**CAMPUS UNIRUY WYDEN PARALELA**

**PROJETO DE EXTENSÃO IOT EM PYTHON**

**Arthur Motta Cumming – 202303532946**

**André Felipe Xavier Oliveira Santos – 202308603949**

**Laís Medeiros Costa Gonçalves – 202308705301**

**Letícia Medeiros Costa Gonçalves – 202308705296**

**Roberta Sued Nascimento Gomes De Santana** **– 202308425986**

**Thiago Emanoel Santos Araújo – 202302374735**

**Professor orientador: Vitor Emmanuel**

**2025**

**Salvador/BA**

Sumário

[1. DIAGNÓSTICO E TEORIZAÇÃO 3](#_Toc119686561)

[1.1. Identificação das partes interessadas e parceiros 3](#_Toc119686562)

[1.2. Problemática e/ou problemas identificados 3](#_Toc119686563)

[1.3. Justificativa 3](#_Toc119686564)

[1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos) 3](#_Toc119686565)

[1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão) 3](#_Toc119686566)

[2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO 4](#_Toc119686567)

[2.1. Plano de trabalho (usando ferramenta acordada com o docente) 4](#_Toc119686568)

[2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los. 4](#_Toc119686569)

[2.3. Grupo de trabalho (descrição da responsabilidade de cada membro) 4](#_Toc119686570)

[2.4. Metas, critérios ou indicadores de avaliação do projeto 4](#_Toc119686571)

[2.5. Recursos previstos 5](#_Toc119686572)

[2.6. Detalhamento técnico do projeto 5](#_Toc119686573)

[3. ENCERRAMENTO DO PROJETO 5](#_Toc119686574)

[3.1. Relatório Coletivo (podendo ser oral e escrita ou apenas escrita) 5](#_Toc119686575)

[3.2. Avaliação de reação da parte interessada 5](#_Toc119686576)

[3.3. Relato de Experiência Individual 5](#_Toc119686577)

[3.1. CONTEXTUALIZAÇÃO 5](#_Toc119686578)

[3.2. METODOLOGIA 6](#_Toc119686579)

[3.3. RESULTADOS E DISCUSSÃO: 6](#_Toc119686580)

[3.4. REFLEXÃO APROFUNDADA 6](#_Toc119686581)

[3.5. CONSIDERAÇÕES FINAIS 6](#_Toc119686582)

# DIAGNÓSTICO E TEORIZAÇÃO

## Identificação das partes interessadas e parceiros

O projeto conta com a participação de 6 (seis) integrantes, e todos colaboraram ativamente em todas as etapas da confecção do trabalho com ajuda do docente Vitor Emmanuel.

## Problemática e/ou problemas identificados

O projeto propõe o uso de um NodeMCU ESP8266 conectado a um sensor capaz de realizar essas leituras e monitoramento de dados ambientais (temperatura, pressão e fizemos o cálculo de altitude) de forma remota, utilizando uma rede Wi-Fi, e enviá-las por meio do protocolo MQTT.

## Justificativa

Este trabalho foi desenvolvido com o objetivo de aplicar, de forma prática, os conhecimentos adquiridos na disciplina de IoT em Python. A ideia central é realizar a leitura de temperatura, e pressão utilizando um sensor conectado ao microcontrolador NodeMCU ESP8266, com transmissão dos dados via Wi-Fi por meio do protocolo MQTT. A justificativa da proposta está na importância de compreender, na prática, como funciona a comunicação entre dispositivos embarcados e o envio de dados para a internet, algo essencial em aplicações reais de IoT. Além disso, o projeto serve como base para aplicações futuras mais complexas, como sistemas de automação, monitoramento ambiental ou integração com plataformas em nuvem. Por ser fácil implementação, também se mostra acessível para fins educacionais, incentivando assim o aprendizado ativo e colaborativo entre nos integrantes do grupo.

## Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)

O objetivo foi implementar a leitura de temperatura e pressão por meio de sensores conectados ao NodeMCU ESP8266, transmitir os dados dos sensores em tempo real via Wi-Fi utilizando o protocolo MQTT, e aplicar, na prática, os conhecimentos da disciplina de IoT em Python por meio da integração entre hardware, comunicação sem fio e linguagem de programação. Essa situação é relevante pois permite a criação de soluções inteligentes para ambientes internos, contribuindo para o conforto e segurança. A demanda foi identificada a partir de discussões em grupo e com base nos conteúdos abordados na disciplina, evidenciando a importância de aplicar conceitos de IoT de maneira prática.

## Referencial teórico (subsídio teórico para propositura de ações da extensão)

O presente projeto visa o desenvolvimento de soluções de automação e Internet das Coisas (IoT) utilizando o NodeMCU (ESP8266), programado com MicroPython, uma alternativa leve e poderosa para programação de microcontroladores.

* **Contribuições do Professor Vitor**

O Professor Vitor tem sido uma referência fundamental no desenvolvimento de projetos de IoT, com uma abordagem prática e teórica que abrange desde os conceitos básicos até a aplicação de soluções mais complexas. Ele enfatiza a importância do uso de ferramentas como o MicroPython para a programação de dispositivos como o NodeMCU, possibilitando o controle remoto de sistemas de forma simples e eficiente. Seu repositório no GitHub (https://github.com/vitorpq/WydenClasses) oferece materiais valiosos, como códigos de exemplo e tutoriais práticos, que são essenciais para o sucesso do projeto.

* **Referenciais Teóricos sobre NodeMCU e MicroPython**

Fábio Souza: https://www.youtube.com/@fabio\_maker

Fábio Souza, através de seu canal, contribui significativamente para a educação em sistemas embarcados. Embora seu foco principal seja o ESP8266, sua abordagem de integração de microcontroladores com MicroPython e plataformas de IoT é extremamente útil para quem deseja entender como programar o NodeMCU de forma eficiente. Ele ensina a trabalhar com sensores, atuadores e comunicação sem fio, permitindo que o aluno desenvolva projetos práticos de automação.

LNPBR: https://www.youtube.com/@LNPBR

O canal LNPBR é uma excelente fonte de aprendizado sobre o uso do NodeMCU e MicroPython em projetos de automação. Filipeflop oferece tutoriais detalhados sobre a instalação e programação do MicroPython no NodeMCU, com ênfase em como conectar sensores e dispositivos para criar soluções de IoT. Seus vídeos são bem explicados e ideais para iniciantes.

