Associação paranaense de cultura

Centro de educação profissional irmão mário cristóvão

Curso ENGENHARIA DE SOFTWARE

SISTEMA DE METEOROLOGIA

Curitiba

JUNHO/2024

Delvity Taverni Chevonica

FELIPE NATHAN mendes brito

VINÍCIUS de oliveira garcia

mateus zanettin DALL’ AGNOL

MATHEUS SILVA DA SILVA

SISTEMA DE METEOROLOGIA

Trabalho de TDE apresentado ao Curso Engenharia de Software do Centro Educação Profissional Irmão Mário Cristóvão vinculado à Associação Paranaense de Cultura

Orientadores: Marcelo do Carmo Camargo Gaiotto

Curitiba

junho/2024

Delvity Taverni Chevonica

FELIPE NATHAN mendes brito

mateus zanettin DALL’ AGNOL

MATHEUS SILVA DA SILVA

VINÍCIUS de oliveira garcia

SISTEMA DE METEOROLOGIA

Trabalho de TDE apresentado ao Curso Engenharia de Software do Centro Educação Profissional Irmão Mário Cristóvão.

Comissão Examinadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Marcelo do Carmo Camargo Gaiotto

Centro de Educação

Curitiba, 16 de Junho de 2024

# AGRADECIMENTO

Neste momento, em um sentimento profundo de gratidão, desejamos expressar nossos sinceros agradecimentos a Deus por nos conceder a oportunidade ímpar de servir e auxiliar aqueles que necessitam de nosso conhecimento e habilidades.

Além disso, queremos estender nossos agradecimentos a todos os envolvidos que tornaram possível a concretização deste trabalho significativo. Nesse gesto de reconhecimento, não podemos deixar de mencionar nossos queridos familiares e amigos, cujo apoio incondicional nos acompanhou nos momentos de maior tensão, proporcionando-nos força e alento.

Igualmente, expressamos nossa gratidão a todos os colegas do estimado Curso Engenharia de Software, cuja colaboração e parceria foram essenciais para a construção de uma jornada enriquecedora.

E, com imensa gratidão, reconhecemos a dedicação e orientação do nosso respeitado orientador, o senhor Marcelo do Carmo Camargo Gaiotto. Sua sabedoria, paciência e comprometimento foram pilares fundamentais no desenvolvimento deste trabalho.

Por fim, desejamos compartilhar essa gratidão com todos aqueles que, de alguma forma, contribuíram para este projeto, sabendo que suas influências e apoio foram essenciais para a concretização de nossos esforços.

Que este sentimento de gratidão perdure, e que possamos, com humildade e dedicação, seguir buscando auxiliar e servir àqueles que cruzarem nossos caminhos, honrando o aprendizado e experiência adquiridos nesta jornada.

# RESUMO

Cidades inteligentes, com seus sistemas complexos de infraestrutura e serviços, precisam de um entendimento preciso e em tempo real do clima para funcionar de forma eficiente e sustentável. É aí que os medidores meteorológicos entram em cena, atuando como os olhos e ouvidos das cidades inteligentes, coletando dados sobre temperatura, umidade, pressão atmosférica, direção e velocidade do vento, entre outros.

Essa informação crucial permite a otimização de diversos aspectos da vida urbana. Imagine a gestão de energia: dados precisos sobre temperatura e insolação permitem ajustar o funcionamento de sistemas de ar condicionado e iluminação pública, reduzindo o consumo de energia e as emissões de CO2. Ou a gestão de recursos hídricos: a previsão de chuvas permite otimizar o uso de sistemas de irrigação em parques e jardins, evitando desperdício e garantindo a saúde da vegetação urbana.

Além disso, os medidores meteorológicos contribuem para a segurança e bem-estar da população. O monitoramento em tempo real da temperatura, umidade e precipitação permite otimizar o fluxo de tráfego, prevenindo congestionamentos e garantindo a segurança dos motoristas. A previsão de ondas de calor, tempestades e outros eventos climáticos permite a tomada de medidas preventivas para proteger a saúde da população.

Em suma, a utilização de medidores meteorológicos em cidades inteligentes é essencial para a tomada de decisões mais eficientes e eficazes, contribuindo para a otimização de recursos, a segurança da população e a qualidade de vida em áreas urbanas. A coleta precisa e em tempo real de dados meteorológicos permite que as cidades se adaptem de forma mais inteligente às condições climáticas, garantindo um futuro mais sustentável e resiliente.

# LISTA DE ILUSTRAÇÕES

[FIGURA 1 - ARDUINO UNO R3 12](#_Toc148839439)

[FIGURA 2 - ETHERNET SHIELD W5100 12](#_Toc148839440)

[FIGURA 3 - SENSOR TSL2561 13](#_Toc148839441)

[FIGURA 4 - SENSOR BME 280 14](#_Toc148839442)

[FIGURA 5 - SENSOR ML8511 15](#_Toc148839443)

[FIGURA 6 - PLUVIOMÊTRO 16](#_Toc148839444)

[FIGURA 7 – ANEMOMÊTRO WH-SP-WS01 17](#_Toc148839445)

[FIGURA 8 - RESISTOR 10K 17](#_Toc148839446)

[FIGURA 9 - PROTOBOARD 18](#_Toc148839447)

[FIGURA 10 - JUMPERS 19](#_Toc148839448)

[FIGURA 11 - SQLServer 21](#_Toc148839449)

[Figura 12 - Modelo do Fritzing 31](#_Toc148839450)

# SUMÁRIO

[FOLHA DE APROVAÇÃO i](#_Toc39486691)

[AGRADECIMENTO II](#_Toc143388778)

[RESUMO III](#_Toc143388779)

[LISTA DE ILUSTRAÇÕES IV](#_Toc143388780)

[SUMÁRIO V](#_Toc143388781)

[1. INTRODUÇÃO 10](#_Toc143388782)

[2. FUNDAMENTAÇÃO TEÓRICA 12](#_Toc143388783)

[2.1. NEGÓCIO 12](#_Toc143388784)

[2.1.1. O problema da precariedade de dados Meteorológicos precisos 12](#_Toc143388785)

[2.1.2. O problema de aproveitamento da safra 13](#_Toc143388786)

[2.2. TECNOLOGIAS 14](#_Toc143388787)

[2.2.1. Hardware 14](#_Toc143388788)

[2.2.1.1. Arduino Uno R3 14](#_Toc143388789)

[2.2.1.2. Ethernet Shield W5100 15](#_Toc143388790)

[2.2.1.3. Sensor TSL2561 16](#_Toc143388791)

[2.2.1.4. Sensor BME280 17](#_Toc143388792)

[2.2.1.5. Sensor ML8511 18](#_Toc143388793)

[2.2.1.6. Pluviômetro 19](#_Toc143388794)

[2.2.1.7. Anemômetro WH-SP-WS01 20](#_Toc143388795)

[2.2.1.8. Resistor 10K 21](#_Toc143388796)

[2.2.1.9. Diversos 21](#_Toc143388797)

[2.2.2. Software 24](#_Toc143388798)

[2.2.2.1. Linguagem C++ 24](#_Toc143388799)

[2.2.2.2. MySQL 25](#_Toc143388800)

[2.2.2.3. JavaScript 26](#_Toc143388801)

[2.2.2.4. Linguagem HTML 26](#_Toc143388802)

[2.2.2.5. Linguagem CSS 27](#_Toc143388803)

[2.2.2.6. Vue 27](#_Toc143388804)

[2.2.2.7. C# 28](#_Toc143388805)

[2.2.2.8. TailWind 28](#_Toc143388806)

[2.2.2.9. Entity Framework 29](#_Toc143388807)

[3. DESENVOLVIMENTO 30](#_Toc143388808)

[3.1. TESTE DOS SENSORES E PLACA ETHERNET 30](#_Toc143388809)

[3.1.1. Teste Placa Ethernet 30](#_Toc143388810)

[3.1.2. Teste Sensor ML8511 30](#_Toc143388811)

[3.1.3. Teste Sensor BME280 30](#_Toc143388812)

[3.1.4. Teste Sensor TSL2561 31](#_Toc143388813)

[3.2. BACK-END 31](#_Toc143388814)

[3.2.1. Instalação do Banco de dados no Azure 31](#_Toc143388815)

[3.2.2. Configuração do SQL Server 31](#_Toc143388816)

[3.2.3. Criação de Tabelas no SQL Server 32](#_Toc143388817)

[3.2.4. Implementação do CRUD 32](#_Toc143388818)

[3.2.5. Estabelecimento do padrão MVC 32](#_Toc143388819)

[3.2.6. Programação do sistema de login 32](#_Toc143388820)

[3.3. FRONT-END 33](#_Toc143388821)

[3.3.1. Programação Home Page e Contatos 33](#_Toc143388822)

[3.4. CONSTRUÇÃO 33](#_Toc143388823)

[3.4.1. Impressão 3D do Pluviômetro 33](#_Toc143388824)

[3.4.2. Design e Modelagem 33](#_Toc143388825)

[3.4.3. Montagem 34](#_Toc143388826)

[3.4.3.1. Desenvolvimento do cano PVC Modificado 35](#_Toc143388827)

[4. RESULTADOS 35](#_Toc143388828)

[5. CONSIDERACOES FINAIS 35](#_Toc143388829)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 36](#_Toc143388830)

[APÊNDICE 01 – CÓDIGO FONTE DO ETHERNET SHIELD W5100 39](#_Toc143388831)

[APÊNDICE 02 – CÓDIGO FONTE ML8511 42](#_Toc143388832)

[APÊNDICE 03 – CÓDIGO FONTE BME280 44](#_Toc143388833)

[APÊNDICE 04 – CÓDIGO FONTE TSL2561 45](#_Toc143388834)

[APÊNDICE 05 – CÓDIGO ANEMOMÊTRO 47](#_Toc143388835)

[APÊNDICE 06 – CÓDIGO FONTE COM INTEGRAÇÃO DOS PRINCIPAIS SENSORES 48](#_Toc143388836)

# INTRODUÇÃO

O clima e as condições meteorológicas desempenham um papel fundamental e onipresente em inúmeras esferas de nossa vida, exercendo um impacto direto e abrangente em diferentes setores. Desde a essencial agricultura, que depende de chuvas e temperaturas adequadas para garantir colheitas saudáveis e abundantes, até a imprescindível aviação, que requer informações meteorológicas precisas para garantir operações seguras e eficientes, a compreensão acurada do clima se torna indispensável para o sucesso de diversas atividades humanas.

Contudo, a coleta de dados meteorológicos precisos e confiáveis frequentemente se apresenta como um desafio complexo e multifacetado. A obtenção desses dados pode ser dificultada por fatores variados, como a distribuição espacial irregular das estações meteorológicas, a natureza dinâmica e imprevisível das condições atmosféricas e as limitações tecnológicas de alguns métodos de coleta. Essa situação crítica acarreta a possibilidade de tomadas de decisões ineficientes ou, em casos mais graves, potencialmente perigosas, prejudicando tanto atividades cotidianas quanto processos industriais e produtivos.

