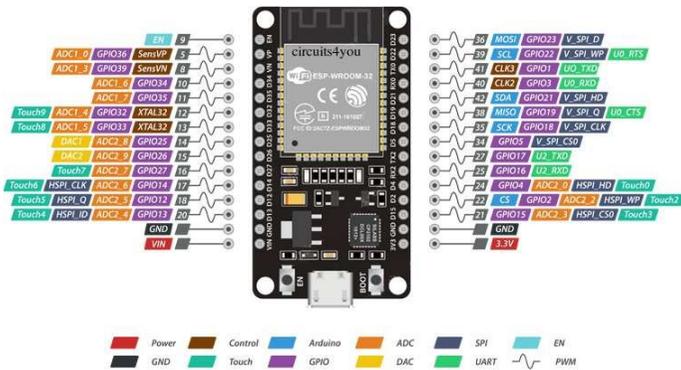


Guia de Programação ESP32



ESP32 Dev. Board Pinout

Hernani Batista da Cruz
Sani de Carvalho Rutz da Silva
Silvio Luiz Rutz da Silva

Guia de Programação ESP32

Hernani Batista da Cruz
Sani de Carvalho Rutz da Silva
Silvio Luiz Rutz da Silva



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos.

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

Sumário

Introdução.....	3
Instalação IDE Arduino.....	5
Fonte: Autoria própria (2022).	9
Instalação do Xampp.....	10
Grafana.....	15
Protocolos de comunicação.....	17
UART	17
SPI.....	19
I ² C	20
Circuito ESP32/DHT11/BMP180	22
Fonte: Autoria própria (2022).	22
Fonte: Autoria própria (2022).	23
Fonte: Autoria própria (2022).	24
Fonte: Autoria própria (2022).	25
Fonte: Autoria própria (2022).	31
Acessando o phpmyadmin.....	32
Fonte: Autoria própria (2022).	36
Recebendo os dados enviados pelo ESP32	37
Fonte: Autoria própria (2022).	38
Dashboard no Grafana	40
Bibliografia	46
Autores	48

Introdução

A utilização de microcontroladores em aulas de Física e Informática possibilita que os professores desenvolvam temas específicos das áreas de Ciência e Computação, além de automatizar a coleta de dados experimentais. Com placas como Uno, Mega e outras, é possível realizar experimentos e salvar os dados em um cartão SD ou, por meio de um programa em Python, ler a porta USB em que o Arduino está conectado a um computador e salvar os dados em arquivos de texto. Adicionando um módulo de Wi-Fi, também se torna possível enviar os dados para um banco de dados externo.

Em 2016, com o surgimento das placas da família ESP32/ESP8266, a aquisição de dados e envio para persistência em bancos de dados tornou-se mais fácil, assim como a utilização de recursos para visualização dos dados, como as ferramentas de MQTT (Message Queue Telemetry Transport).

Este guia tem como objetivo tornar mais acessível aos professores de Física e estudantes de Informática os primeiros passos na configuração e utilização das placas ESP32/ESP8266 em projetos. Foi desenvolvido para auxiliar professores iniciantes na abordagem de atividades baseadas em Aprendizagem Ativa, a partir do aniversário de 50 anos da missão Apollo 11, em 2019.

A plataforma Arduino surgiu na Itália em meados de 2007 e permitiu que professores, robóticos e empreendedores desenvolvessem seus projetos com custos baixos, por meio de um microcontrolador. Atualmente, existem vários modelos de placas da família Arduino, que possibilitam projetos das mais diversas formas. Nas próximas páginas, apresentaremos um guia de utilização de sensores e motores usando o Arduino.

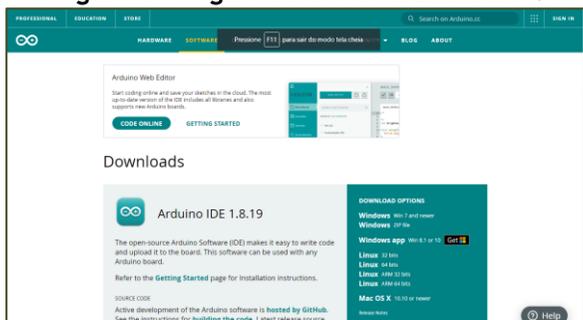
Sempre que possível, serão mostrados exemplos usando a placa ESP32 e armazenamento de dados em banco relacional do tipo SQL.

Para isso, incluímos um capítulo sobre a instalação de um servidor web e exemplos de códigos de conexão com o banco de dados e inserção de dados no banco. Além disso, é importante ressaltar que os procedimentos apresentados partem do princípio de que o leitor nunca teve contato com o Arduino, ESP32, PHP, MySQL e Grafana. Assim, apresentamos os passos fundamentais para a instalação e utilização da plataforma ESP32, enviando um pacote de dados para uma página web, codificada com PHP, para que os valores lidos sejam inseridos no banco de dados MySQL e posteriormente exibidos em um dashboard construído com Grafana.

Instalação IDE Arduino

Para construir um ambiente de desenvolvimento, inicialmente precisamos de uma plataforma de programação para o microcontrolador. O link abaixo é o endereço oficial da IDE (Interface de Desenvolvimento Integrado) do Arduino, atualmente na versão 1.8.19. <https://www.arduino.cc/en/software>. A Figura 1 abaixo apresenta um print da página oficial da IDE.

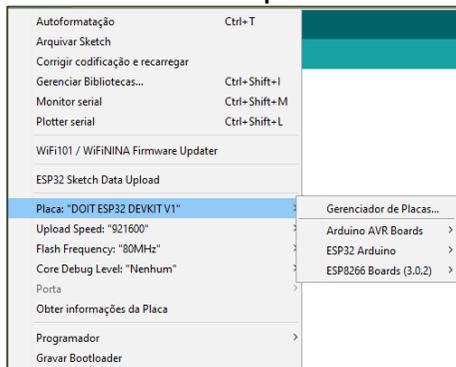
Figura 1 - Página inicial da IDE Arduino.



Fonte: Arduino (2022).

Escolha a versão adequada em relação ao sistema operacional que utiliza e realize o download. Em seguida, se sua máquina estiver rodando o Sistema Operacional Windows, atente para clicar na permissão dos drivers que devem aparecer durante a instalação. Em Sistemas Linux, não é solicitada nenhuma permissão durante a instalação da IDE do Arduino. A Figura 2, mostra a tela inicial da IDE do Arduino.

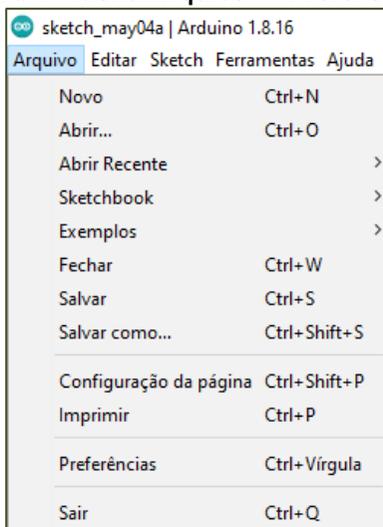
Figura 3 - Menu de escolha da placa de desenvolvimento.



Fonte: Autoria própria (2022).

Para adicionar as placas de desenvolvimento ESP32 e ESP8266, os endereços dos links da web que contêm os dados das placas devem ser adicionados as preferências da IDE do Arduino. Para inseri-las na IDE do Arduino, siga os seguintes passos:

Figura 4 - Menu Arquivo -> Preferências



Fonte: Autoria própria (2022).

Em Preferências busque por “URLs Adicionais para Gerenciamento de Placas” que encontra-se no Menu Arquivo.

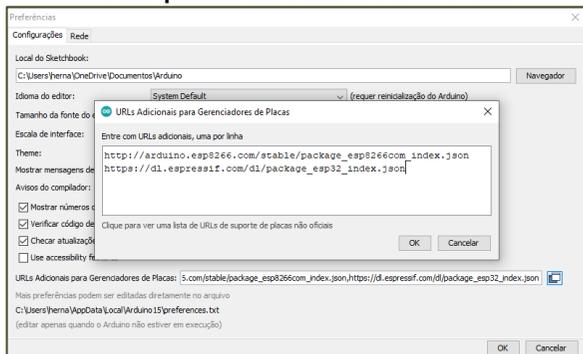
As URLs que contém os dados das placas que devem ser adicionadas em preferências são as seguintes:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

https://dl.espressif.com/dl/package_esp32_index.json

A Figura 5 mostra a caixa de texto onde se insere os links de novas placas de desenvolvimento.