Great Scott!: https://www.youtube.com/@greatscottlab

O canal Great Scott! é uma excelente alternativa que aborda temas de eletrônica, incluindo o uso de NodeMCU e outros microcontroladores. Embora seu foco principal não seja exclusivamente MicroPython, ele oferece conteúdo relevante sobre a programação de microcontroladores, sistemas embarcados e automação. Em seus vídeos, ele explora a construção de circuitos e o controle de dispositivos, muitas vezes utilizando plataformas como NodeMCU e ESP8266. As abordagens de Great Scott! são detalhadas e com uma profundidade que ajuda a entender como aplicar diferentes tecnologias na construção de projetos de IoT.

# PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

## Plano de trabalho (usando ferramenta acordada com o docente)

**Etapa 1 – Entrega e preparação do hardware**

**Objetivo:** Preparar a placa NodeMCU ESP8266 para desenvolvimento de aplicações em Python, utilizando o firmware MicroPython e a ferramenta esptool.

A placa NodeMCU (ESP8266) foi entregue à equipe para iniciar os testes e configurações iniciais e iniciamos a preparação do ambiente de desenvolvimento.

* Conectamos a placa ao computador via cabo USB.
* Identificamos a porta COM correta no Gerenciador de Dispositivos (que pode variar dependendo da porta utilizada).
* Instalamos os pacotes necessários no terminal da IDE PyCharm com o seguinte comando: pip install esptool adafruit-ampy
* esptool: ferramenta usada para se comunicar com a placa pela porta serial, permitindo apagar e gravar a memória flash.
* adafruit-ampy: ferramenta que permite enviar arquivos .py diretamente para a placa.

**Etapa 2 – Apagando a memória da placa**

Antes da gravação do firmware com suporte a MicroPython, foi necessário apagar completamente a memória flash da placa.

Utilizando o comando: esptool --port COMX erase\_flash

**Observação:** A porta COM5 pode variar de acordo com o sistema operacional e a porta USB utilizada. É necessário verificar a porta correta no Gerenciador de Dispositivos (Windows).

Com a memória devidamente apagada, a placa está pronta para receber o firmware MicroPython e testar a comunicação enviando um script simples, como o main.py

**Etapa 3 – instalando o firmware**

Com a ferramenta instalada e a porta identificada, acessamos o site oficial do MicroPython e baixamos a versão mais recente do firmware .bin compatível com o modelo do nosso NodeMCU

Após a limpeza, carregamos o firmware para o NodeMCU com o seguinte comando:

esptool.py --port COMX --baud 460800 write\_flash --flash\_size=detect 0 nome-do-firmware.bin

Por fim, para interagir com o NodeMCU utilizamos o programa PuTTY como terminal serial. Configuramos a porta serial correta, selecionamos a velocidade de 115200 bauds, e ao conectar, tivemos acesso ao prompt do MicroPython (>>>), e começamos a testar nossos primeiros comandos na placa. Assim, finalizamos a instalação com sucesso e deixamos o NodeMCU pronto para desenvolvimento com MicroPython.

Esse processo permitiu configurar o ambiente necessário para desenvolver aplicações embarcadas de forma prática, utilizando Python como linguagem principal.

**Etapa 4 - Validação e teste de conexão**

Antes do desenvolvimento do código de envio do PDF, realizamos testes em sala de aula para validar a conexão Wi-Fi e o envio de dados simples e aleatórios para um servidor MQTT. Esses testes iniciais serviram para garantir que o NodeMCU estava se comunicando corretamente com o broker e que o protocolo MQTT estava funcionando conforme esperado.

Após a validação, seguimos com o seguinte processo no código:

Conectamos o NodeMCU à rede Wi-Fi, utilizando o nome e senha da rede local.

Estabelecemos conexão com o broker MQTT, com as configurações específicas de usuário, servidor, porta e tópico.

E fizemos o envio de alguns dados aleatórios para teste de conexão.

**Etapa 5 – Realização do Upload do PDF**

Com a conexão Wi-Fi testada, desenvolvemos o código responsável por enviar o **link do arquivo PDF** hospedado no GitHub diretamente para a plataforma **ThingsBoard Cloud**, utilizando o protocolo **HTTP POST** com urequests.

**Resumo do que foi feito:**

* Definimos as configurações da rede Wi-Fi e da API do ThingsBoard:
  + SSID e senha
  + Token do dispositivo (ACCESS\_TOKEN)
  + URL de destino da API do ThingsBoard

No código, ao estabelecer a conexão Wi-Fi, enviamos um dicionário JSON contendo a URL do arquivo PDF.

**Etapa 6 – Soldagem, teste e análise do sensor BMP280 com envio para o Adafruit IO**

Após a configuração inicial do ambiente e envio de dados via HTTP para o ThingsBoard, passamos para uma nova fase do projeto: a **integração do sensor BMP280** para monitoramento ambiental com o NodeMCU ESP8266.

**Soldagem do sensor**

O sensor **BMP280**, utilizado para a medição de **temperatura, pressão atmosférica e cálculo da altitude**, chegou em sua forma básica e foi necessário soldar os pinos de conexão antes do uso. Utilizamos ferro de solda e estanho, com cuidado para garantir conexões limpas e seguras nos seguintes pinos:

* **VCC** → Alimentação
* **GND** → Terra
* **SCL** → Clock I2C (ligado ao D1 no NodeMCU)
* **SDA** → Dados I2C (ligado ao D2 no NodeMCU)

**Configuração do barramento I2C e comunicação com o BMP280**

Após a soldagem, realizamos os primeiros testes de leitura I2C com o sensor, utilizando a biblioteca personalizada de leitura e compensação dos dados crus. Validamos a comunicação com o sensor verificando as leituras de temperatura e pressão em tempo real no terminal.

**Cálculo da Altitude com base em Temperatura e Pressão**

Além das leituras diretas de temperatura e pressão fornecidas pelo sensor BMP280, realizamos o cálculo da **altitude aproximada** em relação ao nível do mar. Esse cálculo é essencial para aplicações ambientais, meteorológicas e de geolocalização em projetos IoT.