A resposta a esse desafiante dilema emerge na forma de um medidor meteorológico de vanguarda. Esse instrumento inestimável e altamente sofisticado é habilmente projetado para medir uma extensa gama de variáveis meteorológicas cruciais, tais como temperatura, umidade, pressão atmosférica, direção e velocidade do vento, dentre outras. Com uma precisão ímpar e a capacidade de fornecer informações em tempo real, o medidor meteorológico se torna uma ferramenta poderosa, habilitando as pessoas a tomarem decisões informadas e altamente eficazes em diversas áreas de atuação.

No cerne desse projeto promissor reside a busca pela concepção e desenvolvimento de um medidor meteorológico de alta precisão e confiabilidade, com a versatilidade de se adaptar a diferentes contextos e cenários. O design minuciosamente elaborado pauta-se na acessibilidade, facilidade de uso e baixo custo, almejando permitir que uma vasta gama de usuários, desde especialistas até leigos, possa usufruir desse recurso tecnológico com destreza e proveito em variadas situações cotidianas ou profissionais.

A realização bem-sucedida desse audacioso medidor meteorológico acarretará um significativo avanço na capacidade de coletar informações detalhadas, abrangentes e confiáveis sobre as condições climáticas de um local específico. Essa conquista grandiosa e multifuncional beneficiará uma ampla variedade de atividades que são diretamente influenciadas pelo clima e pelas condições atmosféricas. Setores cruciais, como agricultura, pesca, transporte, construção civil, aviação e previsão do tempo, entre outros, obterão maior segurança, eficiência e eficácia em suas operações, resultando em impactos positivos e substanciais para a sociedade como um todo. O futuro se desenha promissor, impulsionado pelo poder da tecnologia meteorológica, que rege de forma visionária e inteligente os rumos de nossa jornada terrestre.

# FUNDAMENTAÇÃO TEÓRICA

## NEGÓCIO

Neste capítulo, abordaremos a questão da precariedade dos dados meteorológicos precisos e os obstáculos enfrentados no aproveitamento das safras devido à carência de informações meteorológicas confiáveis.

### O problema da precariedade de dados Meteorológicos precisos

O medidor meteorológico, pode resolver o problema de coleta de dados precisos e confiáveis sobre as condições meteorológicas de um determinado local. Ele pode ajudar a medir a temperatura, umidade, pressão atmosférica, direção e velocidade do vento, entre outras variáveis meteorológicas.

Com essas informações em mãos, as pessoas podem tomar decisões mais informadas e eficazes em diversas áreas. A coleta de informações precisas sobre as condições meteorológicas em um determinado local é essencial para várias atividades que dependem do clima e das condições atmosféricas. Agricultura, aviação, construção, previsão do tempo e muitas outras áreas requerem dados meteorológicos precisos para tomar decisões informadas e eficazes. Sem essas informações, podem ocorrer riscos à segurança, perda de recursos e ineficiência em diversas atividades.

### O problema de aproveitamento da safra

É o uso dos dados meteorológicos pelos agricultores, que podem fazer escolhas cruciais, como decidir o momento exato para plantar e colher suas safras. Essas informações meteorológicas também desempenham um papel vital na construção de infraestruturas, como pontes e estradas, garantindo que essas estruturas sejam seguras e resilientes às adversidades climáticas.

Em síntese, o medidor meteorológico assume um papel de destaque como uma ferramenta inestimável na solução do desafio de coletar dados meteorológicos precisos e confiáveis em locais específicos. Com tais informações em suas mãos, as pessoas estão aptas a tomar decisões embasadas e altamente efetivas em diversas esferas, salvaguardando a segurança, eficiência e desempenho de suas atividades.

## TECNOLOGIAS

Neste capítulo, investigamos as tecnologias utilizadas, que englobam desde os componentes físicos até os softwares adotados, realizando uma análise minuciosa de cada um desses elementos.

### Hardware

Neste capítulo, exploramos os aspectos relacionados ao hardware do projeto, com ênfase especial no microcontrolador e no sensor. Além disso, apresentamos diversos componentes menores amplamente reconhecidos, como LEDs e buzzers, que desempenham papéis essenciais no sistema.

#### Arduino uno R3

O Arduino Uno R3 é uma das placas de desenvolvimento mais populares e amplamente utilizadas no mundo da eletrônica e da programação de microcontroladores. Com base no microcontrolador ATmega328P da Atmel, esta placa oferece uma plataforma acessível e versátil para entusiastas e profissionais que desejam criar uma ampla variedade de projetos eletrônicos. O Arduino Uno R3 é conhecido por sua simplicidade de uso, o que o torna uma escolha excelente para iniciantes. Ele possui uma variedade de pinos de entrada/saída digital e analógica que podem ser facilmente programados usando a linguagem de programação Arduino, e sua ampla comunidade de usuários e recursos online oferece suporte e tutoriais para ajudar os criadores a dar vida às suas ideias.

Além disso, o Arduino Uno R3 é altamente expansível, permitindo que os usuários conectem módulos adicionais, sensores e atuadores para personalizar seus projetos de acordo com suas necessidades específicas. Sua combinação de facilidade de uso, acessibilidade e flexibilidade torna o Arduino Uno R3 uma escolha valiosa para projetos que variam desde robótica simples até sistemas de automação residencial complexos e dispositivos de monitoramento ambiental. Seja você um novato em eletrônica ou um engenheiro experiente, o Arduino Uno R3 oferece um ponto de partida sólido e confiável para transformar suas ideias em realidade.

FIGURA 1 - ARDUINO UNO R3



FONTE: **Arduino Uno R3**. Baú da Eletrônica.

#### Ethernet Shield W5100

O Ethernet Shield W5100 é um componente que viabiliza a conexão da placa Arduino com a internet, permitindo a transmissão online de dados coletados por sensores. Para utilizá-lo, basta conectá-lo ao Arduino e ao modem/roteador por meio de um cabo ethernet com conectores RJ45.

O shield é composto por um chip ethernet W5100, fabricado pela WIZnet, que possibilita o acesso à rede IP através dos protocolos TCP ou UDP. Suas funcionalidades são configuradas através das bibliotecas Ethernet Library e SD Library, e ele suporta até quatro conexões independentes simultâneas. Com essa capacidade, torna-se uma solução poderosa para projetos que requerem comunicação online, agregando recursos e conectividade à plataforma Arduino.

FIGURA 2 - ETHERNET SHIELD W5100



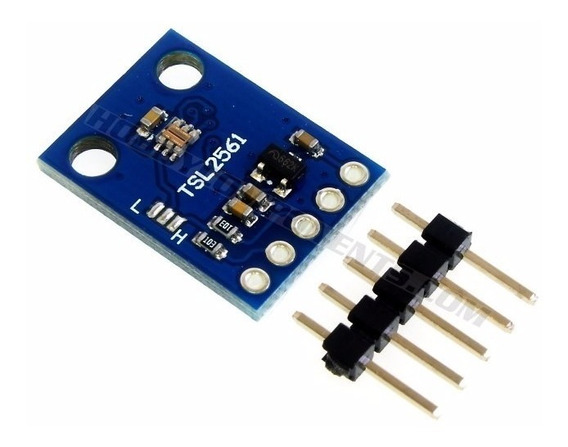
FONTE: **Ethernet Shield W5100**. FilipeFlop.

#### Sensor TSL2561

O sensor TSL2561 é um dispositivo usado para medir a intensidade de luz em ambientes. Ele é baseado na tecnologia de detecção de luz digital e é capaz de medir tanto a luz visível quanto a luz infravermelha, o que o torna adequado para uma variedade de aplicações. O TSL2561 oferece uma alta resolução de medição, permitindo que os usuários obtenham informações detalhadas sobre os níveis de luz em diferentes condições.

Esse sensor é amplamente utilizado em sistemas de iluminação inteligente, dispositivos de controle de exposição em câmeras, aparelhos eletrônicos e muitas outras aplicações onde a detecção precisa de luz é fundamental. Sua capacidade de medir diferentes faixas de luz torna o TSL2561 uma escolha versátil para projetos que envolvam monitoramento e controle de iluminação.

FIGURA 3 - SENSOR TSL2561



FONTE: **Sensor De Luz Luxímetro Tsl2561 Arduino**. Mercado Livre.

#### Sensor BME280

O sensor BME280 é um componente ambiental versátil, capaz de medir temperatura, umidade e pressão atmosférica com alta precisão. Sua faixa de medição abrange desde -40°C a 85°C para temperatura ambiente, 0% a 100% para umidade relativa do ar e 300hPa a 1100hPa para pressão atmosférica. Ele se comunica com a placa Arduino através dos protocolos I2C ou SPI, e destaca-se por um baixo consumo de energia, tornando-o ideal para projetos que utilizam dispositivos alimentados por bateria.

Com dimensões compactas, o sensor BME280 é uma escolha conveniente para projetos que exigem economia de espaço. Sua notável precisão nas medições o torna uma opção valiosa para diversas aplicações.

FIGURA 4 - SENSOR BME 280



FONTE: **Barômetro BME280 | Sensor de Pressão Umidade Temperatura GY-BME**. EasyTronics.

#### Sensor ML8511

O sensor ML8511 é um dispositivo ultravioleta projetado para trabalhar em conjunto com placas Arduino, permitindo medir a intensidade da radiação UV. Suas principais características incluem a capacidade de medir radiação UV na faixa de comprimento de onda de 280 a 390 nm, fornece uma saída analógica proporcional à intensidade detectada e possuir alta sensibilidade para capturar pequenas variações na radiação.

Além disso, o sensor possui baixo consumo de energia, facilitando sua alimentação diretamente pela placa Arduino, e não requer circuitos externos complicados para operação. Ele também conta com um filtro que protege contra a luz visível, permitindo a detecção precisa apenas da radiação UV, tornando-o uma escolha versátil com aplicações diversas, incluindo detecção de raios UV prejudiciais, monitoramento de exposição solar e controle de qualidade de produtos sensíveis à radiação UV.

FIGURA 5 - SENSOR ML8511

Uma imagem contendo circuito

Descrição gerada automaticamente

FONTE: **Sensor Ultravioleta GYML8511**. EasyTronics.

#### Pluviômetro

O pluviômetro é um dispositivo de extrema importância na área da meteorologia, sendo utilizado para medir a quantidade de precipitação, como chuva e neve, que ocorre em uma região específica. Seu funcionamento é simples: um recipiente cilíndrico ou cônico coleta a água da chuva em sua abertura, e após o período de coleta, a quantidade acumulada é medida e registrada. Essas informações são cruciais para os meteorologistas, hidrólogos e cientistas em geral, pois permitem estudar padrões climáticos, realizar previsões climáticas, planejar ações agrícolas, monitorar eventos extremos e entender a disponibilidade de recursos hídricos em determinada área.

Com base nos dados coletados pelo pluviômetro, é possível analisar a média histórica de precipitação em uma localidade, identificar variações sazonais e anomalias climáticas ao longo do tempo, além de auxiliar na gestão sustentável dos recursos naturais. A combinação dessas informações com outras tecnologias, como radares meteorológicos e satélites, amplia ainda mais nosso entendimento do clima e seus impactos na sociedade e no meio ambiente. Assim, o pluviômetro desempenha um papel vital na obtenção de dados precisos sobre a precipitação, possibilitando uma melhor compreensão do clima e contribuindo para a tomada de decisões mais informadas em diversas áreas.

FIGURA 6 - PLUVIOMÊTRO



FONTE: **Pluviômetro Automático PI3**. MercadoLivre.