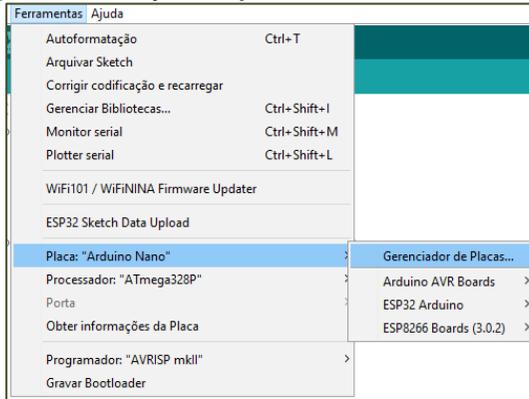
Figura 5 - Caixa de texto onde se insere o link de repositório de novas placas de desenvolvimento.



Fonte: Autoria própria (2022).

Em seguida, no Menu Ferramentas, clique em Placa -> Gerenciador de Placas mostrado na Figura 6.

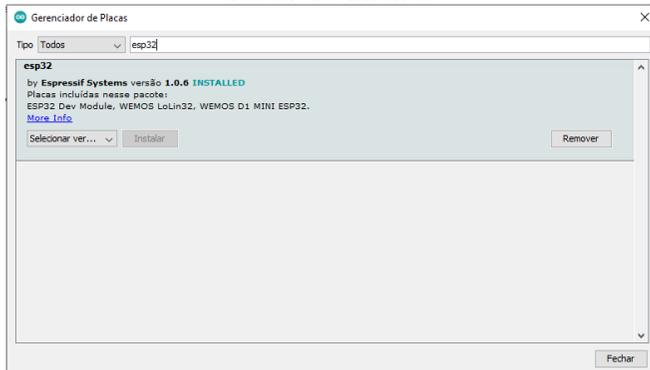
Figura 6 - Seleção de placa de desenvolvimento.



Fonte: Autoria própria (2022).

Na barra de pesquisa, escreva ESP32, como podemos ver na Figura 7.

Figura 7 - Caixa de pesquisa para inserir nova placa de desenvolvimento.



Fonte: Autoria própria (2022).

Instalação do Xampp

XAMPP é um pacote de software livre que contém os principais servidores de código aberto para desenvolvimento web, incluindo Apache, MySQL, FTP e Perl. Esses softwares permitem que os usuários criem e executem um servidor web local em seus computadores pessoais para testar e desenvolver sites e aplicativos web.

O nome XAMPP é um acrônimo que significa "X" (cross-platform), "Apache", "MySQL", "PHP" e "Perl". É uma solução fácil de instalar e configurar que pode ser executada em sistemas operacionais Windows, Linux e MacOS.

O XAMPP é uma ferramenta útil para desenvolvedores web, pois permite testar aplicativos localmente antes de implantá-los em um servidor web real. Ele fornece uma plataforma de desenvolvimento pronta para uso, que inclui as tecnologias mais populares usadas na criação de aplicativos web, como o servidor web Apache, o banco de dados MySQL e a linguagem de programação PHP.

O XAMPP é especialmente útil para desenvolvedores que trabalham com aplicativos baseados em PHP e MySQL, pois esses aplicativos podem ser facilmente criados e testados em um ambiente local antes de serem implantados em um servidor web real.

Além disso, o XAMPP vem com uma interface gráfica do usuário (GUI) que permite gerenciar os servidores e aplicativos instalados. Isso torna mais fácil para os usuários iniciantes gerenciar seus servidores e aplicativos web sem precisar usar linhas de comando. Ferramenta essencial para desenvolvedores web que desejam criar e testar aplicativos

localmente antes de implantá-los em um servidor web real. Com o XAMPP, os usuários podem criar um ambiente de desenvolvimento fácil de usar e configurar rapidamente em seus próprios computadores pessoais.

Acesse a URL (https://www.apachefriends.org/pt_br/index.html) e faça o download do programa em seu computador. A Figura 8 mostra as opções de versionamento disponíveis quando o Guia de Programação ESP32 foi elaborado.

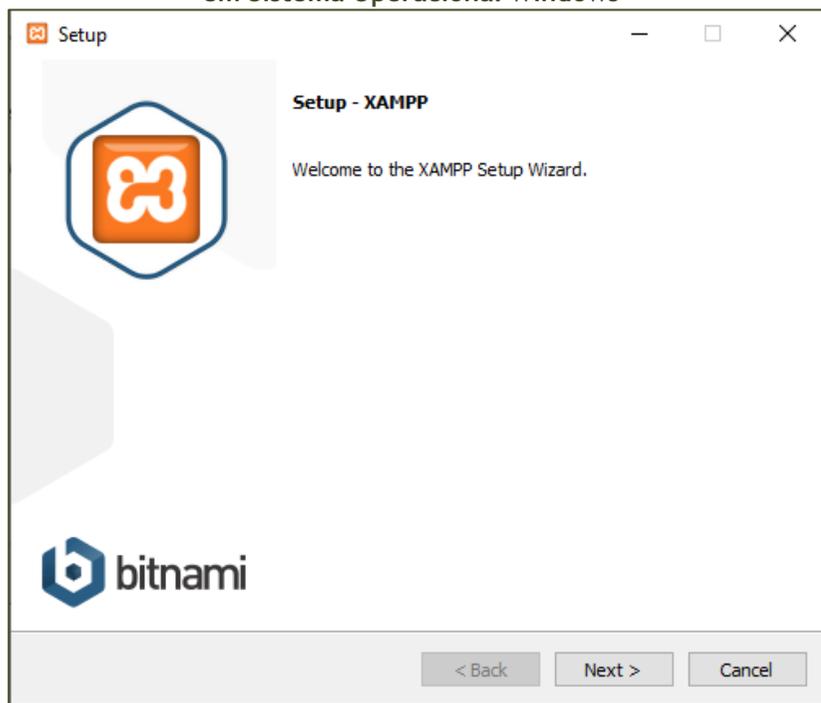
Figura 8 - Nesse passo a passo, utilizei a versão 8.1.6 / PHP 8.1.6.



Fonte: Autoria própria (2022).

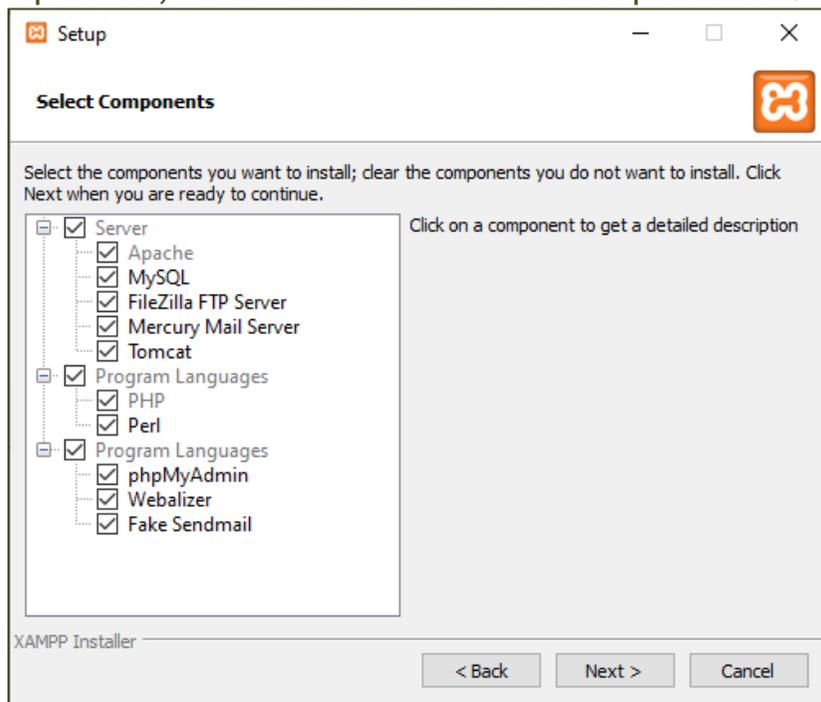
Quando terminar de baixar o programa, clique duas vezes sobre o arquivo e proceda clicando em Next.