**Parâmetros utilizados:**

* **p₀ (pressão ao nível do mar):** 1022 hPa *(valor médio ajustado para Salvador - Pituaçu)*
* **p (pressão lida pelo sensor):** em hPa
* **T (temperatura lida pelo sensor):** em °C

**Fórmula utilizada:**

A fórmula utilizada é baseada na equação barométrica padrão da atmosfera internacional:

altitude = (((p0\_hpa / p\_hpa) \*\* (1 / 5.257)) - 1) \* (t\_celsius + 273.15) / 0.0065

**Envio de dados ambientais ao Adafruit IO**

Com os dados sendo lidos corretamente, o próximo passo foi o envio dessas informações para a plataforma **Adafruit IO**, que permite criar **dashboards** personalizados para visualização remota em tempo real. Para isso:

* Realizamos a conexão com a rede Wi-Fi usando o SSID e senha fornecidos.
* Utilizamos a API REST do Adafruit IO para enviar as leituras para os feeds temp, pressao e alt.

**Feeds utilizados:**

* temp: Temperatura em Celsius
* pressao: Pressão atmosférica em hPa
* alt: Altitude aproximada em metros, calculada com base em Salvador (p₀ = 1022 hPa)

As leituras foram feitas a cada **20 segundos**, durante **2 minutos**, totalizando 6 envios por feed, com os seguintes dados processados:

* Temperatura: leitura direta do sensor.
* Pressão: leitura direta do sensor.
* Altitude: cálculo baseado na fórmula física padrão de barometria.

**Visualização e análise dos dados no Adafruit IO**

Após os envios, acessamos o painel da conta do Adafruit IO e criamos **dashboards interativos** com gráficos para monitoramento em tempo real das variáveis. Cada feed foi vinculado a um componente gráfico correspondente, permitindo observar:

* Variações de temperatura ambiente durante o intervalo.
* Oscilações na pressão atmosférica.
* Cálculo dinâmico da altitude com base nas condições locais.

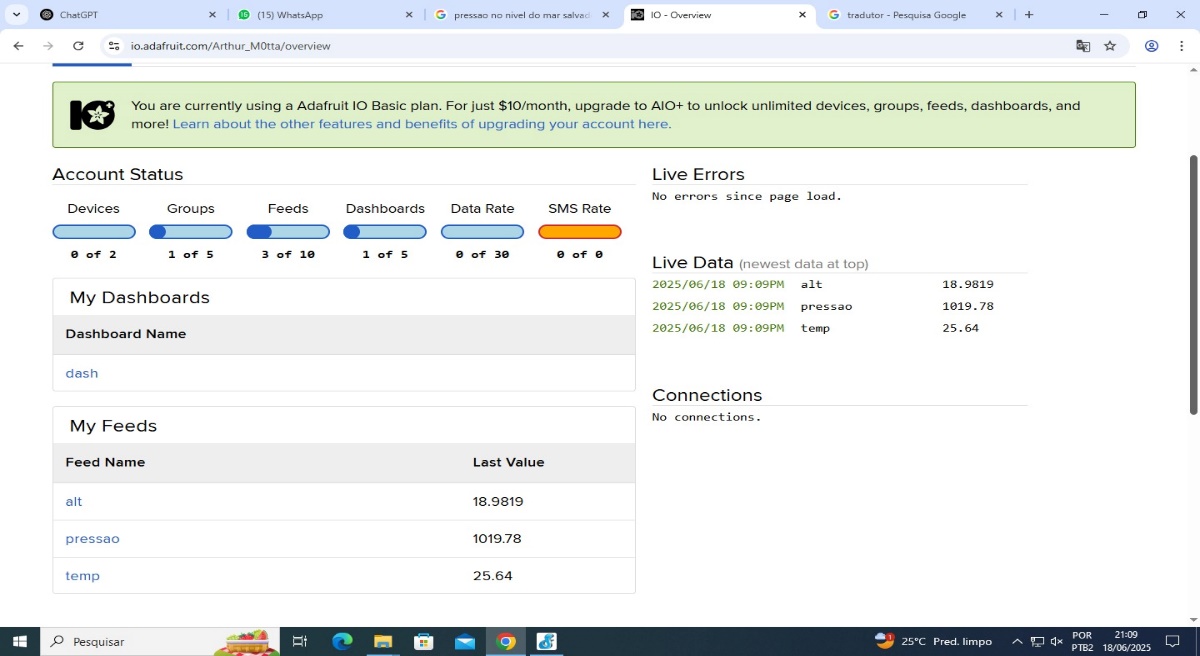
A plataforma também permite configurar alertas e histórico dos dados, ampliando o uso do NodeMCU como estação meteorológica compacta.