#### Anemômetro WH-SP-WS01

O anemômetro é um dispositivo amplamente utilizado para medir a velocidade do vento em uma determinada área. Essa ferramenta essencial na meteorologia possui diferentes tipos, mas o mais comum é o anemômetro de copo, composto por três ou quatro conchas em forma de copo montadas em braços que giram com a ação do vento. A rotação das conchas é convertida em uma velocidade angular, permitindo calcular a velocidade do vento com base na taxa de rotação. Outro tipo popular é o anemômetro de hélice, em que a velocidade do vento gira uma hélice que está conectada a um sensor eletrônico para fornecer leituras precisas.

Esses dados são essenciais para diversos campos, como meteorologia, aviação, agricultura e energia eólica, contribuindo para previsões climáticas, planejamento de cultivos, controle de sistemas de energia e tomada de decisões em várias indústrias. Em suma, o anemômetro desempenha um papel fundamental na obtenção de informações sobre a velocidade do vento, tornando-se uma ferramenta indispensável para o entendimento do clima e suas aplicações práticas. Sua aplicação abrangente oferece suporte em várias áreas, auxiliando na busca por soluções sustentáveis e eficientes, onde o vento é uma importante fonte de energia renovável.

FIGURA 7 – ANEMOMÊTRO WH-SP-WS01



FONTE:**WH-SP-WS01 anemômetro**. Aliexpress.

#### Resistor 10K

Um resistor de 10k ohms, também conhecido como resistor de 10.000 ohms ou resistor de 10 kiloohms, é um componente eletrônico passivo usados para controlar o fluxo de corrente elétrica em um circuito. Ele possui uma resistência de 10.000 ohms, o que significa que limita a quantidade de corrente que pode fluir através dele. Resistor de 10k ohms é frequentemente usado em uma variedade de aplicações, como divisores de tensão, filtros, polarização de transistores e muitos outros circuitos eletrônicos. A cor do código de cores típico para um resistor de 10k ohms é marrom-preto-laranja, que representa os dígitos 1, 0, e um fator multiplicador de 1000. Isso indica uma resistência de 10.000 ohms com uma tolerância padrão.

Resumindo, o resistor de 10k ohms é um componente fundamental para ajustar e direcionar o fluxo de corrente em uma variedade de dispositivos eletrônicos.

FIGURA 8 - RESISTOR 10K



FONTE: **Resistor 10K 1/4W**. Marino Store.

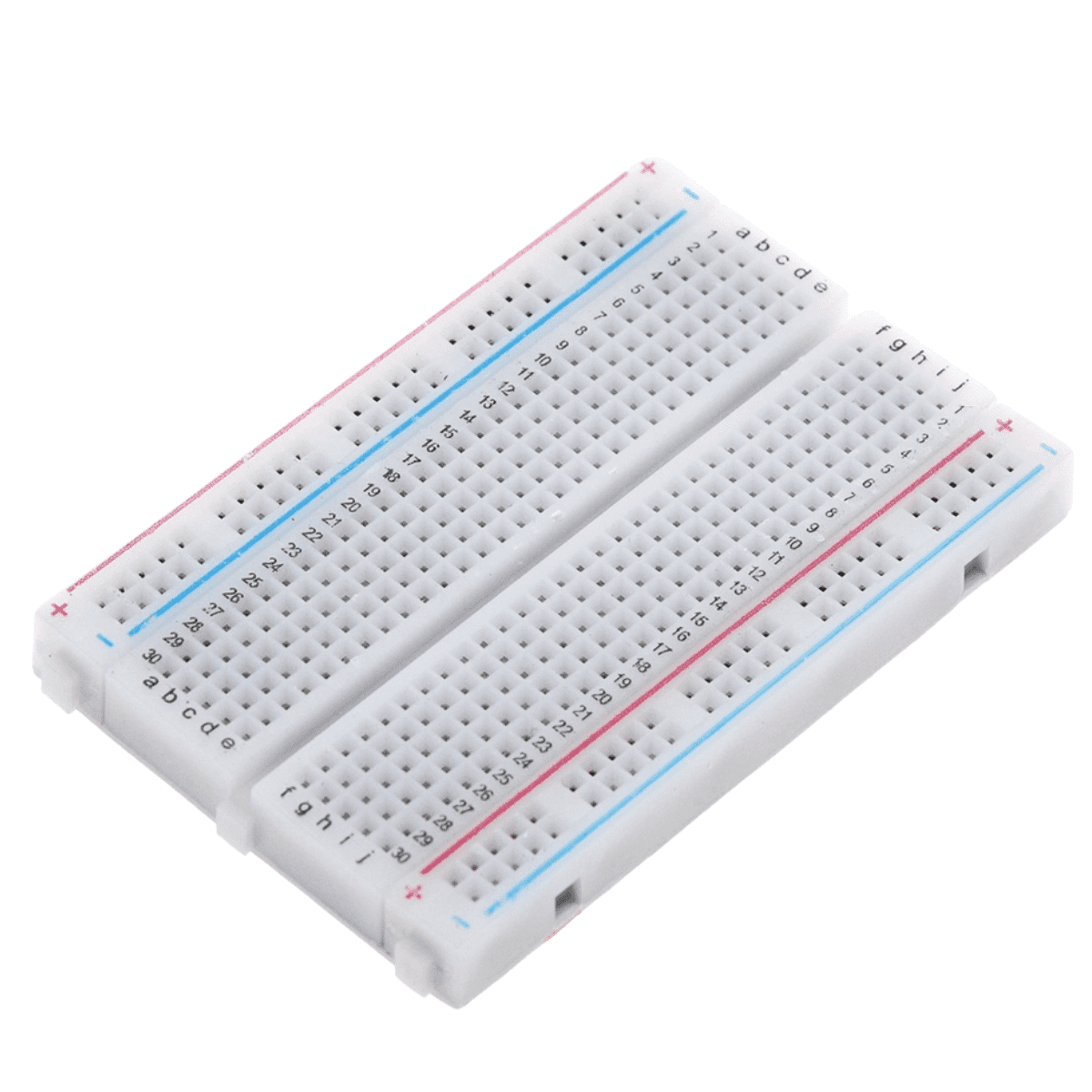
#### Diversos

##### Protoboard

A Protoboard, também conhecida como Placa de Ensaio, é um dispositivo com furos e conexões pré-definidas que tem como objetivo facilitar a montagem de circuitos eletrônicos experimentais de forma simples e rápida. Essa placa é extremamente útil, pois dispensa a necessidade de soldagem dos componentes, permitindo ao usuário realizar testes básicos de aplicação nos circuitos desejados de maneira conveniente.

Com a Protoboard, é possível “prototipar” e testar diferentes configurações de circuitos sem a preocupação de realizar conexões permanentes, o que a torna uma ferramenta valiosa para estudantes, engenheiros e entusiastas que desejam experimentar e desenvolver novos projetos eletrônicos.

FIGURA 9 - PROTOBOARD



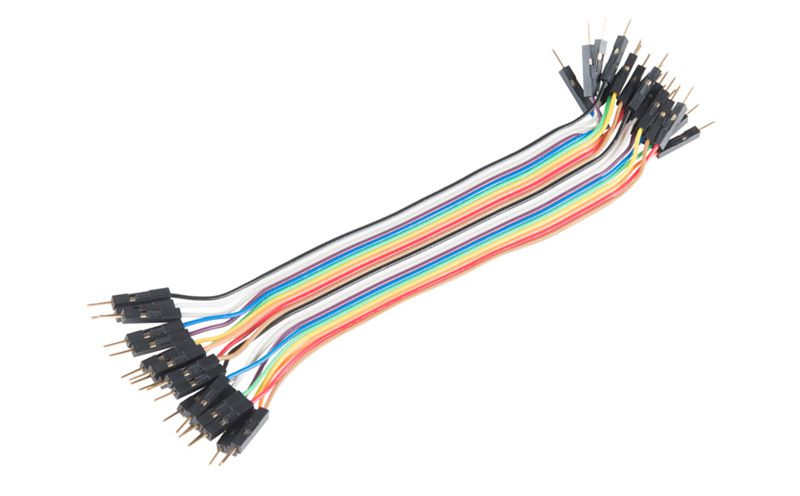
FONTE: **Protoboard 400 Furos**. Casa Da Robótica.

##### Jumpers

Os jumpers são dispositivos elétricos utilizados em placas e alguns mecanismos, como discos rígidos, para configurar, ativar ou desativar funções exclusivas do sistema que não estão disponíveis através de software. Com os jumpers, é possível controlar, por exemplo, a alimentação elétrica de um processador e, consequentemente, ajustar sua velocidade e temperatura.

Outra aplicação dos jumpers é a possibilidade de redefinir os dados da BIOS para os padrões de fábrica, permitindo restaurar as configurações originais do sistema. Essa versatilidade torna os jumpers uma solução prática e eficiente para ajustes específicos em dispositivos eletrônicos e sistemas computacionais.

FIGURA 10 - JUMPERS



FONTE: **Jumpers Macho/Macho 20 Vias**. SmartKits.

##### PCB Universal

A PCB Universal é uma placa de circuito impresso versátil, projetada para acomodar uma variedade de componentes eletrônicos de forma rápida e eficiente. Suas trilhas de cobre são organizadas em um layout padrão que segue uma matriz de furos, possibilitando a inserção e conexão de componentes através desses pontos. Essa flexibilidade na montagem simplifica a prototipagem de projetos eletrônicos, pois elimina a necessidade de criar um design específico para cada circuito. No entanto, é importante considerar que, devido à sua natureza padrão e não personalizada, ela pode não oferecer a mesma otimização de espaço e desempenho que uma PCB projetada sob medida para uma aplicação específica.

Assim, a escolha entre uma PCB Universal e uma PCB personalizada dependerá das necessidades e complexidade do projeto em questão.

### Software

Neste capítulo, concentramos nossa atenção na dimensão do software, fornecendo informações sobre as linguagens de programação utilizadas, os aplicativos e as bibliotecas incorporadas no projeto.

#### Linguagem C++

C++ é uma linguagem de programação de alto nível que combina eficiência do C com recursos de programação orientada a objetos. É amplamente utilizado em sistemas operacionais, jogos, aplicativos, dispositivos embarcados e IA. Oferece recursos avançados, mas requer cuidado com gerenciamento de memória.

Em resumo, a linguagem C++ é uma poderosa ferramenta de programação, oferecendo uma combinação única de controle de baixo nível e recursos de alto nível da programação orientada a objetos. Seu amplo uso em diversos setores da indústria de software destaca a sua importância e relevância no cenário da programação moderna.

