com HTML e CSS. São Paulo, SP. Novatec Editora Ltda, 2008. Disponível em: <[https://www.google.com.br/books/edition/Criando\\_Sites\\_com\\_HTML/HBVQ-w5ZcoC?hl=pt-BR&gbpv=1&printsec=frontcover](https://www.google.com.br/books/edition/Criando_Sites_com_HTML/HBVQ-w5ZcoC?hl=pt-BR&gbpv=1&printsec=frontcover)>. Acesso em: 15 de janeiro de 2022

STRAUB, Matheus Gebert. Anemômetro Arduino – Um Sensor de Vento para Estação Meteorológica. Usinainfo. Disponível em: <<https://www.usinainfo.com.br/blog/anemometro-arduino-um-sensor-de-vento-para-estacao-meteorologica/>>. Acesso em: 14 de janeiro de 2022

THOMSEN, Adilson. Comunicação pela rede com o Módulo Ethernet ENC28J60. Filipeflop. Disponível em: <https://www.filipeflop.com/blog/modulo-ethernet-enc28j60-arduino/>. Acesso em 15 de janeiro de 2022.

USINAINFO, Indicador de Direção de Vento para Estação Meteorológica. Disponível em: <https://www.usinainfo.com.br/estacaometeorologica-arduino/indicador-de-direcao-do-vento-arduino-para-estacaometeorologica-dv10-4638.html>

YNOUE, Rita Yuri; REBOITA, Michelle S.; AMBRIZZI, Tércio; DA SILVA, Gyrlene A. M. Meteorologia: noções básicas. 1. ed. [S. l.]: Oficina de Textos, 2017. 184 p.

ZAMBETTI, Nicholas. Arduino. Wire Library. Arduino.cc, 2019. Disponível em: <<https://www.arduino.cc/en/reference/wire>>. Acesso em: 15 de janeiro de 2022.

## 8. Anexos

### Anexo 1

```

1  #include <bufferfiller.h>
2  #include <enc28j60.h>
3  #include <EtherCard.h>
4  #include <net.h>
5  #include <stash.h>
6
7  #include <Adafruit_Sensor.h>
8  #include <Adafruit_BME280.h>
9
10 # define Hall sensor 2
11 int contador;
12 float temperature ;
13 float humidity;
14 float pressure ;
15 #define ALTITUDE 0
16 static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };
17 byte Ethernet::buffer[700];
18 static uint32_t timer;
19
20 char mensagem[80];
21
22 const byte websiteip[] = {192,168,0,107};
23 const char website[] PROGMEM = "192.168.0.107";
24 static byte myip[]={192,168,0,113};
25 static byte gwip[]={192,168,0,1};
26
27 static void my_callback (byte status, word off, word len)
28 {
29   Serial.println("-----");
30   Serial.println("entrou no mycalllback");
31   Serial.println("-----");
32   Ethernet::buffer[off+len] = 0;    //mensagem do retorno do servidor
33   // Serial.print((const char*) Ethernet::buffer + off); //mostra todo o
retorno
34   Serial.print((const char*) Ethernet::buffer + off + 157); //mostra somente ok
35 }
36
37 // Const def biruta
38
39 int pin=0; // A0 entrada do biruta *
40 float valor =0;
41 float Winddir =0;
42
43
44 // Constants definitions anem.
45 const float pi = 3.14159265; // Numero pi
46 int period = 2500; // Tempo de medida(miliseconds) 5000
47 int radius = 147; // Raio do anemometro(mm)
48
49 // Variable definitions
50
51 unsigned int counter = 0; // magnet counter for sensor
52 unsigned int RPM = 0; // Revolutions per minute
53 float speedwind = 0; // Wind speed (Km/h)
54
55 Adafruit_BME280 bme; // I2C
56
57
58
59 void setup() {
60   Serial.begin(9600); //serial
61   //bme280
62   bool status;
63   status = bme.begin(0x76); //The I2C address of the sensor I use is 0x76