Figura 9 - Tela inicial do procedimento de instalação do Xampp em Sistema Operacional Windows



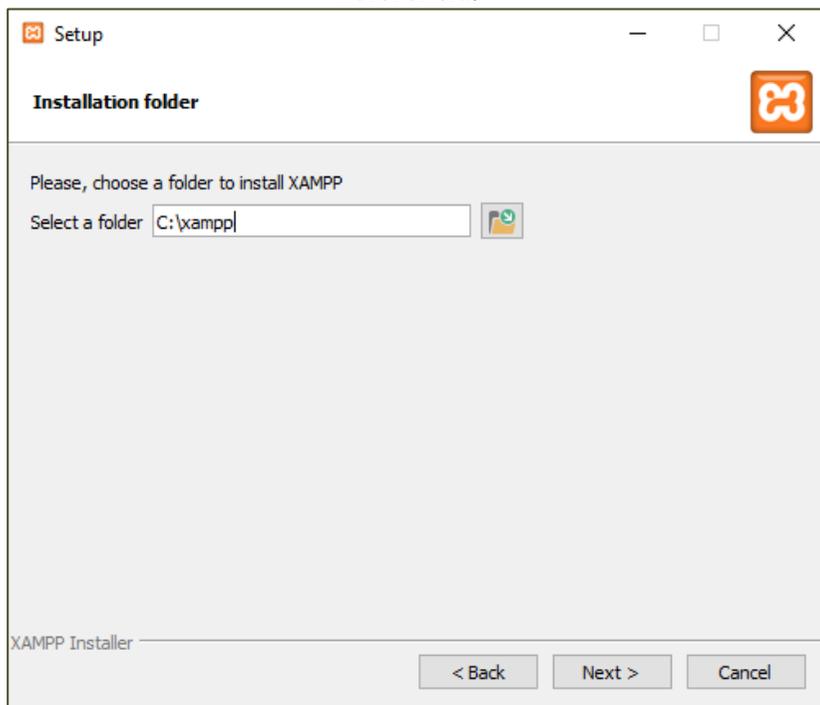
Fonte: Autoria própria (2022).

Figura 10 - Seleção dos componentes a serem instalados. utilizaremos apenas o Apache, PHP e MySQL, mas caso queira se aprofundar, mantenha os demais itens marcados para estudos.



Fonte: Autoria própria (2022).

Figura 11 - Escolha da pasta onde será instalado o Xampp. Por padrão é colocada em c:\xampp. Caso tenha familiaridade pode mudar a pasta. o presente guia, utilizará a pasta padrão como referência.



Fonte: Autoria própria (2022).

Grafana

Grafana é uma plataforma de análise e visualização de dados de código aberto, que permite criar painéis e gráficos interativos para monitorar e analisar dados em tempo real. Ele suporta vários tipos de fontes de dados, incluindo bancos de dados SQL, NoSQL, serviços de nuvem, além de dispositivos de IoT.

É altamente personalizável e oferece uma ampla variedade de opções de visualização de dados, incluindo gráficos de barras, linhas, medidores, tabelas, mapas, entre outros. Ele também suporta alertas para notificar quando os dados atingem um limite predefinido, bem como recursos de compartilhamento para que os painéis possam ser compartilhados com outros usuários.

O Grafana é amplamente utilizado em ambientes de negócios e de TI para monitorar a saúde de aplicativos, infraestrutura de rede e serviços de nuvem. Ele também é usado por equipes de IoT para monitorar dispositivos e sensores em tempo real.

A plataforma é altamente escalável e fácil de usar, com uma grande comunidade de usuários ativos que contribuem para o desenvolvimento contínuo e melhoria da plataforma. Pode fazer o download através do seguinte endereço web: <https://grafana.com/>. A Figura 12 mostra a página inicial do site do Grafana.

Figura 12 - Site oficial da plataforma Grafana.

The image shows the homepage of the Grafana Labs website. At the top, there is a navigation bar with the following items: Grafana Labs logo, Products, Open source, Solutions, Learn, Company, a search icon, Downloads, Contact us, and Sign in. The main heading is "Your observability stack" in a large font, with "observability" in orange. Below this is the tagline "Operational dashboards for your data here, there, or anywhere".

On the left side, there is a section titled "The (actually useful) free forever plan" with three star icons above it. The text below reads: "Grafana, of course + 10K series", "Prometheus metrics + 50GB logs + 50GB traces". Below this text is a blue button that says "Create free account" and a smaller line of text "(No credit card required)".

In the center-right, there is a graphic showing three stacked cards: "Your data and tools" (with icons for Prometheus, Grafana, and others), "Your applications" (with a gear icon), and "Your infrastructure" (with a server rack icon). To the right of these cards is a large dashboard screenshot showing various charts and graphs. Below the dashboard are three orange buttons labeled "Metrics", "Logs", and "Traces".

At the bottom of the page, there is a grey bar with the text "MILLIONS OF USERS ACROSS 800K+ GLOBAL INSTANCES".

Fonte: Grafana (2022).

Protocolos de comunicação

Durante meus estudos em linguagem C++ e no desenvolvimento de projetos com Arduino e ESP32, deparei-me com vários protocolos de comunicação diferentes, como UART, SPI e I²C. Neste capítulo, apresentamos os elementos básicos desses protocolos, sendo específicos nas conexões entre sensores e o ESP32. No entanto, o exemplo de projeto para uso em IoT será montado com dispositivos I2C e um módulo que utiliza a comunicação UART.

UART

UART significa "Universal Asynchronous Receiver/Transmitter" e é um protocolo de comunicação serial assíncrono utilizado para enviar e receber dados entre dispositivos. É uma das formas mais simples de comunicação serial e é amplamente utilizado em dispositivos eletrônicos.

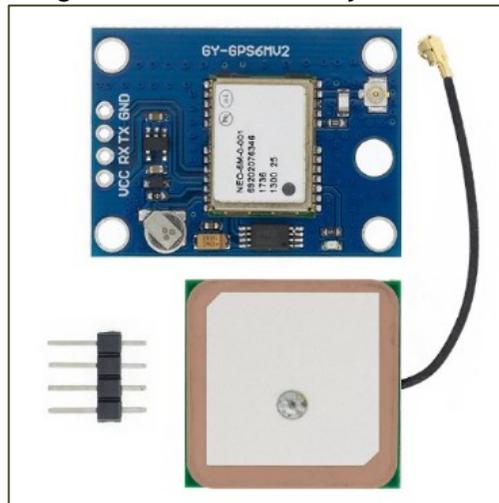
A comunicação UART é baseada em um sinal de clock, que é gerado por um dispositivo e utilizado para sincronizar a transmissão e recepção dos dados. A transmissão dos dados é feita em série, ou seja, um bit é enviado de cada vez. Os bits são agrupados em bytes e, geralmente, é utilizada uma taxa de transmissão fixa, chamada de baud rate, para definir a velocidade de transmissão.

Os dispositivos que utilizam a comunicação UART possuem dois pinos principais: TX (transmit) e RX (receive). O pino TX é utilizado para enviar dados e o pino RX é utilizado para receber dados. A comunicação UART é unidirecional, ou

seja, os dados são transmitidos em apenas uma direção de cada vez.

A comunicação UART é utilizada em uma variedade de aplicações, como comunicação entre microcontroladores, sensores e dispositivos de armazenamento de dados. Também é comumente utilizada em dispositivos de comunicação sem fio, como módulos de Bluetooth e Wi-Fi, para enviar e receber dados através de um link de rádio. A Figura 13 mostra um módulo que utiliza comunicação UART.

Figura 13 - Módulo GPS Gy GPS6MV2



Fonte: Autoria própria (2022).

SPI

SPI (Serial Peripheral Interface) é um protocolo de comunicação síncrono serial utilizado para transferência de dados entre dispositivos eletrônicos, geralmente em circuitos integrados (CI) ou microcontroladores. Ele foi desenvolvido pela Motorola na década de 1980 para ser usado em seus microcontroladores, e desde então foi adotado por muitos outros fabricantes de CI.