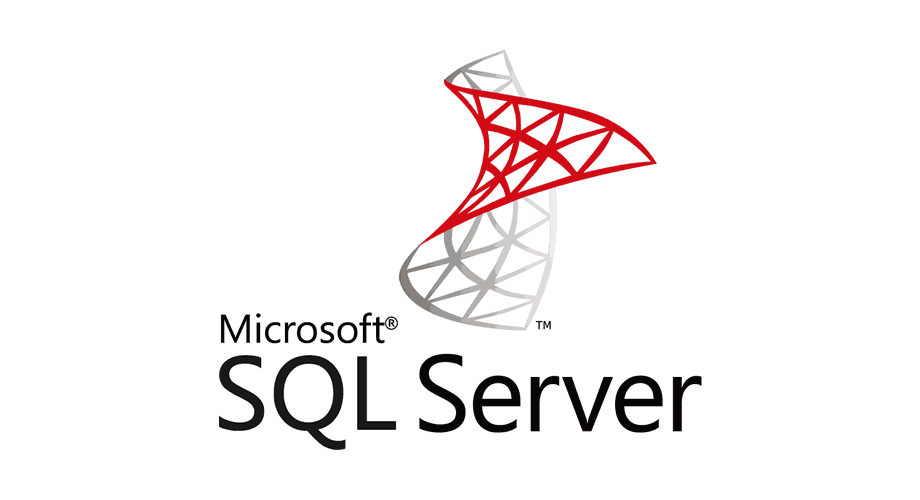
#### SQLServer

O SQL Server, desenvolvido pela Microsoft, é um sistema de gerenciamento de banco de dados relacional amplamente utilizado para armazenar, gerenciar e acessar dados de maneira eficiente e segura. Baseado no modelo relacional, organiza as informações em tabelas inter-relacionadas, seguindo a linguagem SQL como ferramenta fundamental para consultas, inserções, atualizações e exclusões de dados.

Esse sistema oferece um amplo leque de funcionalidades essenciais para a administração eficaz do banco de dados, segurança robusta para proteger os dados sensíveis, otimização do desempenho para garantir consultas rápidas e eficientes, além de mecanismos de recuperação de dados que asseguram a disponibilidade e integridade das informações em situações de falhas.

Além disso, o SQL Server facilita a integração de dados de diferentes fontes, a transformação e carga de informações (ETL) e a análise de dados para geração de relatórios e suporte à tomada de decisões. É uma solução escalável, permitindo o crescimento do sistema conforme as necessidades, e é compatível com diversas plataformas, incluindo Windows e Linux, garantindo sua flexibilidade e adaptabilidade a diversos ambientes.

FIGURA 11 - SQLServer



FONTE: **Banco de** **dados SQLServer.** DBACorp.

#### Java Script

Java Script é uma linguagem de programação amplamente utilizada para adicionar interatividade e dinamismo a páginas web. Ela é executada diretamente no navegador dos usuários e permite a criação de elementos interativos, como animações, formulários dinâmicos e atualizações de conteúdo em tempo real. Com uma sintaxe semelhante à de outras linguagens de programação, como C++ e Java, o JavaScript é uma linguagem orientada a eventos, o que significa que ele pode responder a ações dos usuários, como cliques em botões ou movimentos do mouse.

Além disso, o JavaScript é frequentemente usado em conjunto com HTML e CSS para criar experiências mais ricas e envolventes na web. Sua popularidade e versatilidade o tornam uma parte fundamental do desenvolvimento web moderno, permitindo aos desenvolvedores criar aplicações interativas e dinâmicas para os usuários online.

#### Linguagem HTML

HTML, sigla para Hypertext Markup Language, é uma linguagem de marcação usada para criar páginas web e estruturar seu conteúdo. Utilizando elementos ou tags, os desenvolvedores podem definir a estrutura, o layout e o formato dos documentos na web. O HTML é a base fundamental para a construção de sites e é frequentemente combinado com outras tecnologias, como CSS para estilização e JavaScript para interatividade.

Sua simplicidade e acessibilidade o tornam uma ferramenta essencial para criar páginas web, permitindo que qualquer pessoa com conhecimento básico em programação crie conteúdo na internet. Com a evolução das especificações do HTML, a linguagem continua a se adaptar para atender às necessidades das aplicações modernas na web.

#### Linguagem CSS

CSS, sigla para Cascading Style Sheets, é uma linguagem de estilo utilizada para definir a apresentação e a aparência de páginas web escritas em HTML ou XHTML. Por meio de regras de estilo, o CSS permite que os desenvolvedores controlem a formatação, o layout e a estilização dos elementos do conteúdo em uma página web. Isso inclui definição de cores, fontes, margens, tamanhos e posicionamento de elementos.

O CSS funciona separadamente do HTML, possibilitando a criação de um código mais limpo e organizado, além de facilitar a manutenção e a consistência visual de um site. Com a capacidade de aplicar estilos a várias páginas de uma só vez, o CSS oferece uma abordagem eficiente e flexível para a criação de interfaces atraentes e responsivas na web.

Como resultado, o CSS é uma parte indispensável do desenvolvimento web moderno, aprimorando a experiência do usuário e a estética dos sites em diferentes dispositivos e navegadores.

#### Vue

Vue.js, comumente referido como Vue, é um framework de JavaScript de código aberto utilizado para criar interfaces de usuário interativas e dinâmicas em páginas web. Ele segue o modelo de arquitetura MVVM (Model-View-ViewModel) e é projetado para ser fácil de integrar a projetos existentes e também ser adotado gradualmente em partes específicas de uma aplicação maior. Vue oferece uma sintaxe intuitiva, permitindo que os desenvolvedores definam a estrutura e o comportamento dos componentes da interface de maneira declarativa. Ele também oferece recursos poderosos, como binding de dados bidirecional, manipulação eficiente do DOM (Document Object Model) e componentes reutilizáveis.

A abordagem flexível e modular do Vue o torna uma opção atraente para o desenvolvimento ágil de aplicações front-end. É amplamente adotado por sua comunidade ativa e documentação abrangente, que tornam a aprendizagem e a implementação mais acessíveis. O Vue é empregado em diversos tipos de projetos, desde pequenos sites até aplicações web complexas e dinâmicas. Sua capacidade de simplificar a criação de interfaces responsivas e interativas o torna uma escolha popular para desenvolvedores que buscam uma estrutura eficiente e moderna para criar experiências atraentes na web.

#### C#

C# (pronuncia-se "C sharp") é uma linguagem de programação desenvolvida pela Microsoft que combina elementos da linguagem C++ e do Java. Ela foi projetada para ser moderna, orientada a objetos e segura, sendo amplamente utilizada para o desenvolvimento de aplicações Windows, aplicativos de desktop, jogos, aplicações web e serviços na plataforma .NET. C# é uma linguagem de tipagem estática, o que significa que os tipos de dados das variáveis são definidos em tempo de compilação, resultando em código mais seguro e eficiente.

Uma das características distintas do C# é a sua forte integração com o ambiente de desenvolvimento da Microsoft, o Visual Studio, que oferece ferramentas avançadas para escrita, depuração e testes de código C#. A linguagem também suporta programação assíncrona e paralela, tornando-a adequada para aplicações que requerem alto desempenho e responsividade.

Além disso, C# é uma peça central da plataforma .NET, que oferece um ecossistema de bibliotecas e frameworks para o desenvolvimento de aplicações em várias áreas. Com sua abordagem orientada a objetos e ampla adoção em empresas e na indústria de desenvolvimento de software, o C# continua sendo uma escolha popular para criar uma variedade de aplicações, desde aplicações corporativas até jogos de alta qualidade.

#### TailWind

Tailwind CSS é um framework de design de código aberto que facilita a criação de interfaces de usuário estilizadas e personalizadas em aplicações web. Diferentemente de outros frameworks, como Bootstrap, o Tailwind não fornece estilos pré-definidos, mas sim oferece classes utilitárias que permitem aos desenvolvedores criar estilos personalizados de forma mais eficiente. Ele adota uma abordagem de "design atômico", onde cada classe utilitária se refere a um estilo específico, como margens, cores, tamanhos e tipografia.

O Tailwind CSS é altamente configurável, permitindo aos desenvolvedores personalizarem os estilos de acordo com as necessidades do projeto. Com sua abordagem baseada em classes, os desenvolvedores podem rapidamente compor e modificar estilos sem a necessidade de escrever CSS personalizado. Além disso, o Tailwind inclui recursos como responsividade, onde as classes podem ser usadas para definir estilos diferentes para diferentes tamanhos de tela.

Escolhido por sua eficiência e flexibilidade, o Tailwind ganhou popularidade na comunidade de desenvolvedores front-end. Ele agiliza o processo de design e desenvolvimento, permitindo a criação de interfaces de usuário elegantes e responsivas de maneira mais rápida.

#### Entity Framework

Entity Framework (EF) é um framework de mapeamento objeto-relacional (ORM) desenvolvido pela Microsoft. Ele oferece uma abordagem de desenvolvimento orientada a objetos para interagir com bancos de dados relacionais. O objetivo do Entity Framework é simplificar a maneira como os desenvolvedores acessam, manipulam e gerenciam dados em aplicações .NET, permitindo que trabalhem com objetos no código enquanto o framework lida com a tradução desses objetos para consultas SQL e vice-versa.

Uma das principais características do Entity Framework é o mapeamento de objetos para tabelas do banco de dados, o que permite aos desenvolvedores tratar as operações de banco de dados como operações em objetos. O EF oferece uma variedade de recursos, como o LINQ (Language Integrated Query), que permite realizar consultas complexas em objetos, além de gerenciamento de relacionamentos, controle de transações e rastreamento de mudanças.

O Entity Framework suporta vários provedores de banco de dados, como SQL Server, MySQL, Oracle e PostgreSQL, tornando-o flexível para diversos ambientes. Sua utilização simplifica o desenvolvimento, melhora a manutenção e promove uma abstração eficiente entre a lógica de negócios e o armazenamento de dados. Com essas vantagens, o Entity Framework é amplamente adotado em projetos .NET para agilizar o acesso e a manipulação de dados em bancos de dados relacionais.

#### Arduino IDE

A Arduino Integrated Development Environment (IDE) é uma ferramenta fundamental para entusiastas, estudantes e profissionais que desejam mergulhar no mundo dos projetos eletrônicos com as placas Arduino. Ela se destaca por sua abordagem acessível e amigável, tornando a eletrônica e a programação mais acessíveis a uma ampla gama de pessoas, independentemente de sua experiência prévia. A IDE fornece uma linguagem de programação própria, baseada em C/C++, que é notavelmente fácil de aprender, mesmo para iniciantes. Uma das características que torna a Arduino IDE tão popular é sua vasta biblioteca de funções predefinidas, chamadas de "sketches", que tornam a realização de tarefas comuns, como controlar LEDs, ler sensores ou comunicar-se com outros dispositivos, uma tarefa simplificada.

Outro aspecto notável é a capacidade da Arduino IDE de suportar uma ampla variedade de placas Arduino, desde as mais simples, como o Arduino Uno, até placas mais avançadas, como o Arduino Mega ou o Arduino Due. Isso oferece aos desenvolvedores a flexibilidade de escolher a placa que melhor atende às suas necessidades e, ao mesmo tempo, usar um ambiente de programação consistente e familiar.

A Arduino IDE é compatível com sistemas operacionais como Windows, macOS e Linux, garantindo sua acessibilidade em diferentes plataformas. Além disso, ela oferece recursos avançados, como depuração de código, monitor serial e a capacidade de integrar bibliotecas personalizadas, o que a torna uma escolha poderosa para desenvolvedores experientes.

Em resumo, a Arduino IDE desempenha um papel crucial na promoção da eletrônica DIY (faça você mesmo) e no incentivo à programação, possibilitando que pessoas de todas as idades e níveis de experiência explorem projetos criativos e inovadores. Com sua abordagem simplificada, ela remove barreiras para o aprendizado e a experimentação, tornando a eletrônica e a programação mais acessíveis e empolgantes.