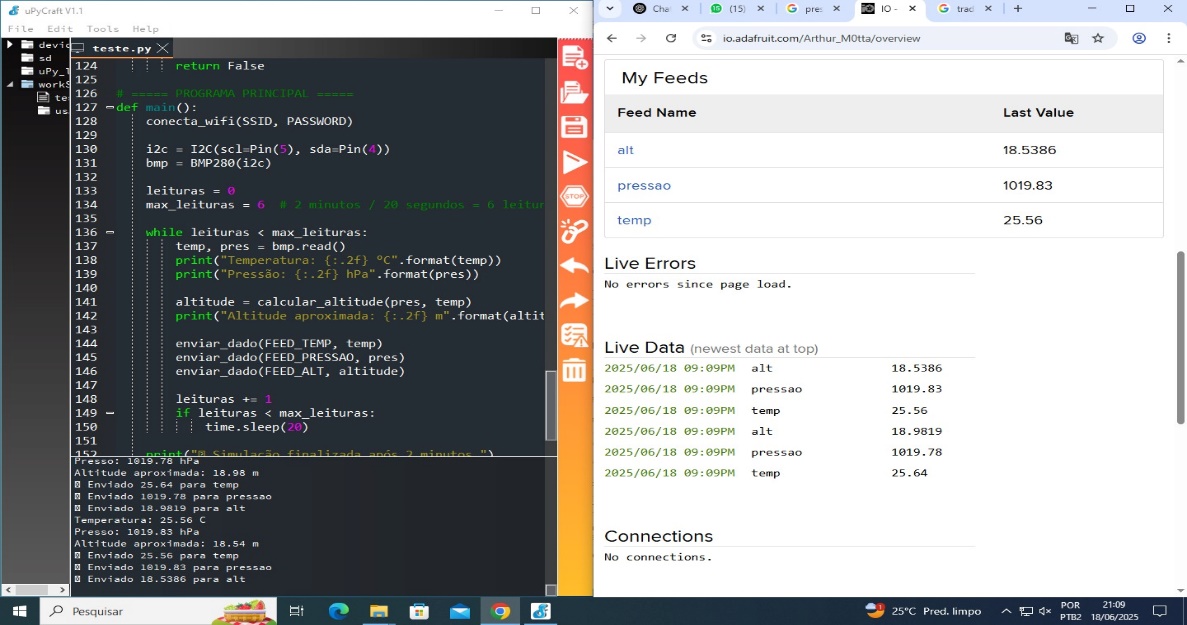
**Conclusão desta etapa**

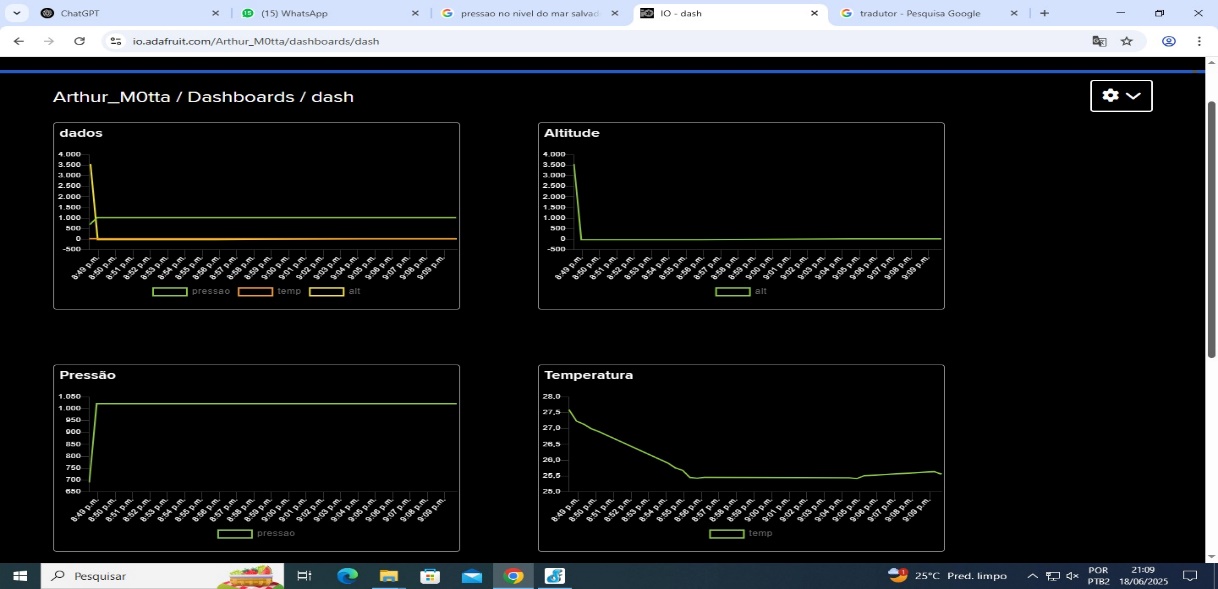
Esta fase consolidou o uso do **NodeMCU ESP8266 com sensores I2C e APIs REST**, demonstrando sua capacidade de atuar como **dispositivo IoT completo**. A integração com o Adafruit IO mostrou-se prática e visualmente eficaz para o acompanhamento dos dados em tempo real.

Para acessar e visualizar os códigos do projeto acesse o link:

<https://github.com/ArthurM0tta/IOT>



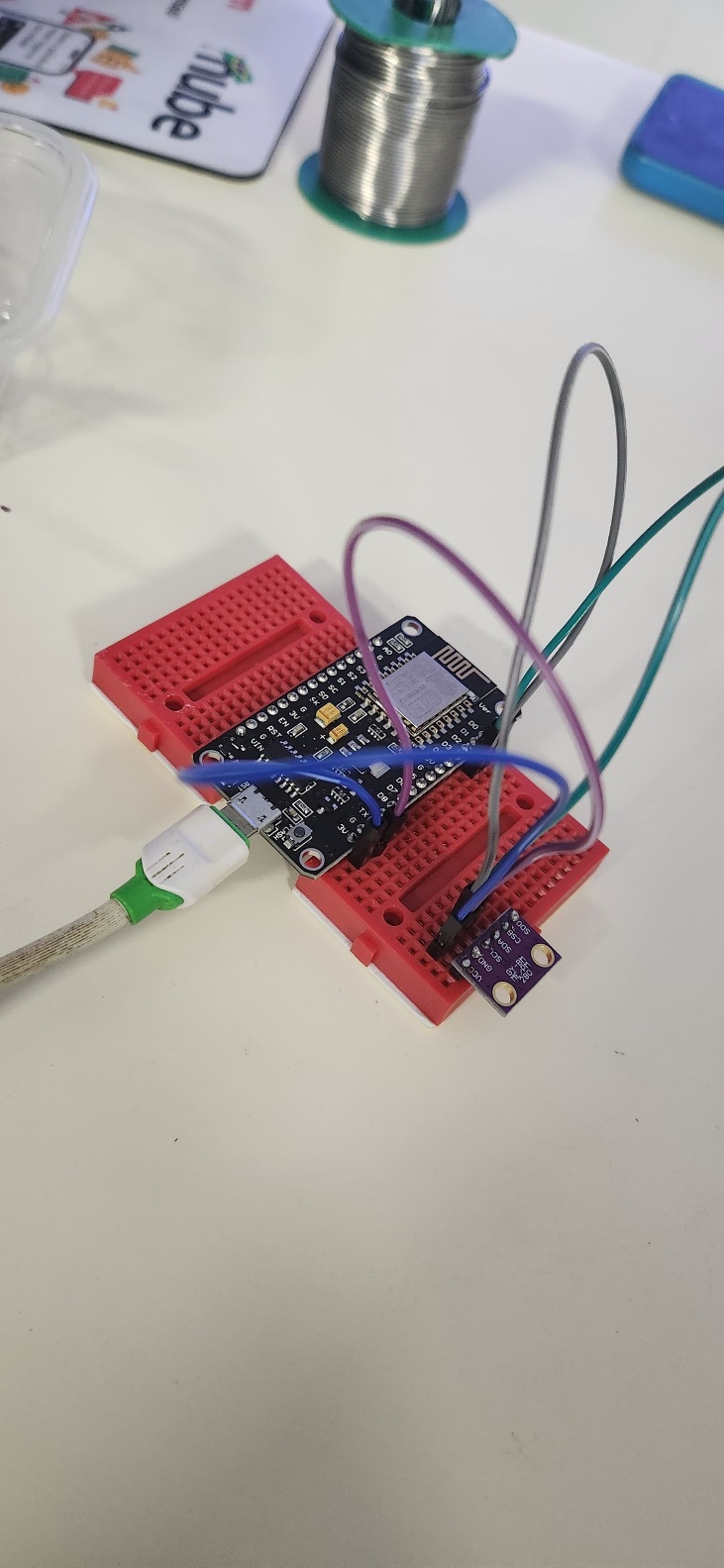


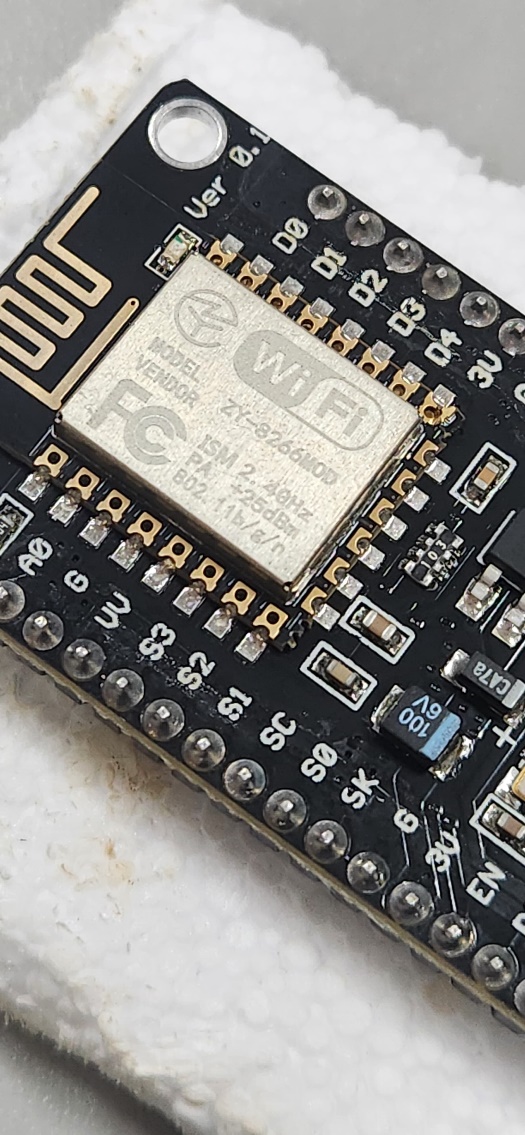


## Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los.

A elaboração do projeto foi realizada de forma colaborativa, envolvendo os colegas do grupo e dois professores, um responsável pela disciplina de IoT, e outro de uma área complementar. As ideias iniciais surgiram a partir de conversas e trocas de experiências durante as aulas e reuniões na biblioteca da faculdade, onde discutimos a aplicação prática dos conceitos estudados. A avaliação do projeto foi feita com base em retornos dos próprios colegas e dos professores, que acompanharam a evolução do trabalho e deram feedbacks quanto à clareza, funcionamento e possibilidades de melhoria. Essa interação entre os membros do grupo e os professores foi fundamental para o amadurecimento do projeto e para garantir que o resultado fosse relevante tanto do ponto de vista acadêmico quanto prático.

Como forma de evidenciar esse processo colaborativo, foram registradas capturas de fotos realizados.



## Grupo de trabalho (descrição da responsabilidade de cada membro)

O grupo de trabalho será composto por 6 membros, com a divisão de responsabilidades conforme a seguir:

**Membros Arthur e Thiago: Escrita e Ambientação dos Códigos**

Responsabilidades:

* Escrever e adaptar os códigos para o NodeMCU, incluindo a integração dos sensores com a comunicação via MQTT.
* Configurar a comunicação do dispositivo com a rede Wi-Fi e a transmissão dos dados via MQTT para o servidor.
* Realizar ajustes e melhorias no código para garantir o envio correto e eficiente dos dados.

Atividades:

* Desenvolver scripts em Python para manipulação e envio dos dados dos sensores.
* Testar e depurar a comunicação entre os sensores, o NodeMCU e o servidor MQTT.
* Trabalhar na implementação do protocolo de comunicação e na estruturação do código para a coleta de dados.

**Membros Roberta e André: Montagem e Funcionamento dos Sensores**

Responsabilidades:

* Montar os sensores de temperatura, umidade e pressão no hardware (NodeMCU).
* Realizar testes de funcionamento dos sensores e garantir que estão capturando e fornecendo dados corretamente.
* Manter o bom funcionamento dos sensores durante o processo de transmissão dos dados.

Atividades:

* Conectar os sensores ao NodeMCU e testar a leitura dos dados.
* Garantir a integridade física e funcional dos sensores.
* Diagnosticar e solucionar problemas de falhas nos sensores.

**Membros Laís e Letícia: Entendimento e Utilização do Hardware**

Responsabilidades:

* Compreender a arquitetura do NodeMCU e os sensores envolvidos, além de estudar as especificações e capacidades do hardware.
* Realizar a configuração do hardware, garantindo que o NodeMCU e os sensores estejam funcionando corretamente.
* Acompanhar a interação entre o software e o hardware para garantir uma integração eficiente.

Atividades:

* Estudar as especificações do NodeMCU e dos sensores.
* Trabalhar na alimentação elétrica e nas conexões do NodeMCU e sensores.
* Ajudar na resolução de problemas relacionados ao funcionamento do hardware.

Embora as responsabilidades do grupo estejam distribuídas conforme as áreas de atuação, é importante destacar que todos os membros se envolveram ativamente nas diferentes etapas do projeto. Independentemente da sua função principal, cada integrante contribuiu de maneira colaborativa tanto nas áreas de software quanto de hardware, buscando otimizar o entendimento global e a evolução do projeto. Esse trabalho conjunto foi fundamental para aprimorar o processo de aprendizado, garantir a integração eficiente entre as partes e alcançar os resultados desejados de forma mais eficaz.