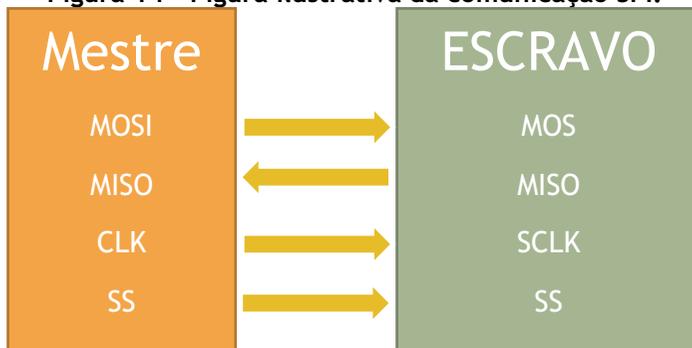
O protocolo SPI é composto por quatro linhas principais: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Clock) e SS (Slave Select). O dispositivo mestre (geralmente o microcontrolador) controla o SS para selecionar qual dispositivo escravo (ou múltiplos dispositivos escravos) será comunicado. Quando um dispositivo é selecionado, a transmissão de dados pode ocorrer em ambos os sentidos simultaneamente usando as linhas MOSI e MISO, sincronizadas pelo clock SCK.

O SPI oferece alta velocidade de transferência de dados, tornando-o adequado para aplicações que exigem comunicação de alta velocidade, como armazenamento em memória flash, controle de motores, comunicação com sensores e outros dispositivos periféricos. Ele também é relativamente fácil de implementar e tem baixo overhead de comunicação, tornando-o uma escolha popular em muitas aplicações embarcadas.

No entanto, o SPI tem algumas limitações, como a falta de um protocolo de confirmação de recebimento de dados, o que significa que o dispositivo mestre não pode confirmar se o dispositivo escravo recebeu com sucesso os dados transmitidos. Além disso, o SPI não suporta endereçamento de

dispositivos, o que pode limitar a capacidade de se comunicar com múltiplos dispositivos em um sistema. A Figura 14 apresenta uma ilustração da comunicação SPI.

Figura 14 - Figura ilustrativa da comunicação SPI.



Fonte: Autoria própria (2022).

Uma das aplicações que encontramos essa comunicação está presente nos relógios RTC (Real Time Clock).

I²C

Inter-Integrated Circuit I²C (Inter-Integrated Circuit), também conhecido como I²C ou IIC, é um protocolo de comunicação serial síncrono utilizado para a comunicação entre circuitos integrados (CI) em um sistema eletrônico. Foi desenvolvido pela empresa Philips (atualmente NXP Semiconductors) em 1982 para permitir a comunicação entre dispositivos de baixa velocidade, como sensores, conversores

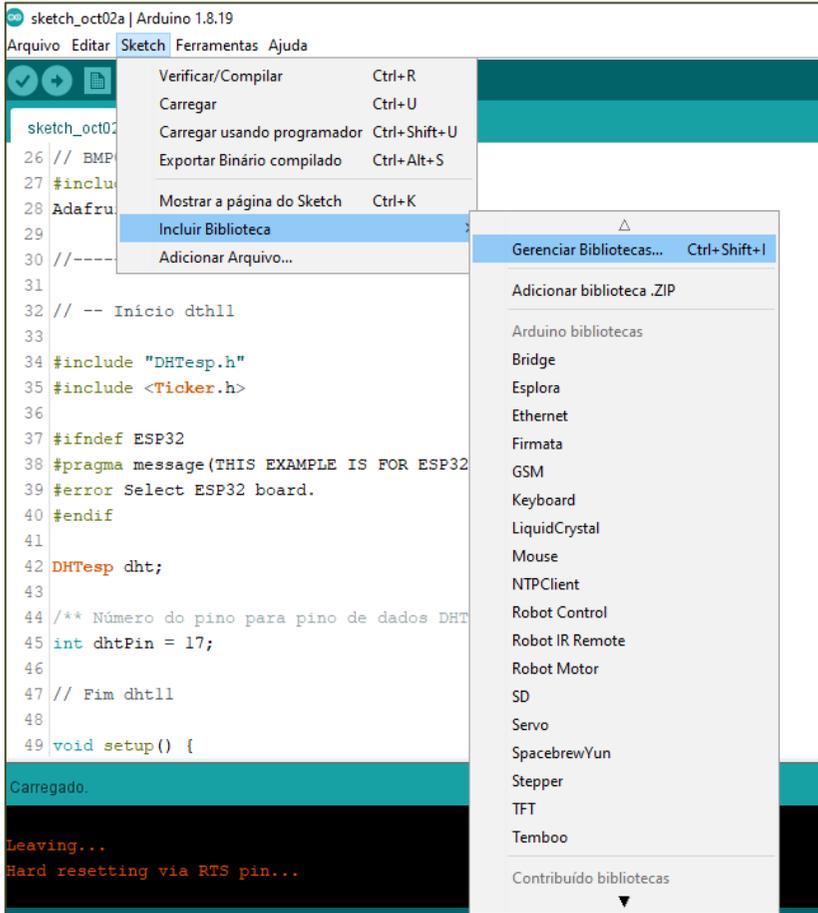
analógico-digitais (ADC) e digital-analógicos (DAC), memórias EEPROM, entre outros.

O I2C utiliza apenas duas linhas para a comunicação: a linha de dados (SDA) e a linha de clock (SCL). A linha SDA é utilizada para enviar e receber dados entre os dispositivos, enquanto a linha SCL é utilizada para sincronizar o envio e recebimento dos dados. O protocolo I2C também permite a comunicação entre vários dispositivos em um mesmo barramento, sendo que cada dispositivo possui um endereço único.

O I2C possui duas velocidades de comunicação: standard mode (100 kbps) e fast mode (400 kbps). Além disso, existem também as modalidades high-speed mode (3,4 Mbps) e ultra-fast mode (5 Mbps), porém, elas não são compatíveis com todos os dispositivos.

Uma das vantagens do protocolo I2C é a simplicidade da sua implementação e a possibilidade de conectar vários dispositivos em um mesmo barramento, reduzindo assim o número de pinos necessários para a comunicação entre os dispositivos. No entanto, o I2C pode ser suscetível a interferências eletromagnéticas (EMI) devido à sua baixa voltagem e ao uso de fios longos em aplicações com muitos dispositivos conectados.

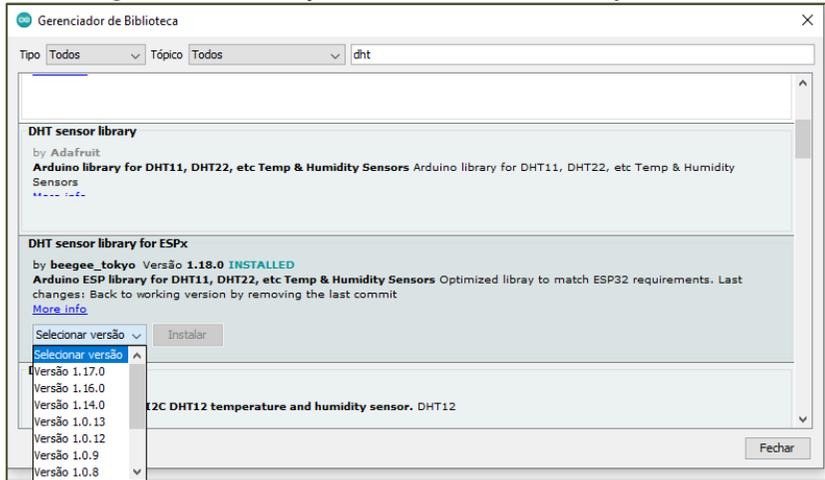
Figura 16: Caminho para adicionar as bibliotecas DHT11 e BMP180.



Fonte: Autoria própria (2022).

Na caixa de pesquisa, digite: **dht** e escolha a opção **DHT Sensor library for ESP** como mostrado na Figura 17.