#### Bibliotecas

* Wire.h

A biblioteca "Wire.h" é uma parte essencial da plataforma Arduino, simplificando a comunicação entre dispositivos por meio do protocolo I2C. Ela permite conectar diversos componentes eletrônicos, como sensores e displays, a uma placa Arduino e facilita a troca eficiente de informações. Essa biblioteca é amplamente utilizada em projetos de eletrônica e robótica, tornando a programação mais acessível e eficaz.

* Adafruit\_Sensor.h

A biblioteca "Adafruit\_Sensor.h" simplifica o uso de uma variedade de sensores em placas Arduino, fornecendo uma estrutura comum para acessar dados de diferentes tipos de sensores. Isso facilita a coleta de informações e o desenvolvimento de projetos que envolvem medições e interações com o ambiente.

* Adafruit\_TSL2561\_U.h

A biblioteca "Adafruit\_TSL2561\_U.h" é usada para interagir com o sensor de luminosidade TSL2561 da Adafruit. Ela simplifica a programação e a coleta de dados de luminosidade, tornando mais fácil a incorporação de medições de luz em projetos Arduino. É uma escolha conveniente para aplicações que envolvem controle de iluminação, detecção de luz ambiente e outras funções relacionadas à luminosidade.

* Adafruit\_BME280.h

A biblioteca "Adafruit\_BME280.h" é usada para trabalhar com o sensor de temperatura, umidade e pressão atmosférica BME280 da Adafruit. Ela simplifica a programação e a coleta de dados dessas variáveis climáticas, tornando mais fácil a inclusão de medições precisas em projetos Arduino. Essa biblioteca é uma escolha comum em aplicações que requerem monitoramento do ambiente, como estações meteorológicas, sistemas de controle de clima e projetos de IoT relacionados ao clima.

* SPI.h

A biblioteca "SPI.h" é uma parte fundamental da plataforma Arduino e é usada para a comunicação SPI (Serial Peripheral Interface). Ela simplifica a conexão e a comunicação entre placas Arduino e uma variedade de dispositivos, como sensores, displays, módulos de memória e outros componentes eletrônicos. O protocolo SPI é conhecido por sua alta velocidade de comunicação e é amplamente utilizado em projetos que exigem transferência rápida de dados. Com a biblioteca "SPI.h", os desenvolvedores podem aproveitar a eficiência da comunicação SPI de forma mais conveniente, economizando tempo de programação e recursos. Essa biblioteca é particularmente útil em aplicações que envolvem leitura e gravação de dados em dispositivos de armazenamento, como cartões SD, ou na interação com displays gráficos de alta velocidade. Resumidamente, a "SPI.h" é uma ferramenta importante que simplifica a implementação do protocolo SPI em projetos Arduino, tornando a comunicação eficiente e rápida com uma variedade de dispositivos eletrônicos.

* Ethernet.h

A biblioteca "Ethernet.h" é essencial para projetos que envolvem conectividade Ethernet com placas Arduino. Ela fornece as funções e ferramentas necessárias para a criação de servidores web, comunicação pela rede e acesso a recursos online. Essa biblioteca é amplamente utilizada em aplicações que exigem monitoramento remoto, controle de dispositivos via internet e troca de informações com servidores. Ela torna mais acessível a implementação de soluções de IoT (Internet das Coisas) e automação residencial que dependem da conectividade Ethernet, proporcionando uma maneira eficaz de criar projetos que se comunicam com o mundo exterior por meio da rede. Resumindo, a "Ethernet.h" é uma biblioteca fundamental para adicionar capacidades de rede a projetos Arduino, tornando-os capazes de se comunicar pela internet e acessar recursos remotos de forma eficiente e conveniente.

* EthernetUdp.h

A biblioteca "EthernetUdp.h" é uma extensão da biblioteca "Ethernet.h" que habilita a comunicação usando o protocolo UDP em projetos Arduino com conectividade Ethernet. Isso é útil para aplicações que exigem comunicação em tempo real e baixa latência pela rede.

# DESENVOLVIMENTO

## TESTE DOS SENSORES E PLACA ETHERNET

Neste capítulo, é abordado como foram realizados de forma detalhada os testes da Placa Ethernet e dos Sensores utilizados no projeto.

### Teste Placa Ethernet

O teste da placa ethernet W5100 foi realizado da seguinte maneira, conectamos a placa ethernet diretamente ao notebook/PC, programamos o código para o Arduino com as devidas exigências da placa, utilizando as bibliotecas SPI.h e Ethernet.h, fizemos uma conexão de ponte entre a rede de Wifi e a rede do Cabo Ethernet, definido um IP estático. Após isso, utilizamos um servidor socket em Python, para observar se a conexão estava funcionando e se os dados enviados estavam corretos. Com resultado obtivemos a confirmação de que a placa Ethernet estava funcionando devidamente e enviando os dados corretamente.

### Teste Sensor ML8511

O teste do sensor ML8511 foi realizado da seguinte maneira, usando jumpers o sensor foi conectado a protoboard e consequentemente ao Arduino com suas devidas exigências. Após isso o desenvolvimento de um código com as devidas transformações e cálculos para conversão dos sinais mandados para o sensor, o valor é impresso na serial. Com os resultados deste teste confirmamos que o sensor estava se comunicando corretamente com o Arduino e uma leitura correta e precisa da luz UV.

### Teste Sensor BME280

O teste do sensor BME280 foi realizado da seguinte maneira, usando jumpers o sensor foi conectado a protoboard e consequentemente ao Arduino com suas devidas exigências, utilizando uma comunicação I2C. Após programar o código para o Arduino, utilizando de sua biblioteca Adafruit\_BME280.h, obtivemos os valores do sensor e os imprimimos na serial. Assim, foi possível determinar que o sensor estava funcionando corretamente, lendo com precisão os valores de temperatura, umidade e pressão.

### Teste Sensor TSL2561

O teste do sensor TSL2561 foi realizado da seguinte maneira, usando jumpers o sensor foi conectado a protoboard e consequentemente ao Arduino, com suas devidas exigências, utilizando uma comunicação I2C. Após programar o código para o Arduino, utilizando a biblioteca Adafruit\_TSL2561\_U.h, obtivemos os valores do sensor e os imprimimos na serial. Assim, foi possível afirmar que o sensor estava funcionando corretamente, lendo com precisão os valores de luz visível e luz infravermelha do ambiente.

## BACK-END

Neste capítulo, direcionamos nosso foco para a área do back-end, onde exploramos as linguagens de programação, frameworks e bibliotecas utilizados na construção da parte interna do sistema.

### Instalação do Banco de dados no Azure

A instalação de um banco de dados no Azure envolve criar um grupo de recursos para organização, configurar um servidor de banco de dados escolhendo o tipo desejado (por exemplo, SQL Database), definir regras de firewall, criar o banco de dados propriamente dito, gerenciar usuários e permissões, conectar-se à instância, importar ou criar dados, monitorar o desempenho, configurar backups e recuperação. Esses passos garantem um ambiente de armazenamento e gerenciamento de dados na nuvem do Azure. Consulte a documentação para detalhes específicos.

### Configuração do SQL Server

A configuração do SQL Server envolve instalar o software, definir configurações de instância, criar bancos de dados, configurar segurança e autenticação, ajustar recursos e memória, definir planos de backup e manutenção, implementar monitoramento e alertas, adotar criptografia e segurança avançada, documentar configurações e fazer backups. Essas etapas garantem um ambiente de banco de dados otimizado, seguro e bem gerenciado. Consulte a documentação da Microsoft para orientações detalhadas.

### Criação de Tabelas no SQL Server

A criação de tabelas no SQL Server envolve escrever um comando "CREATE TABLE" com o nome da tabela, colunas e seus tipos de dados. Adicione restrições como chaves primárias, chaves estrangeiras e índices, se necessário. Execute o comando SQL para criar a tabela e verifique a criação. Se desejado, insira dados usando o comando "INSERT INTO" e valide os resultados com consultas SELECT. Certifique-se de seguir a sintaxe correta e consultar a documentação do SQL Server para detalhes específicos.

### Implementação do CRUD

A implementação do CRUD abrange a criação, leitura, atualização e exclusão de dados. Comece modelando os dados, implemente operações para criar, ler, atualizar e excluir registros. Integre essas operações com a interface do usuário, assegurando validações e segurança. Teste e depure cada operação, documentando os processos para facilitar a manutenção. Siga as melhores práticas para garantir eficácia, confiabilidade e segurança na manipulação de dados.

### Estabelecimento do padrão MVC

O estabelecimento do padrão MVC envolve dividir um aplicativo em três componentes principais: Model (lógica de negócios e dados), View (interface do usuário) e Controller (gerenciamento da interação). Crie e desenvolva cada componente com responsabilidades claras. A Controller coordena a interação entre Model e View, mantendo a separação e evitando acoplamento excessivo. Teste cada componente e configure a comunicação adequada entre eles. Mantenha a modularidade para facilitar manutenção e atualizações futuras.

### Programação do sistema de login

A programação de um sistema de login envolve criar uma página de login, capturar e validar as credenciais inseridas pelos usuários, criptografar as senhas armazenadas no banco de dados, opcionalmente gerar tokens de sessão, implementar bloqueios após tentativas malsucedidas, e possivelmente adicionar recuperação de senha. Gerenciar sessões, testar exaustivamente, monitorar a segurança e manter o sistema atualizado são etapas essenciais para garantir a proteção das informações e a funcionalidade segura do sistema de login.

## FRONT-END

Neste capítulo, concentramos nossa atenção na área do front-end, onde apresentamos as linguagens de programação, ferramentas e bibliotecas empregadas no desenvolvimento da interface e interação com o usuário.

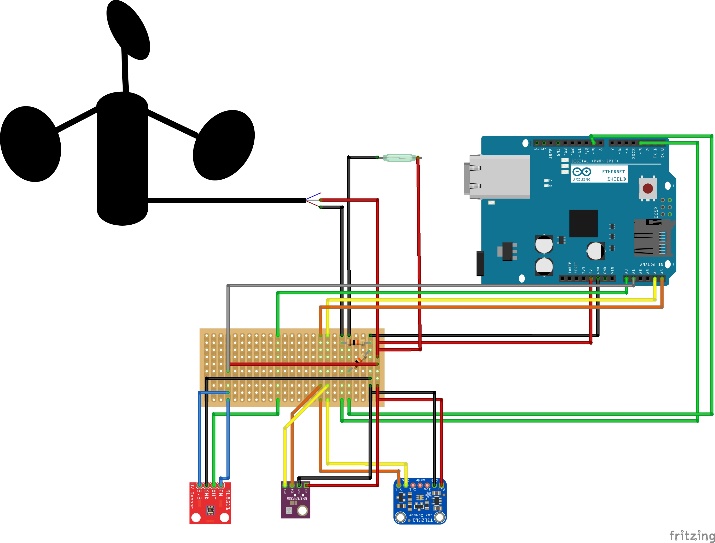
### Programação Home Page e Contatos

A programação da página inicial (Home Page) envolve a criação de uma estrutura HTML e estilos CSS atrativos, com elementos interativos em JavaScript, para apresentar informações relevantes e incentivar a exploração do site. A programação da página de contatos inclui a criação de um formulário HTML com campos para nome, e-mail, assunto e mensagem, validações, envio de e-mails por meio de linguagens de programação como PHP ou Python, feedback ao usuário após o envio e informações de contato adicionais. Teste exaustivamente e assegure a segurança dos dados. Personalize as interfaces para atender às necessidades da empresa e garantir a usabilidade, a aparência visual e a privacidade dos usuários.