## Metas, critérios ou indicadores de avaliação do projeto

Para garantir que o projeto atinja os objetivos estabelecidos, será necessário acompanhar de perto o progresso de cada etapa. As metas, critérios e indicadores de avaliação serão definidos da seguinte forma:

**Meta 1: Implementação do sistema de leitura de sensores (Temperatura e Pressão)**

Critério de Avaliação: O sistema deve ser capaz de ler e fornecer dados consistentes de temperatura e pressão em tempo real.

Indicadores:

* Sensores funcionais e conectados corretamente ao NodeMCU.
* Dados obtidos de cada sensor devem ser enviados com precisão para o servidor MQTT.

**Meta 2: Transmissão dos dados via protocolo MQTT**

Critério de Avaliação: Os dados dos sensores devem ser transmitidos sem falhas em tempo real via Wi-Fi utilizando MQTT.

**Meta 3: Aplicação prática dos conhecimentos da disciplina de IoT em Python**

Critério de Avaliação: O código desenvolvido deve ser funcional e eficiente, implementando corretamente as funções de integração entre hardware e software.

Indicadores:

O código deve ser escrito de forma modular e bem documentada, com cada função clara e objetiva.

**Meta 4: Validação da solução em ambiente real**

Critério de Avaliação: O sistema de monitoramento deve ser validado em um ambiente real, como uma sala ou ambiente interno, para testar a aplicabilidade da solução.

## Recursos previstos

Para o desenvolvimento deste projeto, serão necessários recursos materiais, institucionais e humanos. A seguir, estão detalhados os principais recursos previstos para a execução do projeto de forma eficiente e com baixo custo:

**Recursos Materiais**

* Hardware:

NodeMCU: utilizado para comunicação via Wi-Fi e controle dos sensores.

* Sensores:

Sensores de temperatura e pressão (BMP280) para coleta de dados ambientais.

* Cabos e conectores:

Utilizados para realizar as conexões entre o NodeMCU e os sensores.

* Fonte de Alimentação:

Para alimentar o NodeMCU e os sensores, garantindo a continuidade do funcionamento.

Esses materiais são de baixo custo e já possuem disponibilidade na instituição.

**Recursos Institucionais**

Espaço físico: o projeto será desenvolvido no laboratório de informática da instituição, onde os estudantes terão acesso ao ambiente adequado para o manuseio dos componentes de hardware e testes do sistema.

Equipamentos Institucionais: O uso dos computadores e da rede Wi-Fi da instituição será fundamental para a programação e testes do sistema, além da utilização de servidores para realizar as conexões MQTT.

**Recursos Humanos**

Equipe de Estudantes: Os seis membros do grupo de trabalho possuem conhecimentos pré-existentes na área de programação e eletrônica, e estão comprometidos com a execução das tarefas conforme o plano de trabalho descrito anteriormente.

Orientação Acadêmica: O professor responsável pela disciplina fornecerá supervisão técnica e acadêmica, garantindo a qualidade do desenvolvimento do projeto e a aplicação dos conceitos de IoT e comunicação sem fio.

**Fonte de Recursos Financeiros**

Dada a natureza do projeto, não serão necessários recursos financeiros adicionais, uma vez que os materiais e recursos necessários já estão disponíveis na instituição.

Uso de Ferramentas de Software Gratuitas: A programação será realizada utilizando o MicroPython e o MQTT, que são plataformas de código aberto e gratuitas, evitando gastos com licenças de software.

O planejamento visa a execução do projeto com o mínimo de custo, maximizando o uso de recursos já disponíveis na instituição.

## Detalhamento técnico do projeto

O projeto teve como foco o desenvolvimento monitoramento ambiental em tempo real, utilizando sensores a um microcontrolador NodeMCU ESP8266, com transmissão de dados via Wi-Fi utilizando o protocolo MQTT. A aplicação prática dos conhecimentos em Internet das Coisas (IoT) permitiu uma abordagem interdisciplinar envolvendo eletrônica, redes e programação em Python.

**Etapas técnicas do desenvolvimento:**

* Escolha e configuração dos sensores:

Foram utilizados sensores para medir temperatura e pressão. Os sensores selecionados foram conectados ao NodeMCU ESP8266, com alimentação adequada e testes de comunicação inicial.

* Programação do NodeMCU (firmware):

O microcontrolador foi programado em Python. O código realiza a leitura periódica dos sensores e publica os dados via protocolo MQTT para um broker.

* Comunicação com o broker MQTT:

Utilizou-se o protocolo MQTT para a comunicação entre o NodeMCU e um broker (Mosquitto ou HiveMQ). Os dados são enviados a tópicos específicos (ex: /sensor/temperatura, sensor/pressao).

* Implementação da interface de recepção em Python:

Foi desenvolvido um script em Python utilizando a biblioteca paho-mqtt para subscrever os tópicos e receber os dados em tempo real. Os dados foram apresentados no terminal.

* 5. Testes e validação:

Foram realizados testes em ambiente interno simulando condições reais de uso. Verificou-se a precisão das leituras, a estabilidade da conexão Wi-Fi e a eficiência da comunicação MQTT. Ajustes no código e nos sensores foram feitos com base nos testes.

* 6. Registros e documentação:

Durante o desenvolvimento, foram gerados registros como fotos do protótipo, capturas de tela do código e da execução do sistema, além de mensagens trocadas entre os membros da equipe para fins de documentação técnica e comprovação da construção coletiva.

Essa abordagem permitiu integrar teoria e prática, promovendo a consolidação dos conteúdos abordados na disciplina e oferecendo uma solução de fácil adaptação para diferentes contextos de uso.