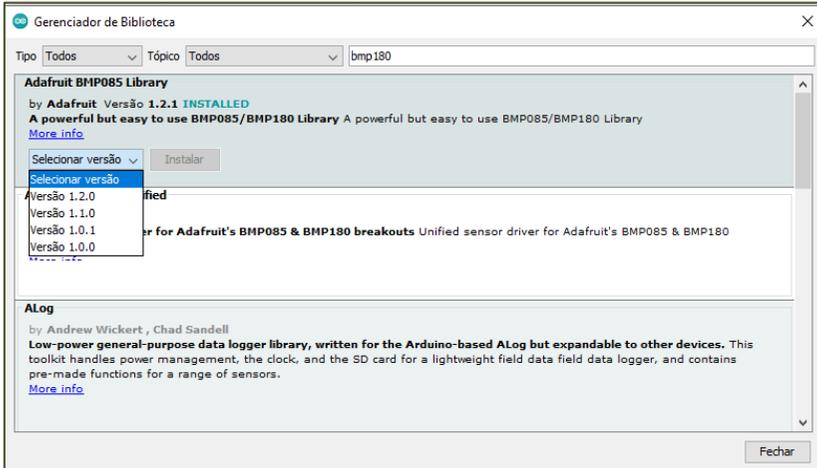
Figura 17: Instalação da biblioteca DHT11 para ESP.



Fonte: Autoria própria (2022).

Na mesma janela, agora instale a biblioteca BMP180, mas atenção: ao digitar BMP180 na pesquisa, encontrará a biblioteca Adafruit BMP085 Library e conforme a própria descrição, ela é compatível com o BMP180. A Figura 18 apresenta a situação mencionada.

Figura 18: Instalação da biblioteca BMP180.



Fonte: Autoria própria (2022).

O código comentado é apresentado na Figura 19:

Figura 19: Código comentado do ESP32 para o BMP180 e DHT11.

```
// PLACA DOIT ESP32 DEVKIT V1

//CONDIGURACAO WIFI
#include <WiFi.h>
#include <WiFiUdp.h>
#include <NTPCClient.h>

#include <HTTPClient.h>
WiFiClient client;

// DADOS DE ACESSO DA REDE WIFI
const char *ssid = "Insira seu nome de WiFi";
const char *pass = "Insira sua senha WiFi";

String ip = "mude esse valor conforme o ip do
seu computador"; // exemplo: 192.168.1.100
String site = "http://" + ip +
"/meteorologia/dados.php?";
```

```

//MODULOS I2C

/*COMUNICAÇÃO I2C
  SDA - GPIO21
  SCL - GPIO23
*/

// BMP085/BMP180
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;

//----- FIM BMP180 -----

// -- INICIO DHT11

#include "DHTesp.h"
#include <Ticker.h>

#ifndef ESP32
#pragma message(THIS EXAMPLE IS FOR ESP32 ONLY!)
#error Select ESP32 board.
#endif

DHTesp dht;

/** Número do pino para pino de dados DHT11 */
int dhtPin = 17;

// FIM DHT11

void setup() {
  Serial.begin(115200);

  // Inicialize o sensor de temperatura
  dht.setup(dhtPin, DHTesp::DHT11);
  Serial.println("DHT inicializado");

  if (!bmp.begin()) {
    Serial.println("Não foi possível encontrar
um sensor BMP085 válido, verifique a fiação!");
  }
}

```

```

    while (1) {}
}

while (!Serial)
    delay(10);

Serial.println("Conectando à ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print("."); // imprimir ... até não
estar conectado
}
Serial.println("");
Serial.println("WiFi conectado");
Serial.println("Endereço IP é: ");
Serial.println(WiFi.localIP());

Serial.println("Servidor iniciado");
}
void loop() {
    HTTPClient http; //Declare objeto da classe
HTTPClient

    // Imprime as informacoes no serial monitor
    Serial.println("Dados : ");

    //Serial.println(accelerometro);
    Serial.print("Temp= ");
    float temperaturaBMP = bmp.readTemperature();
    Serial.print(temperaturaBMP);
    Serial.print(" *C");
    Serial.print(" | ");
    Serial.print("Pressao= ");
    float pressaobmp = bmp.readPressure();
    Serial.print(pressaobmp);
    Serial.print(" Pa");
    Serial.print(" | ");

```

```

// Calcula a altitude assumindo barométrico
'padrão'
// pressão de 1013,25 milibar = 101325 Pascal
Serial.print("Altitude= ");
float altitudebmp = bmp.readPressure();
Serial.print(altitudebmp);
Serial.print(" metros");
Serial.print(" | ");
Serial.print("Ao Nível Mar(Calculada)= ");
float pressaoNivelmar =
bmp.readSealevelPressure();
Serial.print(pressaoNivelmar);
Serial.print(" Pa");
Serial.print(" | ");

// você pode obter uma medição mais precisa da
altitude
// se você conhece a pressão atual do nível do
mar que irá
// variam com o clima e tal. Se for 1015
milibares
// que é igual a 101500 Pascals.
Serial.print("Altitude Real= ");
float altitudeReal = bmp.readAltitude(101500);
Serial.print(altitudeReal);
Serial.println(" metros");

// # BMP Salvar
String leituraBMP;
leituraBMP = String(temperaturaBMP) + ";" +
String(pressaobmp) + ";" + String(altitudebmp) +
";" + String(pressaoNivelmar) + ";" +
String(altitudeReal) + ";";

// A leitura da temperatura da umidade leva
cerca de 250 milissegundos!
// As leituras do sensor também podem ter até
2 segundos de 'antiga' (é um sensor muito lento)
TempAndHumidity newValues =
dht.getTempAndHumidity();

```

```

// Verifica se alguma leitura falhou e sai
mais cedo (para tentar novamente).
if (dht.getStatus() != 0) {
    Serial.println("DHT11 error status: " +
String(dht.getStatusString()));
}

float heatIndex =
dht.computeHeatIndex(newValues.temperature,
newValues.humidity);
float dewPoint =
dht.computeDewPoint(newValues.temperature,
newValues.humidity);

Serial.println(" T:" +
String(newValues.temperature) + " H:" +
String(newValues.humidity));

Serial.println("-----
-----
-----");

String parametros;
parametros = String(site) + "&tempbmp=" +
String(temperaturaBMP) + "&pressao=" +
String(pressaobmp);
parametros += "&altitude=" +
String(altitudebmp);
parametros += "&tempdht=" +
String(newValues.temperature) + "&umidade=" +
String(newValues.humidity);
Serial.println();
Serial.println(parametros);
Serial.println();

client.println("GET " + parametros + "
HTTP/1.1");
http.begin(parametros); //Specify request
destination

```

```
int httpCode = http.GET(); //Send
the request
Serial.println(httpCode); //Print HTTP
return code
String payload = http.getString();
Serial.println(payload); //Print request
response payload

Serial.println("");

http.end(); //Fechando a conexão

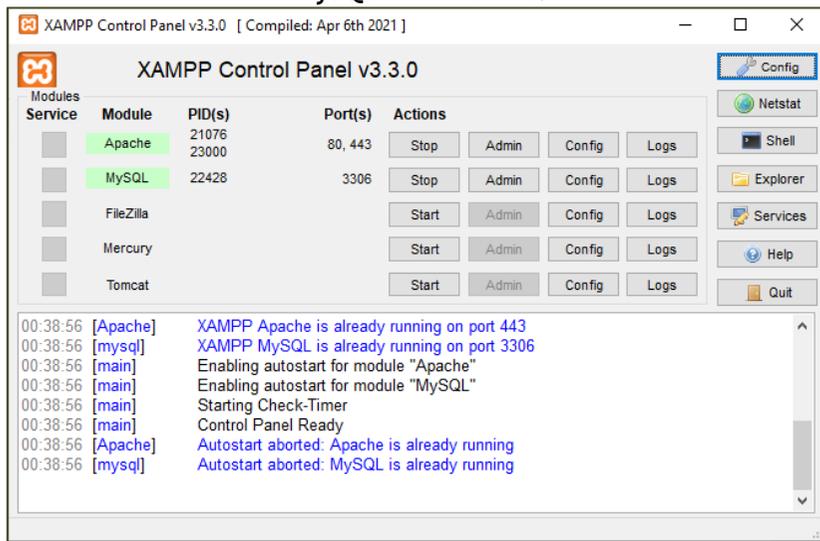
Serial.println();

Serial.println(" ----- ");

// Ajuste o intervalo de coleta de dados
delay(10000);
}
```

O próximo passo agora é criar uma tabela no banco de dados do MySQL para armazenar os dados dos sensores. Para isso vamos iniciar no Xampp o Apache e o MySQL conforme mostrado na Figura 20.

Figura 2019: Tela do Xampp onde pode-se verificar o Apache e o MySQL inicializados.



Fonte: A autoria própria (2022).