## MODELO DO FRITZING

O modelo fritzing desempenha um papel fundamental no processo de criação de projetos eletrônicos, permitindo que os usuários esbocem, projetem e visualizem circuitos de maneira eficiente e intuitiva, com isso foi realizado o modelo do projeto Nimbus para facilitar a concepção do projeto.

Figura 12 - Modelo do Fritzing



FONTE: BASSOI, Gabriel. **Modelo do Fritzing.** Curitiba, 2023.

## CONSTRUÇÃO

Neste capítulo, voltamos nossa atenção para a etapa de construção do projeto, onde detalhamos os processos de execução, as técnicas de montagem e a integração de componentes que deram vida à iniciativa.

### Impressão 3D do Pluviômetro

Nosso pluviômetro impresso em 3D é um dispositivo para medir a quantidade de chuva em uma área específica. Feito a partir de tecnologia de impressão tridimensional, esse instrumento oferece uma abordagem prática e acessível para monitorar as condições climáticas.

Composto por componentes principais impressos em 3D:

Funil Coletor: Um funil otimizado para direcionar e coletar a água da chuva.

Base Giratória: Uma base giratória onde o peso da água é calculado e movido gerando um pêndulo, utilizamos um Sensor Reed resistente a água e um imã para detecção de movimento.

Base Durável: A base impressa em 3D proporciona a fixação do pluviômetro, que foi colocado sobre um frame de alumínio e fixado por uma barra roscada de 5mm.

### Design e Modelagem

Um design funcional foi criado utilizando software de modelagem 3D, garantindo a geometria adequada para o funil, a câmara de medição e a base, além de ser desenvolvido para ser simples e compacto.

O cano de PVC modificado apresenta as seguintes características notáveis:

Compartimentos Estratégicos: O cano foi dividido em compartimentos personalizados, cada um projetado para acomodar com precisão o Arduino e um sensor específico. Essa divisão garante que os componentes se encaixem de forma segura e ordenada.

Fixação por Barras Roscadas: Barras roscadas foram estrategicamente instaladas no interior do cano para servirem como suportes robustos para o Arduino e os sensores. Essas barras mantêm os componentes no lugar e minimizam qualquer movimento indesejado.

Modificações para Cabos: Aberturas estrategicamente posicionadas no cano permitem a passagem organizada dos cabos dos sensores e do Arduino. Isso assegura que as conexões elétricas sejam mantidas intactas e não interfiram com a funcionalidade dos dispositivos.

Vedação Reforçada: Juntas de vedação à prova d'água foram aplicadas nas extremidades do cano e dos compartimentos, criando uma barreira resistente à umidade e protegendo os componentes internos.

### Montagem

A Seleção de Materiais foi pensado em plásticos resistentes ao clima (PLA), foram escolhidos, assegurando a durabilidade do pluviômetro.

O dispositivo passou por testes básicos em condições de chuva simulada, avaliando a coleta de água e a clareza das leituras. Com base nos testes, pequenos ajustes foram feitos para otimizar a precisão e o desempenho.

O cano de PVC modificado, projetado para abrigar um Arduino e quatro sensores - BME280, ML8511, TSL2561 e um sensor reed impermeável - é um desenvolvimento para integração tecnológica e análise climática. Este dispositivo multifuncional foi concebido para coletar dados meteorológicos e ambientais de maneira eficaz, enquanto mantém a funcionalidade dos sensores.

Figura 13 - PROJETO FISICO

Uma imagem contendo no interior, objeto, pequeno, mesa

Descrição gerada automaticamente

FONTE: BASSOI, Gabriel. **Projeto física**. Curitiba, 2023.

Figura 14 - PROJETO FISICO

Uma imagem contendo no interior, pequeno, pia, mesa

Descrição gerada automaticamente

FONTE: BASSOI, Gabriel. **Projeto física**. Curitiba, 2023.

#### Desenvolvimento do cano PVC Modificado

O processo de desenvolvimento do cano de PVC modificado ocorreu em várias etapas:

Planejamento Detalhado: Um plano foi elaborado considerando as dimensões do cano, a posição dos sensores, a disposição interna e os requisitos de fixação.

Aquisição de Materiais: Canos de PVC, barras roscadas, juntas de vedação à prova d'água e outras peças necessárias foram reunidos para a montagem.

Modificações Estruturais: O cano de PVC foi cortado e perfurado para acomodar as barras roscadas e as aberturas para cabos.

Fixação e Montagem: As barras roscadas foram instaladas e fixadas nas posições apropriadas dentro do cano. Os sensores e o Arduino foram afixados nas barras utilizando suportes impressos em 3D.

Conexões Elétricas: Os cabos dos sensores foram passados pelas aberturas do cano e conectados ao Arduino, garantindo a funcionalidade adequada.

Testes de Vedação: Foi realizada uma verificação minuciosa da vedação à prova d'água para garantir que o interior do cano permanecesse seco e protegido.

Testes Funcionais: O Arduino e os sensores foram testados para garantir que todas as funções continuassem a operar sem problemas, mesmo após as modificações.

Otimização Final: Pequenos ajustes foram feitos para otimizar a organização interna, garantindo que os componentes permanecessem fixos e os cabos organizados.

Implantação e Uso: Após os testes e ajustes, o cano modificado foi implantado para coletar dados meteorológicos e ambientais em seu ambiente pretendido.

## Integração

A etapa final de integração do sistema envolveu a fusão dos códigos dos sensores BME280, ML8511, TSL2561 e da placa Ethernet. Isso foi realizado no ambiente de desenvolvimento do Arduino IDE, onde funções foram criadas para coletar dados dos sensores e consolidá-los. Em seguida, uma solicitação GET foi enviada para o servidor, transmitindo os dados obtidos. O servidor, escrito em C#, assumiu a responsabilidade de inserir e consultar informações no banco de dados SQL Server, permitindo o armazenamento e exibição dos dados na página web.

Figura 15 – Diagrama de Dados do Sistema

Diagrama

Descrição gerada automaticamente

FONTE: KAI, Guilherme. **DIAGRAMA DE DADOS DO SISTEMA.** Curitiba, 2023.

# RESULTADOS

O Sistema meteorológicos desempenham um papel insubstituível na obtenção de dados climáticos precisos e confiáveis, beneficiando uma ampla gama de setores, desde a agricultura até a aviação e a segurança civil. Essas informações não apenas aprimoram a tomada de decisões e aumentam a eficiência, mas também desempenham um papel fundamental na prevenção de riscos e garantia da segurança em várias atividades.

Com diversos testes realizados, os sensores ML8511, BME280 e o sensor TSL2561, demostraram extrema eficácia para conseguir detectar os seguintes dados: luz UV, temperatura, umidade, pressão, luz visível e luz infravermelho, por conseguinte, exibido uma média e os dados em tempo real no site.

A disponibilidade dessas informações confiáveis é essencial para enfrentar os desafios em constante evolução associados às condições climáticas, ajudando a moldar um futuro mais seguro e sustentável para nossa sociedade.

# CONSIDERAÇÕES FINAIS

Ao longo da execução do projeto, o grupo se deparou com inúmeros obstáculos, por mais intrincados que pudessem ser, todos foram vencidos em busca da conclusão do projeto.

O primeiro desafio, foi a escolha de um projeto decente, em que foi realizado uma brainstorm de ideias de projetos, dessa forma, optando pelo projeto de estação meteorológico "Nimbus", que se demonstrou ser um projeto aplicável e de forma acessível.

Por conseguinte, enfrentamos o problema de realizar um planejamento mais detalhado de forma mais desenvolvida e que seja realista e acessível para um projeto com um prazo menor de 4 meses.

Ademais, enfrentamos desafios ao considerar a integração de aspectos relacionados a custos na disciplina de empreendedorismo, devido à natureza predominantemente social do projeto.

Além disso, enfrentamos desafios consideráveis na configuração da placa Ethernet, uma vez que a atenção dada a essa etapa não foi tão ampla quanto o necessário. A complexidade da configuração e a falta de recursos dedicados tornaram essa fase especialmente exigente. Além disso, tivemos dificuldades no estabelecimento da conexão da Placa Ethernet com o Banco de Dados, devido à necessidade de garantir a compatibilidade entre o hardware e o software, bem como a resolução de eventuais conflitos de protocolo, o que exigiu esforço adicional e expertise técnica para superar esses desafios.

No aspecto de design, enfrentamos um desafio inicial ao nos familiarizar com a ferramenta "Figma". A curva de aprendizado para dominar essa ferramenta de design específica foi um pouco íngreme, uma vez que exigia a aquisição de novas habilidades e o entendimento de seu funcionamento.

Vale ressaltar que ter desafios e problemas em projetos é fundamental, pois estimula o pensamento crítico, a criatividade e a resolução de problemas. Esses obstáculos não apenas promovem o crescimento pessoal e profissional, mas também aprimoram a qualidade do resultado final. Enfrentar dificuldades leva a soluções mais inovadoras e eficazes, e, muitas vezes, os problemas revelam oportunidades de melhoria que podem ser aplicadas em projetos futuros. Portanto, encarar desafios de maneira positiva e como parte inerente do processo de desenvolvimento é essencial para o sucesso e o progresso contínuo em qualquer empreendimento.

# REFERÊNCIAS BIBLIOGRÁFICAS

**Arduino uno R3 + Cabo USB 2.0 - A-B**. Baú da Eletrônica, 2021. Disponível em: https://www.baudaeletronica.com.br/arduino-uno-r3.html. Acesso em 9 de ago. 2023 às 18:16h.

BABOS, Flávio. **30 Melhores Projetos Com Arduino Para Criar em 2023.** FlavioBabos, 2023. Disponível em: <https://flaviobabos.com.br/melhores-projetos-com-arduino/>. Acesso em 28 de fev. 2023 às 22:24h.

BABOS, Flávio. **Estação Meteorológica com Arduino: Tutorial Completo.** FlavioBabos, 2022. Disponível em: <https://flaviobabos.com.br/estacao-meteorologica-arduino/>. Acesso em 16 de fev. 2023 às 20:48h.

BENNE. **A Complete Guide On Ethernet Network Shield 5100 For Arduino UNO.** Markerguides, c2022. Disponível em: <https://www.makerguides.com/ethernet-network-shield-5100-arduino-uno/>. Acesso em 12 de abr. 2023 às 20:15h.

**Barômetro BME280 Sensor de Pressão Umidade Temperatura GY-BME.** EasyTronics, 2022. Disponível em: <https://www.easytronics.com.br/barometro-bme280>. Acesso em 14 de ago. 2023 às 15:02h.

**Ethernet Shield W5100**. FilipeFlop, c2021. Disponível em: https://www.filipeflop.com/produto/ethernet-shield-w5100-para-arduino/. Acesso em 20 de jul. 2021 às 15:32h.