# ENCERRAMENTO DO PROJETO

## Relato Coletivo:

Nosso projeto teve como principal objetivo a implementação de um sistema IoT capaz de realizar a coleta de dados ambientais (temperatura, pressão e um cálculo que fizemos para altitude) por meio do sensor conectado ao microcontrolador NodeMCU ESP8266, transmissão desses dados via Wi-Fi utilizando o protocolo MQTT. Durante o desenvolvimento, buscamos aplicar todos os nossos conhecimentos adquiridos, nas aulas teóricas e práticas, e também para mostrando a viabilidade de soluções tecnológicas acessíveis e de fácil implementação. Conseguimos cumprir nossos objetivos e atingir uma boa parte dos resultados esperados, especialmente em relação à contribuição para o aprendizado prático de IoT e à sensibilização sobre os benefícios do monitoramento remoto de dados ambientais. A possibilidade de integrar essas tecnologias com outras plataformas, como a nuvem, também abre um leque de aplicações futuras em várias áreas.

### Avaliação de reação da parte interessada

Como o projeto foi realizado no nosso ambiente acadêmico e com foco no aprendizado entre os nos integrantes do grupo, não houve uma interação com uma parte interessada externa, como comunidade ou usuários fora da sala de aula. Porém, realizamos uma avaliação de reação interna entre os seis membros do grupo para avaliar como cada um percebeu o desenvolvimento e os resultados alcançados. Essa avaliação foi feita por meio de discussões em grupo, nas quais cada membro compartilhou suas experiências durante o projeto.

## Relato de Experiência Individual (Pontuação específica para o relato individual)

Aluna: Laís Medeiros Costa Gonçalves -202308705301

### CONTEXTUALIZAÇÃO

O projeto tem como foco o uso de um NodeMCU ESP8266 conectado a um sensor capaz de realizar essas leituras e monitoramento de dados ambientais, utilizando sensor de temperatura e pressão. Fui responsável pela configuração do **hardware**, ou seja, pelo estudo das especificações do NodeMCU e sensores, garantindo que todos os componentes estivessem corretamente conectados e funcionais. Meu papel também incluiu a resolução de problemas técnicos que surgiram durante a integração entre o hardware e o software, além de auxiliar na alimentação elétrica dos dispositivos.

### METODOLOGIA

A experiência foi vivenciada principalmente em um laboratório na própria faculdade onde a maior parte das atividades ocorreram. O grupo foi composto por seis integrantes, que se dividiram nas seguintes tarefas:

* Pesquisa e estudo de sensores de temperatura, pressão e o cálculo de altitude;
* Programação do microcontrolador NodeMCU ESP8266;
* Implementação do protocolo MQTT para envio dos dados via Wi-Fi;
* Testes e validação dos dados coletados.

Foi feito testes de conexão do NODEMCU com o sensor BMP280 para ter as leituras dos sensores. O sensor foi conectado ao NodeMCU utilizando o protocolo I2C, com as seguintes ligações: VCC no 3V do NodeMCU, GND no G, SCL no pino D1 e SDA no pino D2. Essa configuração garantiu a comunicação adequada entre os dispositivos.

### RESULTADOS E DISCUSSÃO:

A expectativa inicial do projeto era desenvolver uma aplicação prática que possibilitasse a leitura de dados ambientais pelo sensor BMP280 para ter as leituras de temperatura e pressão, utilizando um sensor conectado ao NodeMCU ESP8266 e transmitindo essas informações por meio do protocolo MQTT via Wi-Fi. Durante a experiência, observei que eu e minha equipe foi capaz de realizar a coleta dos dados com precisão e transmiti-los de forma eficiente para um broker MQTT. A comunicação entre o NodeMCU e o sensor foi estável, e os dados recebidos no cliente MQTT correspondiam aos valores esperados, confirmando o correto funcionamento da integração entre hardware e software. O projeto atendeu às expectativas, proporcionando uma rica experiência de aprendizado e servindo como um excelente ponto de partida para projetos mais robustos na área de Internet das Coisas.

### REFLEXÃO APROFUNDADA

Ao comparar a experiência vivida como foi dito no relato coletivo, percebi que o desafio de integrar hardware e software, que na teoria parece ser relativamente simples, é na prática algo bem mais complexo. O aprendizado teórico sobre microcontroladores e sensores foi crucial, mas a execução real do projeto me ensinou a importância de testar cada parte do sistema constantemente e de trabalhar em equipe para resolver problemas inesperados.

### CONSIDERAÇÕES FINAIS

Este projeto proporcionou uma base sólida para o entendimento das **tecnologias de comunicação em IoT**, especialmente o uso do **protocolo MQTT**. Essa experiência foi extremamente relevante para mim, pois o MQTT é um dos protocolos mais utilizados em sistemas de automação para troca de dados entre dispositivos de maneira eficiente e em tempo real. A possibilidade de **implementar o MQTT em ambientes industriais** pode trazer grandes benefícios, como a **monitorização remota de sistemas**, controle de equipamentos e otimização de processos. Saber como configurar e integrar dispositivos utilizando esse protocolo permite que eu aplique esse conhecimento diretamente em soluções industriais mais complexas, como sistemas de **automação predial**, **controle de máquinas** e **monitoramento de processos industriais**. Em minha área de atuação, a **comunicação eficiente entre dispositivos** é crucial, e o MQTT pode facilitar a integração de diferentes componentes, mesmo em ambientes industriais com grande quantidade de dados.

## 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

Aluno: Arthur Motta Cumming - 202303532946

### 3.2.1. CONTEXTUALIZAÇÃO

Durante este projeto prático, participei do desenvolvimento de um sistema embarcado utilizando um NodeMCU com o sensor BMP280 para monitoramento de variáveis ambientais: temperatura e pressão, a partir das quais foi calculada a altitude utilizando uma fórmula física implementada no código. O objetivo principal era coletar esses dados e enviá-los para a plataforma Adafruit IO via conexão Wi-Fi, tornando possível o acompanhamento remoto em tempo real. O sensor BMP280 foi conectado ao NodeMCU por meio de uma protoboard, o que facilitou a prototipagem e os testes do circuito. Esse projeto foi desenvolvido como parte das atividades da disciplina de Internet das Coisas (IoT), e envolveu programação em MicroPython, manipulação de sensores e integração com serviços em nuvem. Além disso, a plataforma Adafruit IO possibilitou a criação automática de dashboards visuais, que exibem a variação dos dados em gráficos de linha e outros formatos, permitindo uma visualização clara e dinâmica das mudanças nas variáveis monitoradas ao longo do tempo.