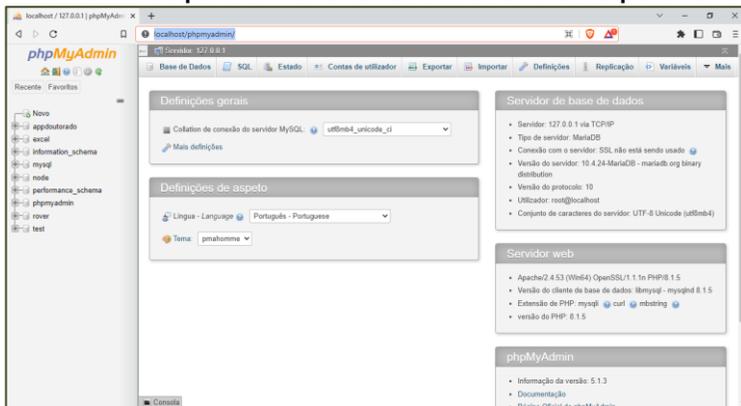
Acessando o phpmyadmin

No seu navegador preferido, insira o seguinte endereço:

<http://localhost/phpmyadmin/>

Ao digitar o endereço acima, será possível acessar uma interface web de gerenciamento do MySQL. É através dessa interface que vamos criar uma tabela para persistir os dados enviados pelo ESP32. A Figura 21 mostra a página inicial do PHPMyAdmin.

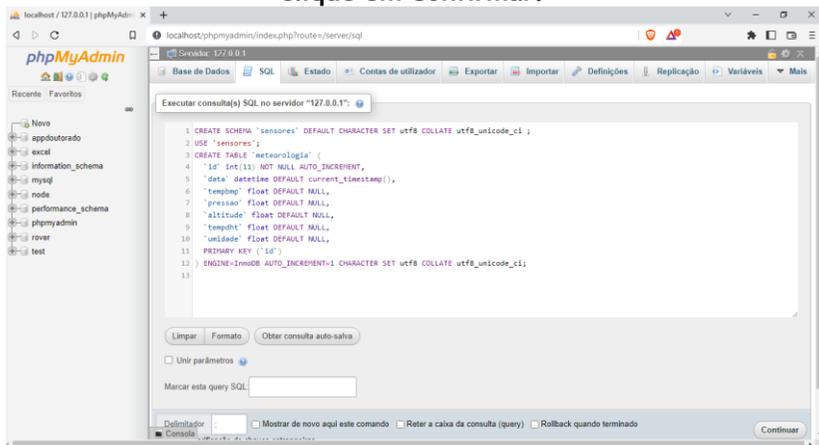
Figura 21: Interface web do phpmyadmin para criar a tabela de banco de dados para armazenar os dados enviados pelo ESP32.



Fonte: Autoria própria (2022).

Clicando em SQL como mostrado na Figura 22 insira o script SQL abaixo que irá criar um schema, ou seja, uma base chamada 'sensores' e em seguida cria uma tabela intitulada 'meteorologia' com os campos id, data, tempbmp, pressão, altitude, tempdht e umidade.

Figura 22: Clicando no menu central em SQL, copie o script SQL mostrado a seguir e insira na caixa central e para executar clique em Confirmar.



Fonte: Autoria própria (2022).

A Figura 23 apresenta o script executado no phpmysql para criação do banco de dados sensores e da tabela meteorologia.

Figura 23 - Script SQL executado no phpmysql para criar o banco de dados sensores e a tabela meteorologia

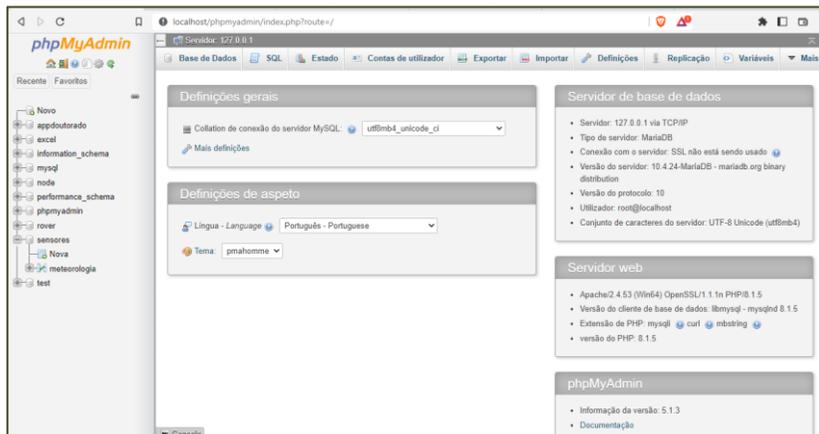
```
CREATE DATABASE IF NOT EXISTS `sensores` DEFAULT
CHARACTER SET utf8 COLLATE utf8_unicode_ci ;
USE `sensores`;
CREATE TABLE `meteorologia` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `data` datetime DEFAULT current_timestamp() ,
  `tempbp` float DEFAULT NULL,
  `pressao` float DEFAULT NULL,
  `altitude` float DEFAULT NULL,
  `tempdht` float DEFAULT NULL,
  `umidade` float DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 CHARACTER SET
utf8 COLLATE utf8_unicode_ci;
```

Fonte: A autoria própria (2022).

A persistência dos dados obtidos no ESP32 será realizada enviando uma requisição do tipo GET para uma url, um endereço no seu computador de modo a tratar os dados recebidos e os armazenar na base de dados sensores na tabela meteorologia. O passo seguinte é escrever os dois arquivos em php necessários para armazenar os valores obtidos dos sensores.

Caso receba alguma mensagem que o usuário 'root' não tem privilégio para acessar o banco de dados, haverá necessidade de liberar o acesso ao usuário root de modo que ele seja acessado no MySQL por qualquer host. Para realizar essa liberação de acesso pode ser realizada da seguinte forma: Acesse o phpmyadmin em seu navegador. Na tela inicial do phpmyadmin, clique em 'Contas de utilizador' mostrado na Figura 24 e em seguida, localize a opção 'Adicionar conta de utilizador'.

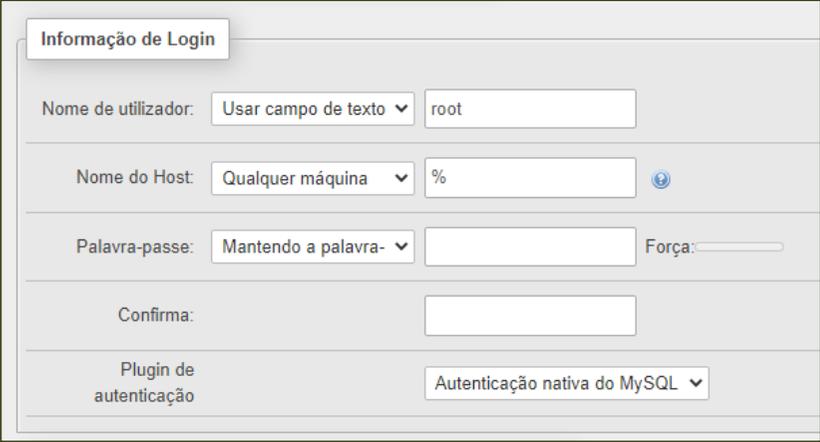
Figura 24: Procedimento para liberação de acesso ao MySQL em outros programas. Localize no meu central a opção ‘Contas de Utilizadores’.



Fonte: Autoria própria (2022).

Em Nome de utilizador, insira a palavra 'root' e em Nome do Host, escolha 'Qualquer máquina' e no campo a seguir, coloque o caracter '%'. Logo abaixo, clique em Continuar. A Figura 25.

Figura 25: Tela de cadastro do usuário root para ser acessado por qualquer outro computador em sua rede.



The image shows a user registration form titled "Informação de Login". It contains the following fields and options:

- Nome de utilizador:** A dropdown menu set to "Usar campo de texto" and a text input field containing "root".
- Nome do Host:** A dropdown menu set to "Qualquer máquina" and a text input field containing "%". A small blue icon is visible to the right of the field.
- Palavra-passe:** A dropdown menu set to "Mantendo a palavra-", a text input field, and a "Força:" strength indicator.
- Confirma:** A text input field.
- Plugin de autenticação:** A dropdown menu set to "Autenticação nativa do MySQL".

Fonte: Autoria própria (2022).