GALVÃO, Michel. **Previsão do Tempo com Arduino e Display LCD.** Eletrogate, 2021. Disponível em: <https://blog.eletrogate.com/previsao-do-tempo-com-arduino-mega-e-display-lcd/>. Acesso em 28 de fev. 2023 às 22:29h.

**I**MARZEN**. ESP32 E-Paper Weather Display.** HACKADAY.IO, c2023. Disponível em: <https://hackaday.io/project/189708-esp32-e-paper-weather-display>. Acesso em 24 de mar. 2023 às 19:06h.

IVAN. **TshWatch, watch based on esp32.** HACKADAY.IO, 2022. Disponível em: <https://hackaday.io/project/184604-tshwatch-watch-based-on-esp32>. Acesso em 24 de mar. 2023 às 19:09h.

**Jumpers Macho/Macho 20 Vias**. SmartKits. Disponível em: https://www.smartkits.com.br/prototipagem/jumpers/jumpers-machomacho-20-unidades. Acesso em 9 de ago. 2023 às 14:06h.

KASHIF**. How to Use Multiple I2C devices with Arduino.** Linuxhint, 2022. Disponível em: <https://linuxhint.com/use-multiple-i2c-devices-arduino/#:~:text=Using%20I2C%20we%20can%20connect,SDA%20and%20SCL%20line%20high>. Acesso em 24 de mar. 2023 às 22:22h.

NEWTON, Alex. **How to Measure Wind Speed using Anemometer & Arduino.** How2eletronics, 2023. Disponível em: <https://how2electronics.com/measure-wind-speed-using-anemometer-arduino/>. Acesso em 4 de abr. 2023 às14:07h.

OOSTERWIJCK, Patrick. **WiFi Logger.** HACKADAY.IO, 2016. Disponível em: <https://hackaday.io/project/13364-wifi-logger>. Acesso em 24 de mar. 2023 às 19:13h.

**Banco de Dados SQL Server.** DBACorp, 2023. Disponível em: https://www.dbacorp.com.br/bancos-de-dados/licenciamento/microsoft-sql-server-logo/. Acesso em 10 de out. 2023 às 13:41h.

**Pluviômetro Automático PI3**. MercadoLivre. Disponível em: <https://produto.mercadolivre.com.br/MLB-3862622974-pluvimetro-automatico-pl3-cabrigo-externo-parduino-etc-_JM>. Acesso em 12 de set. 2023 às 14:57h.

**Pluviômetro: Conheça o melhor modelo.** SWEETESTHOME, 2021. Disponível em: https://www.sweetesthome.com.br/pluviometro-melhores-modelos/. Acesso em 26 de jul. 2023 às 11:57h.

**Protoboard 400 Furos**. Casa Da Robótica, 2019. Disponível em: https://www.casadarobotica.com/prototipagem-e-ferramentas/prototipagem/protoboard/protoboard-400-furos-pontos. Acesso em 9 de ago. 2023 às 14:05h.

**Sensor De Luz Luxímetro Tsl2561 Arduino.** Mercado Livre, c2022. Disponível em:

[https://produto.mercadolivre.com.br/MLB-819465978-sensor-de-luz-luximetro-tsl2561-arduino-0201-\_JM](https://pucpredu-my.sharepoint.com/personal/gabriel_bassoi_tecpuc_com_br/Documents/Projeto%20final/Design)**.** Acesso em 14 de ago. 2023 às 15:01h.

**Sensor Ultravioleta GYML8511**. EasyTronics, 2019. Disponível em: <https://www.easytronics.com.br/gyml8511-sensor-ultra-violeta>. Acesso em 24 de jul. 2023 às 12:12h.

VASSILIADES, Charles. **Dispositivo que faz a previsão do tempo.** Laboratório de Garagem, 2014. Disponível em: <https://labdegaragem.com/profiles/blogs/dispositivo-que-faz-a-previs-o-do-tempo>. Acesso em 28 de fev. 2023 às 22:28h.

**O que significa placa de circuito impresso (PCB)?.** The Astrology Page, 2023. Disponível em: <https://pt.theastrologypage.com/printed-circuit-board>. Acesso em 17 de out. 2023 às 10:38h.

**WH-SP-WS01 anemômetro**. Aliexpress. Disponível em: <https://pt.aliexpress.com/item/1005002799949064.html>. Acesso em 9 de ago. 2023 às 14:04h.

**Wind Speed Meter Using Anemometer & Arduino.** IotProjectsIdeas, 2022. Disponível em: <https://iotprojectsideas.com/wind-speed-meter-using-anemometer-arduino/>. Acesso em 4 de abr. 2023 às 14:08h.

**Resistor 10K 1/4W**. Marino Store. Disponível em: <https://www.marinostore.com/componentes/resistores/resistor-10k-14w>. Acesso em 14 de ago.

2023 às 13:47h.

KINSTA. **O Que é MySQL? Uma Explicação Simples para Quem Está Começando.** Kinsta, 2022. Disponível em: <https://kinsta.com/pt/base-de-conhecimento/o-que-e-mysql/#:~:text=que%20%C3%A9%20MySQL%3F-,MySQL%20%C3%A9%20um%20sistema%20open%2Dsource%20de%20gerenciamento%20de%20base,da%C3%AD%20o%20seu%20nome%20relacional>. Acesso em 18 de ago. às 12:20h.

KRIGER, Daniel. **QUAL A LINGUAGEM DE PROGRAMAÇÃO UTILIZADA EM ARDUINO.** Kenzie, 2022. Disponível em: <https://kenzie.com.br/blog/programacao-para-arduino/> Acesso em 18 de ago. às 13:40h.

**O que é e para que serve C++ em programação?.** Suas Aulas Particulares. Disponível em: <https://www.suasaulasparticulares.com.br/blog/que-e-que-serve-programacao#:~:text=C%2B%2B%20%C3%A9%20uma%20linguagem%20de,pr%C3%B3pria%20e%20seu%20pr%C3%B3prio%20nome>. Acesso em 18 de ago. às 13:40h.

ROBOCORE. **Como utilizar uma Protoboard**. 2017. Disponível em: <https://www.robocore.net/tutoriais/como-utilizar-uma-protoboard>. Acesso em 18 de ago. às 13:40h.

BARRETO, André. **Arduino IDE: o que é e como instalar?**. Victor Vision, 2023. Disponível em: <https://victorvision.com.br/blog/arduino-ide/#:~:text=O%20que%20%C3%A9%20Arduino%20IDE,significa%20Ambiente%20de%20Desenvolvimento%20Integrado>. Acesso em 18 de ago. às 13:40h.

ESTRELLA, Carlos. **O que é JavaScript e Para Que Serve na Programação Web**. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-javascript>. Acesso em 18 de ago. às 15:10h.

GUEDES, Marylene. **O que é e como começar com C# (C Sharp)?**. 2017. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-e-como-comecar-com-c-sharp>. Acesso em 18 de ago. às 15:13h.

# APÊNDICE 01 – CÓDIGO FONTE DO ETHERNET SHIELD W5100

#include <SPI.h>

#include <Ethernet.h>

#include <EthernetUdp.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

byte gateway[] = {192, 168, 0, 2};

byte subnet[] = {255, 255, 255, 0};

IPAddress ip(192, 168, 0, 20);

int port = 80;

char server[] = "192.168.0.16";

char path[] = "/caminho\_do\_post";

float temperatura = 25.5;

float pressao = 1013.25;

float umidade = 50.0;

float luzUV = 0.75;

float luzNormal = 100.0;

float luzInfravermelha = 10.5;

float velocidadeVento = 5.6;

float nivelPrecipitacao = 2.3;

void setup() {

Ethernet.begin(mac, ip, gateway, sunet);

delay(1000);

Serial.begin(9600);

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

Serial.println("Arduino UNO");

Serial.println("Ethernet Shield W5100");

Serial.println("Conectando...");

if (Ethernet.linkStatus() == LinkON) {

Serial.println("Conectado com sucesso!");

} else {

Serial.println("Falha na conexão Ethernet!");

while ();

}

Serial.print("MAC address: ");

Serial.println(Ethernet.MACAddress());

Serial.print("Conectado a rede no IP: ");

Serial.println(Ethernet.localIP());

Serial.print("Máscara de sub-rede: ");

Serial.println(Ethernet.subnetMask());

Serial.print("Gateway: ");

Serial.println(Ethernet.gatewayIP());

Serial.print("DNS: ");

Serial.println(Ethernet.dnsServerIP());

Serial.print("Hardware status: ");

Serial.println(Ethernet.hardwareStatus());

}

void loop() {

EthernetClient client;

if (client.connect(server, port)) {

Serial.println("Conectado ao servidor");

String postData = "temperatura=" + String(temperatura) +

"&pressao=" + String(pressao) +

"&umidade=" + String(umidade) +

"&luzUV=" + String(luzUV) +

"&luzNormal=" + String(luzNormal) +

"&luzInfravermelha=" + String(luzInfravermelha) +

"&velocidadeVento=" + String(velocidadeVento) +

"&nivelPrecipitacao=" + String(nivelPrecipitacao);

client.println("POST " + String(path) + " HTTP/1.1");

client.println("Host: " + String(server));

client.println("Content-Type: application/x-www-form-urlencoded");

client.println("Content-Length: " + String(postData.length()));

client.println();

client.println(postData);

delay(500);

while (client.available()) {

char c = client.read();

Serial.write(c);

}

Serial.println("\nDados enviados com sucesso!");

} else {

Serial.println("Falha ao se conectar ao servidor");

}

}

# APÊNDICE 02 – CÓDIGO FONTE ML8511

class Ml8511 {

public:

float uvInt

void getUV() {

int uv\_Level = analogRead\_average(UV\_OUT);

int ref\_Level = analogRead\_average(REF\_3V3);

float output\_Voltage = 3.3 / ref\_Level \* uv\_Level;

float uvIntensity = mapfloat(output\_Voltage, 0.99, 2.8, 0.0, 15.0);

uvInt = uvIntensity;

delay(500);

}

private:

int UV\_OUT = A0;

int REF\_3V3 = A1;

int analogRead\_average(int pinToRead) {

int NumberOfSamples = 8;

int runningValue = 0;

for(int x = 0; x < NumberOfSamples; x++)

runningValue += analogRead(pinToRead);

runningValue /= NumberOfSamples;

return(runningValue);

}

float mapfloat(float x, float in\_min, float in\_max, float out\_min, float out\_max) {

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

};

Ml8511 ml8511;

void setup()

{

Serial.begin(9600);

Serial.println("ML8511 example");

}

void loop()

{

ml8511.getUV();

}

# APÊNDICE 03 – CÓDIGO FONTE BME280

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BME280.h>

Adafruit\_BME280 bme;

class Bme280 {

public:

float temp

float umid

float pres

void start() {

bme.begin();

}

void getData() {

temp = bme.readTemperature();

umid = bme.readHumidity();

pres = bme.readPressure() / 100.0F;

}

};

Bme280 bme280;

void setup() {

Serial.begin(9600);

Serial.println("Teste do sensor BME280");

bme280.start();

}

void loop() {

bme280.getData();

delay(1000);