```

```

64     if (!status) {
65         Serial.print("Sensor not working, check connections");
66         while (1);
67     }
68
69     // Anemômetro
70     pinMode(2, INPUT); // Entrada anemometro *
71     digitalWrite(2, HIGH); //internall pull-up active
72
73     Serial.println("Obtendo IP:");
74     if (ether.begin(sizeof Ethernet::buffer, mymac,53) == 0){
75         Serial.println( "Falha ao acessar o controlador Ethernet");
76     }
77     ether.staticSetup(myip, gwip);
78     ether.printIp("IP: ", ether.myip);
79     ether.printIp("Gateway: ", ether.gwip);
80     ether.printIp("Sevidor DNS: ", ether.dnsip);
81     if (!ether.dnsLookup(website)){
82         Serial.println("Falha no DNS");
83     }
84     ether.parseIp(ether.hisip, "192.168.0.107"); //IP do servidor manual
85     ether.printIp("Servidor: ", ether.hisip);
86     Serial.println(" ");
87 }
88
89 // Calculo da velocidade wind *****
90 void windvelocity(){
91     speedwind = 0;
92     counter = 0;
93     attachInterrupt(0, addcount, RISING);
94     unsigned long millis();
95     long startTime = millis();
96     while(millis() < startTime + period) {
97     }
98 }
99
100
101 void RPMcalc(){
102     RPM=((counter)*60)/(period/500); // Calculate revolutions per minute (RPM)
103 }
104
105
106 void SpeedWind(){
107     speedwind = (((4 * pi * radius * RPM)/60) / 500)*3.6; // Calculate wind
speed on km/h
108 }
109
110
111 void addcount(){
112     counter++;
113 }
114 // Fim calculo veloc wind *****
115
116
117
118 void loop() {
119
120
121     //anemometro
122     windvelocity();
123     RPMcalc();
124     SpeedWind();
125
126     //biruta
127     valor = analogRead(pin)* (5.0 / 1023.0);
128
129     if (valor <= 0.27) {
130         Winddir = 315;

```

```
131     }
132     else if (valor <= 0.32) {
133     Winddir = 270;
134     }
135     else if (valor <= 0.38) {
136     Winddir = 225;
137     }
138     else if (valor <= 0.45) {
139     Winddir = 180;
140     }
141     else if (valor <= 0.57) {
142     Winddir = 135;
143     }
144     else if (valor <= 0.75) {
145     Winddir = 90;
146     }
147     else if (valor <= 1.25) {
148     Winddir = 45;
149     }
150     else {
151     Winddir = 000;
152     }
153
154     //pluviometro
155
156     Serial.println(valor);//bme280
157     getPressure();
158     getHumidity();
159     getTemperature();
160     char t[50];
161     char h[50];
162     char p[50];
163     char w[50];
164     char sw[50];
165     dtostrf(pressure,1,2,p);
166     dtostrf(Winddir,1,2,w);
167     dtostrf(speedwind,1,2,sw);
168     dtostrf(temperature,1,2,t);
169     dtostrf(humidity,1,2,h);
170     //Printing Temperature
171     String temperatureString = String(temperature,1);
172     Serial.write("Temperatura:");
173     Serial.print(temperatureString);
174     Serial.write((char)223);
175     Serial.write("C ");
176     Serial.println("");
177
178     //Printing Humidity
179     String humidityString = String(humidity,0);
180     Serial.write("Umidade: ");
181     Serial.print(humidityString);
182     Serial.write("%");
183     Serial.println("");
184
185     //Printing Pressure
186     Serial.write("Pressao: ");
187     String pressureString = String(pressure,2);
188     Serial.print(pressureString);
189     Serial.write(" hPa");
190     Serial.println("");
191
192     Serial.print("Direcao: ");
193     Serial.print(Winddir);
194     Serial.print(" graus ");
195     Serial.print("--Veloc ");
196     Serial.print(speedwind);
197     Serial.print(" Km/h ");
198     Serial.println("");
```

```
199     ether.packetLoop(ether.packetReceive());
200     if (millis() > timer){
201         timer = millis() + 8000;//4000
202         ether.persistTcpConnection(true);
203         sprintf(mensagem,
"?temperatura=%s&pressao=%s&umidade=%s&velocidade=%s&direcao=%s",t,p,h,sw,w);//diz que
o valor é nulo,e nao insere no banco
204         ether.browseUrl(PSTR("/teste.php"),mensagem, website, my_callback);
205     }
206
207 }
208
209 float getTemperature()
210 {
211     temperature = bme.readTemperature();
212 }
213
214 float getHumidity()
215 {
216     humidity = bme.readHumidity();
217 }
218
219 float getPressure()
220 {
221     pressure = bme.readPressure();
222     pressure = bme.seaLevelForAltitude(ALTITUDE,pressure);
223     pressure = pressure/100.0F;
224 }
```