Recebendo os dados enviados pelo ESP32

O ESP32 nos envia uma requisição do tipo GET a cada 10 segundos. Recebemos esse GET em um arquivo .php que os valida, verifica se não há caracteres utilizados para realizar ataques aos bancos de dados e, caso todos estejam preenchidos, realiza a inserção desses dados no banco de dados MySQL.

Podemos chamar esse endereço que vai receber os dados vindos do ESP32 como uma API (Application Programming Interface, ou em português Interface de programação de aplicações). É composto de dois arquivos, sendo um chamado db.php que possui as credenciais de conexão com o banco MySQL e outro que trata os dados recebidos e os insere no banco.

O conteúdo do arquivo db.php é o seguinte:

Figura 26: Exemplo de string de conexão entre PHP e MySQL

```
<?php
header("Access-Control-Allow-Origin: *");
$mysqli = new mysqli('localhost', 'root', '',
'sensores');

if ($mysqli->connect_error) {
    die('Erro de conexão: (' . $mysqli-
>connect_error);
}
?>
```

Fonte: Autoria própria (2022).

O conteúdo do arquivo dados.php é o seguinte:

Figura 27: Exemplo de arquivo receptor de dados dos sensores em PHP

```
<?php
//Inclui a conexão com o BD
include 'bd.php';

$tempbmp = trim(strip_tags($_GET['tempbmp']));
$pressao = trim(strip_tags($_GET['pressao']));
$altitude = trim(strip_tags($_GET['altitude']));
$tempdht = trim(strip_tags($_GET['tempdht']));
$umidade = trim(strip_tags($_GET['umidade']));

$sql = "INSERT INTO meteorologia (tempbmp,
pressao, altitude, tempdht, umidade) VALUES
($tempbmp, $pressao, $altitude, $tempdht,
$umidade) ";

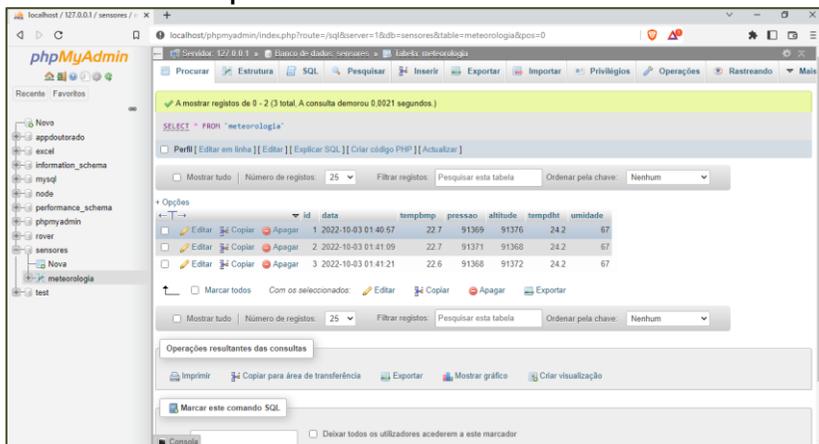
//echo $sql;
echo "Dados gravados com sucesso!";
echo "<br>";
echo "Temperatura BMP: $tempbmp";
echo "<br>";
echo "Pressão: $pressao";
echo "<br>";
echo "Altitude: $altitude";
echo "<br>";
echo "Tempdht: $tempdht";
echo "<br>";
echo "Umidade: $umidade";

if (!$mysqli->query($sql)) {
    //Gravar log de erros
    die("Erro na gravação dos dados no BD");
}
$mysqli->close();
?>
```

Fonte: Autoria própria (2022).

Clicando sobre o nome do banco de dados no phpmyadminn, poderá verificar que os dados estão sendo gravados com sucesso como mostrado na Figura 28.

Figura 28: Acessando o phpmyadmin para verificar os dados armazenados na tabela sensores.



Fonte: Autoria própria (2022).

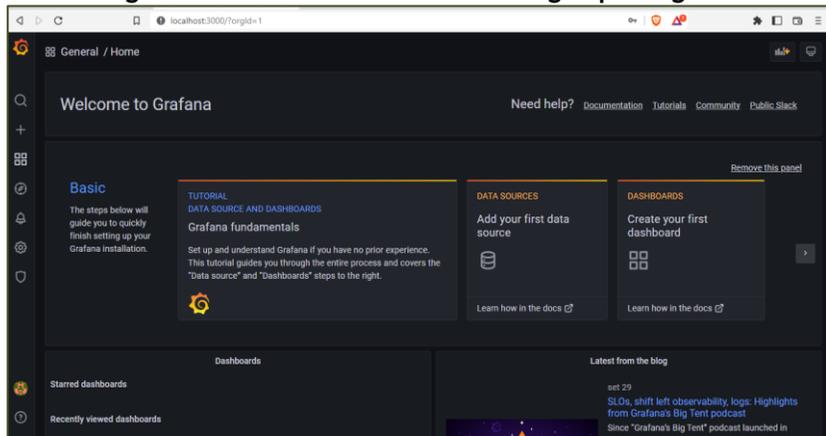
Dashboard no Grafana

No passo anterior finalizamos a persistência dos dados obtidos no ESP32. Agora, vamos criar um Dashboard para acompanhamento em tempo real dos dados armazenados no banco sensores na tabela meteorologia. Para isso, vamos abrir no navegador a seguinte url:

<http://localhost:3000/login>

Ao acessar o Grafana, vemos a seguinte tela mostrada na Figura 29:

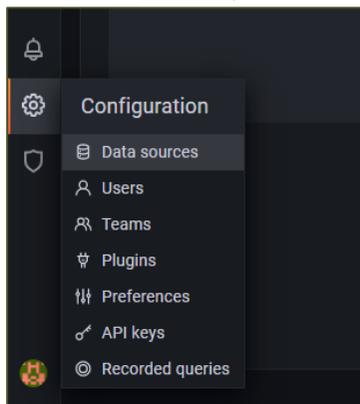
Figura 29: Tela inicial do Grafana logo após login.



Fonte: Autoria própria (2022).

Clique no lado esquerdo no ícone da engrenagem e escolha Data Source como mostrado na Figura 30.

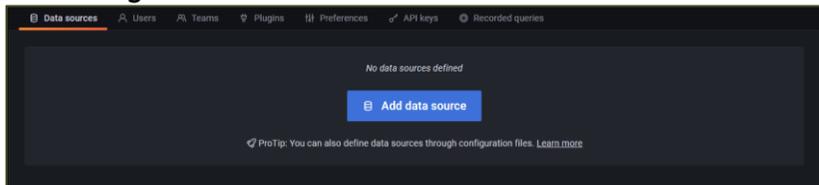
Figura 30: Configuração do bando de dados no aplicativo Grafana.



Fonte: Autoria própria (2022).

Em seguida, clique em Add data source indicado na Figura 31.

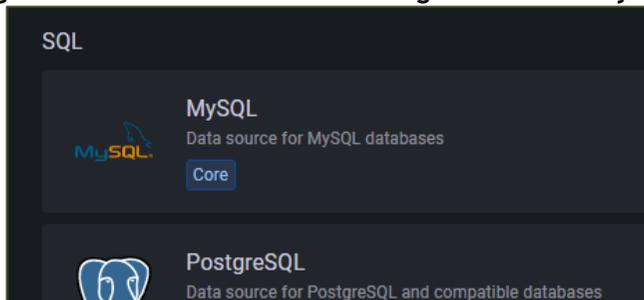
Figura 3120: Adicionar um novo banco de dados.



Fonte: Autoria própria (2022).

Localize o MySQL como é apresentado Figura 32.

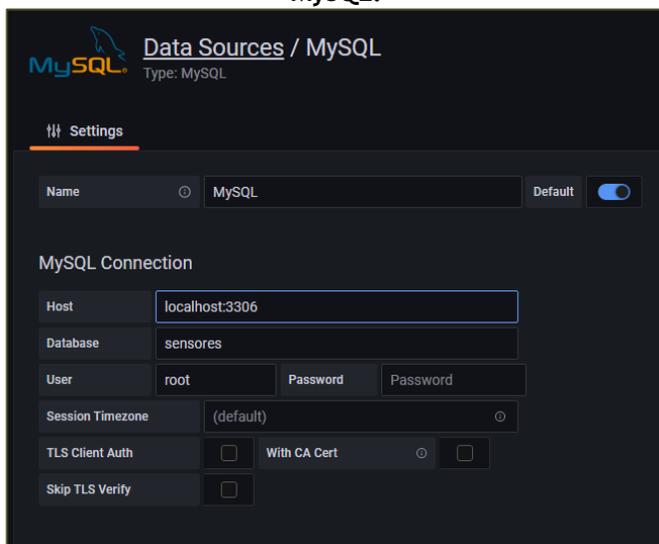
Figura 32: Descendo a barra de rolagem encontre MySQL.