}

# APÊNDICE 04 – CÓDIGO FONTE TSL2561

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_TSL2561\_U.h>

Adafruit\_TSL2561\_Unified tsl = Adafruit\_TSL2561\_Unified(TSL2561\_ADDR\_FLOAT, 12345);

class Tsl2561 {

public:

float lux;

void start() {

if (!tsl.begin()) {

Serial.println("ERROR: No TSL2561");

exit(0);

}

configureSensor();

}

void getLux() {

sensors\_event\_t event;

tsl.getEvent(&event);

if (event.light) {

Serial.println(event.light);

lux = event.light;

} else {

Serial.println("Sensor overload");

}

delay(250);

}

private:

void configureSensor(void) {

tsl.enableAutoRange(true);

tsl.setIntegrationTime(TSL2561\_INTEGRATIONTIME\_402MS);

}

};

Tsl2561 tsl2561;

void setup(void) {

Serial.begin(9600);

tsl2561.start();

}

void loop(void) {

tsl2561.getLux();

}

# APÊNDICE 05 – CÓDIGO ANEMOMÊTRO

const int RecordTime = 3; //Define Measuring Time (Seconds)

const int SensorPin = 3; //Define Interrupt Pin (2 or 3 @ Arduino Uno)

int InterruptCounter;

float WindSpeed;

void setup()

{

Serial.begin(9600);

}

void loop() {

meassure();

Serial.print("Wind Speed: ");

Serial.print(WindSpeed); //Speed in km/h

Serial.print(" km/h - ");

Serial.print(WindSpeed / 3.6); //Speed in m/s

Serial.println(" m/s");

}

void meassure() {

InterruptCounter = 0;

attachInterrupt(digitalPinToInterrupt(SensorPin), countup, RISING);

delay(1000 \* RecordTime);

detachInterrupt(digitalPinToInterrupt(SensorPin));

WindSpeed = (float)InterruptCounter / (float)RecordTime \* 2.4;

}

void countup() {

InterruptCounter++;

}

# APÊNDICE 06 – CÓDIGO FONTE COM INTEGRAÇÃO DOS PRINCIPAIS SENSORES

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_TSL2561\_U.h>

#include <Adafruit\_BME280.h>

#define PressaoaNivelDoMar\_HPA (1013.25)

Adafruit\_TSL2561\_Unified tsl = Adafruit\_TSL2561\_Unified(TSL2561\_ADDR\_FLOAT, 12345);

Adafruit\_BME280 bme;

class Tsl2561 {

public:

float lux;

void start() {

if (!tsl.begin()) {

Serial.println("ERROR: No TSL2561");

exit(0);

}

configureSensor();

}

void getLux() {

sensors\_event\_t event;

tsl.getEvent(&event);

if (event.light) {

Serial.println(event.light);

lux = event.light;

} else {

Serial.println("Sensor overload");

}

delay(250);

}

private:

void configureSensor(void) {

tsl.enableAutoRange(true);

tsl.setIntegrationTime(TSL2561\_INTEGRATIONTIME\_402MS);

}

};

class Bme280 {

public:

float temp;

float umid;

float pres;

void start() {

bme.begin(0x76);

}

void getData() {

temp = bme.readTemperature();

umid = bme.readHumidity();

pres = bme.readPressure() / 100.0F;

}

};

class Ml8511 {

public:

float uvInt;

void getUV() {

int uv\_Level = analogRead\_average(UV\_OUT);

int ref\_Level = analogRead\_average(REF\_3V3);

float output\_Voltage = 3.3 / ref\_Level \* uv\_Level;

float uvIntensity = mapfloat(output\_Voltage, 0.99, 2.8, 0.0, 15.0);

uvInt = uvIntensity;

delay(500);

}

private:

int UV\_OUT = A0;

int REF\_3V3 = A1;

int analogRead\_average(int pinToRead) {

int NumberOfSamples = 8;

int runningValue = 0;

for(int x = 0; x < NumberOfSamples; x++)

runningValue += analogRead(pinToRead);

runningValue /= NumberOfSamples;

return(runningValue);

}

float mapfloat(float x, float in\_min, float in\_max, float out\_min, float out\_max) {

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

};

Tsl2561 tsl2561;

Bme280 bme280;

Ml8511 ml8511;

void setup(void) {

Serial.begin(9600);

tsl2561.start();

bme280.start();

}

void printValues() {

Serial.print("BME280: T:");

Serial.print(bme280.temp);

Serial.print(" P: ");

Serial.print(bme280.pres);

Serial.print(" U: ");

Serial.println(bme280.umid);

Serial.print("TSL2561: LUX: ");

Serial.println(tsl2561.lux);

Serial.print("ML8511: ");

Serial.println(ml8511.uvInt);

Serial.print("----------------------------------------------------");

}

void loop(void) {

tsl2561.getLux();

bme280.getData();

ml8511.getUV();

printValues();

delay(1000);

}

# APÊNDICE 07 – CÓDIGO FINAL

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_TSL2561\_U.h>

#include <Adafruit\_BME280.h>

#include <SPI.h>

#include <Ethernet.h>

#include <EthernetUdp.h>

#define PressaoaNivelDoMar\_HPA (1013.25)

Adafruit\_TSL2561\_Unified tsl = Adafruit\_TSL2561\_Unified(TSL2561\_ADDR\_FLOAT, 12345);

Adafruit\_BME280 bme;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress serverIP(192, 168, 43, 36);

int serverPort = 5290; //5290

char path[] = "/";

EthernetClient client;

int InterruptCounter;

void countup() {

InterruptCounter++;

}

class Anemometro {

private:

const int RecordTime = 3;

const int SensorPin = 3;

void meassure() {

InterruptCounter = 0;

attachInterrupt(digitalPinToInterrupt(SensorPin), countup, RISING);

delay(1000 \* RecordTime);

detachInterrupt(digitalPinToInterrupt(SensorPin));

WindSpeed = (float)InterruptCounter / (float)RecordTime \* 2.4;

}

public:

float WindSpeed;

float getData() {

meassure();

}

};

class Pluviometro {

private:

const float mmPerPulse = 0.173;

int sensor = 0;

int statusBefore = 0;

void getMm() {

for (int n = 0; n <= 30; n++) {

sensor = digitalRead(9);

if (sensor != statusBefore) {

mmTotal = mmTotal + mmPerPulse;

}

delay(100);

statusBefore = sensor;

}

}

public:

float mmTotal = 0;

void start() {

pinMode(9, INPUT);

}

void getData() {

getMm();

}

};

class Tsl2561 {

public:

float visible;

float inf;

void start() {

if (!tsl.begin()) {

Serial.println("ERROR: No TSL2561");

exit(0);

}

configureSensor();

}

void getLux() {

uint16\_t broadband;

uint16\_t ir;

tsl.getLuminosity(&broadband, &ir);

inf = ir;

visible = tsl.calculateLux(broadband, ir);

}

private:

void configureSensor(void) {

tsl.enableAutoRange(true);

tsl.setIntegrationTime(TSL2561\_INTEGRATIONTIME\_402MS);

}

};

class Bme280 {

public:

float temp;

float umid;

float pres;

void start() {

bme.begin(0x76);

}

void getData() {

temp = bme.readTemperature();

umid = bme.readHumidity();

pres = bme.readPressure() / 100.0F;

}

};

class Ml8511 {

public:

float uvInt;

void getUV() {

int uv\_Level = analogRead\_average(UV\_OUT);

int ref\_Level = analogRead\_average(REF\_3V3);

float output\_Voltage = 3.3 / ref\_Level \* uv\_Level;

float uvIntensity = mapfloat(output\_Voltage, 0.99, 2.8, 0.0, 15.0);

uvInt = uvIntensity;

delay(500);

}

private:

int UV\_OUT = A0;

int REF\_3V3 = A1;

int analogRead\_average(int pinToRead) {

int NumberOfSamples = 8;

int runningValue = 0;

for(int x = 0; x < NumberOfSamples; x++)

runningValue += analogRead(pinToRead);

runningValue /= NumberOfSamples;

return(runningValue);

}

float mapfloat(float x, float in\_min, float in\_max, float out\_min, float out\_max) {

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

};

class W5100 {

public:

void start() {

Ethernet.begin(mac);

macAddress = getMacAddress();

}

void send(float temperatura, float pressao, float umidade, float luzUV, float luzNormal, float luzInfravermelha, float velocidadeVento, float nivelPrecipitacao) {

post(temperatura, pressao, umidade, luzUV, luzNormal, luzInfravermelha, velocidadeVento, nivelPrecipitacao);

}

private:

String macAddress;

String getMacAddress() {

byte mac[6];

String macAddr = "";

Ethernet.MACAddress(mac);

for (int i = 0; i < 6; ++i) {

if (mac[i] < 16) {

macAddr += "0";

}

macAddr += String(mac[i], HEX);

if (i < 5) {

macAddr += ":";

}

}

return macAddr;

}

void post(float temperatura, float pressao, float umidade, float luzUV, float luzNormal, float luzInfravermelha, float velocidadeVento, float nivelPrecipitacao) {

if (client.connect(serverIP, serverPort)) {

client.println("GET /api/NimbusBox/postNimbusProduct?nrTemperatura=" + String(temperatura) + "&nrPressao=" + String(pressao) + "&nrUmidade=" + String(umidade) + "&nrLuzUv=" + String(luzUV) + "&nrEspectroLuz=" + String(luzNormal) + "&nrInfraVermelho=" + String(luzInfravermelha) + "&nrVelVento=" + String(velocidadeVento) + "&nrAguaAltura=" + String(nivelPrecipitacao) + " HTTP/1.1"); ///api/NimbusBox/getAllNimbusBox

client.println("Host: " + String(serverIP));

client.println("Content-Type: application/x-www-form-urlencoded");

//client.println("Content-Length: " + String(pData.length()));

client.println();

delay(1000);

while (client.available()) {

char c = client.read();

Serial.write(c);

}

client.stop();

Serial.println("\nDados enviados com sucesso!");

} else {

Serial.println("Falha ao se conectar ao servidor");

}

}

};

Tsl2561 tsl2561;

Bme280 bme280;

Ml8511 ml8511;

Anemometro ane;

Pluviometro pluv;

W5100 w5100;

void setup(void) {

Serial.begin(9600);

tsl2561.start();

bme280.start();

pluv.start();

w5100.start();

}

void printValues() {

Serial.print("BME280: T:");

Serial.print(bme280.temp);

Serial.print(" P: ");

Serial.print(bme280.pres);

Serial.print(" U: ");

Serial.println(bme280.umid);

Serial.print("TSL2561: visible: ");

Serial.println(tsl2561.visible);

Serial.print("TSL2561: Infrared: ");

Serial.println(tsl2561.inf);

Serial.print("ML8511: ");

Serial.println(ml8511.uvInt);

Serial.print("Anemometer: ");

Serial.println(ane.WindSpeed);

Serial.print("Pluviometer: ");

Serial.println(pluv.mmTotal);

Serial.print("----------------------------------------------------");

}

void loop(void) {

tsl2561.getLux();

bme280.getData();

ml8511.getUV();

ane.getData();

pluv.getData();

printValues();

w5100.send(bme280.temp, bme280.pres, bme280.umid, ml8511.uvInt, tsl2561.visible, tsl2561.inf, ane.WindSpeed, pluv.mmTotal);

delay(1000);

}