Fonte: Autoria própria (2022).

Em seguida, informe os seguintes dados como mostrado na Figura 33:

Figura 33: Inserção das credenciais de acesso ao banco de dados MySQL.

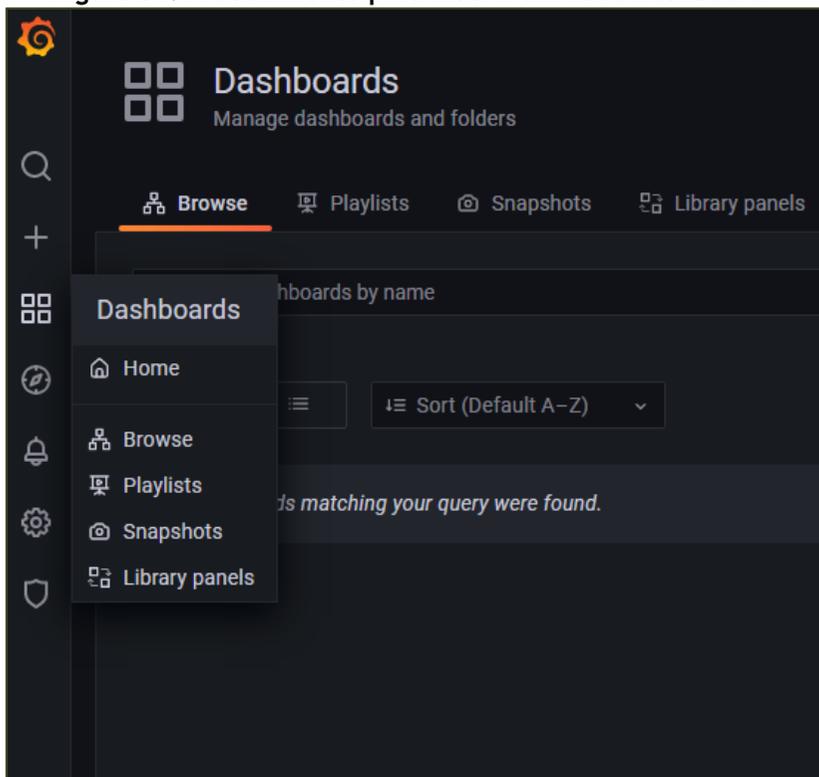


Fonte: Autoria própria (2022).

Observe que a senha ficou em branco. Como instalamos o Xampp, o usuário padrão do MySQL é o root e sua senha é em branco.

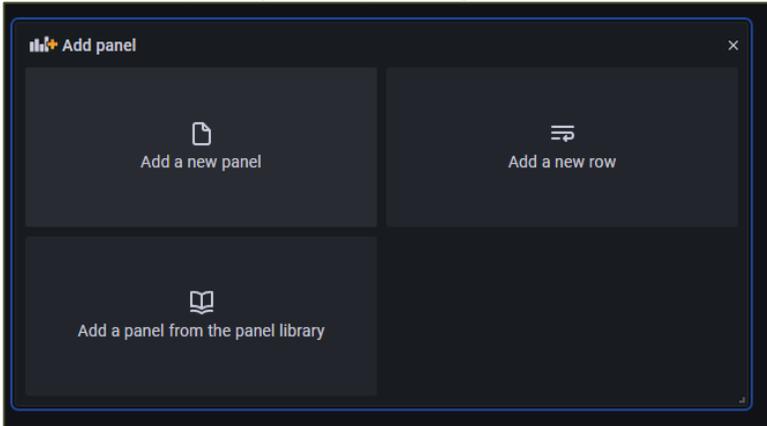
Em Dashboard, clique em Browse como é mostrado a Figura 34 clique em New Dashboard e em seguida, Add a new panel Figura 35.

Figura 34: Procedimento para inserir um novo Dashboard



Fonte: Aatoria própria (2022).

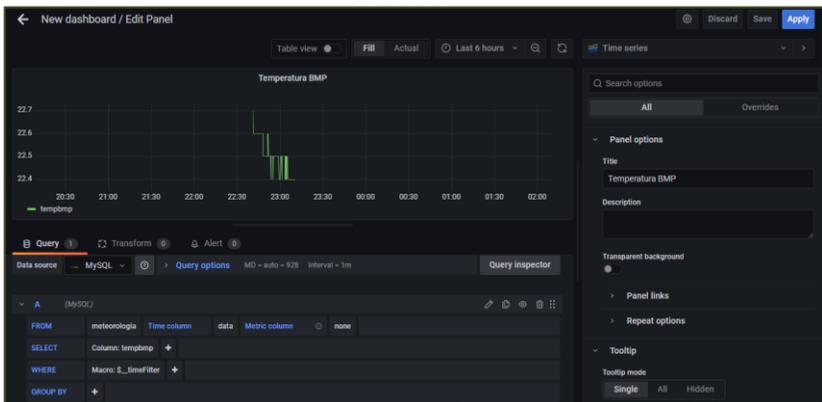
Figura 35: Clique em Add a new panel para seleccionar os dados do banco MySQL que deseja acompanhar no Dashboard.



Fonte: Autoria própria (2022).

Informe na linha do Select qual coluna deseja apresentar, na Figura 36 e clique em Apply.

Figura 36: Configuração para exibir os dados da temperatura do BMP180.



Fonte: Autoria própria (2022).

Fazendo o mesmo procedimento para os demais sensores, pode-se obter uma interface que nos permite acompanhar os dados dos sensores em tempo real.

Bibliografia

ARDUINO. **A Guide to Arduino & the I2C Protocol (Two Wire)**. Disponível em:
<https://docs.arduino.cc/learn/communication/wire>. Acesso em: 05 jun. 2022.

_____. **Arduino & Serial Peripheral Interface (SPI)**. Disponível em:
https://docs.arduino.cc/learn/communication/spi?_gl=1*tz1r21*_ga*MzcxMTcwODg2LjE2NjQ3NTAyNDQ.*_ga_NEXN8H46L5*MTY2NDc1MDI0NC4xLjEuMTY2NDc1MDU3Ni4wLjAuMA. Acesso em: 05 jun. 2022.

_____. **Communication**. Disponível em:
<https://www.arduino.cc/reference/pt/language/functions/communication/serial/>. Acesso em: 05 jun. 2022.

_____. **Página inicial**. Disponível em:
<https://www.arduino.cc>. Acesso em: 05 jun. 2022.

_____. **SPI**. Disponível em:
<https://www.arduino.cc/reference/en/language/functions/communication/spi/>. Acesso em: 05 jun. 2022.

_____. **Software**. Disponível em:
<https://www.arduino.cc/en/software>. Acesso em: 15 set. 2022.

GRAFANA. **GrafanaLabs**. Disponível em:
<https://grafana.com/>. Acesso em: 16 set. 2022.

MAXIM INTEGRATED. Determining Clock Accuracy Requirements for SCC269. Universal asynchronous receiver/transmitter. Disponível em: <https://www.nxp.com/docs/en/data-sheet/SCC2691.pdf#page=14>. Acesso em: 05 jun. 2022.

NXP. UM10204 I2C-bus specification and user manual. Disponível em: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. Acesso em: 05 jun. 2022.

SPI: Block Guide. Disponível em: <https://web.archive.org/web/20150413003534/http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf>. Acesso em: 05 jun. 2022.

UART. Communications. Disponível em: <https://pdfserv.maximintegrated.com/en/an/AN2141.pdf>. Acesso em: 05 de jun. de 2022.

XAMPP. Xampp Apache + MariaDB + PHP + Perl. Página inicial. Disponível em: https://www.apachefriends.org/pt_br/index.html. Acesso em: 05 jun. 2022.

Autores

Hernani Batista da Cruz



Sani de Carvalho Rutz da Silva



Silvio Luiz Rutz da Silva