### 3.2.2. METODOLOGIA

A experiência foi vivenciada em ambiente laboratorial, com parte do desenvolvimento também realizada em casa. Os sujeitos envolvidos foram os membros do grupo de trabalho e o professor orientador. O projeto ocorreu ao longo de várias semanas, com foco no aprendizado incremental e testes constantes.

As etapas principais foram:

* Configuração do ambiente de desenvolvimento: instalação do firmware MicroPython no NodeMCU e testes básicos com o sensor BMP280.
* Conexão do NodeMCU ao Wi-Fi usando a biblioteca network.
* Leitura dos dados do BMP280.
* Cálculo da altitude com base nos valores de temperatura e pressão, utilizando uma fórmula padrão para conversão física.
* Envio dos dados para o Adafruit IO por meio de requisições HTTP com a biblioteca urequests.
* Visualização automática dos dados em dashboards do Adafruit IO, que atualizaram os gráficos conforme os valores eram enviados.
* Testes de precisão e estabilidade, verificando tanto os dados quanto a conectividade com o servidor.

### 3.2.3. RESULTADOS E DISCUSSÃO:

Expectativa vs. realidade: Esperava-se um processo simples e direto, mas a experiência revelou complexidades práticas, especialmente na comunicação com o sensor e no envio de dados.

Observações:

* O sensor BMP280 mostrou-se preciso e confiável nas leituras de temperatura e pressão.
* A altitude foi calculada dinamicamente com base nos dados obtidos, não sendo uma medição direta do sensor.
* A conexão com a plataforma Adafruit IO foi bem-sucedida após ajustes no formato dos dados e na autenticação.
* Os dashboards automáticos do Adafruit IO mostraram-se extremamente úteis, permitindo acompanhar graficamente a variação de temperatura, pressão e altitude em tempo real.

Aprendizagens:

* Entendimento aprofundado da comunicação e estrutura de sensores.
* Compreensão de como calcular a altitude com base em parâmetros físicos e do comportamento do ar atmosférico.
* Conhecimento sobre o uso de bibliotecas específicas em MicroPython.
* Compreensão da estrutura de APIs REST e da autenticação via chave (API Key).

Facilidades:

* A biblioteca do MicroPython para rede e requisições HTTP é simples e direta.
* A plataforma Adafruit IO possui uma boa documentação e cria dashboards automaticamente.
* O uso da protoboard tornou os testes de conexão rápidos e práticos.

Dificuldades:

* Dificuldade de conexão com o NodeMCU em algumas máquinas, devido provavelmente a problemas com o driver.
* Delay nas respostas HTTP em caso de sinal fraco de Wi-Fi.

### 3.2.4. REFLEXÃO APROFUNDADA

Ao comparar a experiência prática com a teoria estudada em sala, notei o quanto os conceitos se tornam mais claros na aplicação real. Por exemplo, entender a leitura e o funcionamento do sensor BMP280 é muito mais intuitivo ao lidar diretamente com os dados extraídos pelo NodeMCU. Além disso, o cálculo da altitude a partir da pressão e temperatura envolveu o uso de fórmulas físicas reais, como a equação barométrica, o que fortaleceu a interdisciplinaridade entre física, programação e engenharia. A teoria forneceu a base, mas foi na prática que realmente compreendi os desafios de lidar com hardware real, como ruídos, alimentação, precisão e latência de comunicação.

### 3.2.5. CONSIDERAÇÕES FINAIS

Como continuidade do projeto, seria interessante:

* Integrar uma bateria e um sistema de gerenciamento de energia para tornar o projeto portátil.
* Testar sensores alternativos como o BME280, que também mede umidade.
* Implementar um algoritmo de suavização dos dados de altitude para evitar oscilações em tempo real.
* Integrar a biblioteca pandas do Python para realizar tanto a análise quanto a limpeza dos dados retirados do sensor.

Essa experiência mostrou o potencial da IoT e ampliou minha visão sobre como sensores, microcontroladores e serviços em nuvem podem se integrar para formar soluções reais, escaláveis e inteligentes. A funcionalidade de dashboards automáticos do Adafruit IO foi um diferencial importante, proporcionando uma visualização clara da variação dos dados e contribuindo para análises mais rápidas e eficientes.

## 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

Aluno: André Felipe Xavier Oliveira Santos - 202308603949

### 3.2.1. CONTEXTUALIZAÇÃO

Neste projeto, desenvolvi um sistema embarcado usando o NodeMCU junto com o sensor BMP280 para medir temperatura e pressão. Com esses dados, calculei a altitude utilizando uma fórmula física dentro do código. O objetivo principal foi enviar essas informações para a plataforma Adafruit IO via Wi-Fi, permitindo o monitoramento em tempo real de qualquer lugar. O sensor foi conectado ao NodeMCU com uma protoboard, facilitando a montagem e os testes, tudo dentro da disciplina de Internet das Coisas (IoT).

### 3.2.2. METODOLOGIA

O trabalho foi feito entre o laboratório e minha casa, com ajuda do grupo e orientação do professor. As principais etapas foram: instalar o MicroPython no NodeMCU; conectar o dispositivo ao Wi-Fi; ler os dados do sensor; calcular a altitude; enviar os dados para o Adafruit IO usando requisições HTTP; e, por fim, acompanhar os dados em dashboards que atualizam automaticamente. Durante o processo, fiz vários testes para garantir a precisão das leituras e a estabilidade da conexão.

### 3.2.3. RESULTADOS E DISCUSSÃO:

Embora a ideia fosse simples, enfrentei alguns desafios, principalmente na comunicação com o sensor e na transmissão dos dados. O sensor BMP280 apresentou leituras confiáveis de temperatura e pressão, e a altitude calculada se mostrou consistente, apesar de ser uma estimativa baseada em fórmula. A conexão com o Adafruit IO foi ajustada e estabilizada, e os dashboards facilitaram muito a visualização dos dados em tempo real. Essa experiência permitiu entender melhor a integração entre hardware, software e serviços em nuvem.

### 3.2.4. REFLEXÃO APROFUNDADA

A prática tornou muito mais claro o que vimos na teoria, especialmente o funcionamento do sensor BMP280 e o cálculo da altitude com base em pressão e temperatura. Foi importante perceber como o trabalho com hardware real traz desafios, como ruídos e atrasos na comunicação, que não apresenta na teoria. Além disso, pude unir conceitos de física e programação, o que tornou o aprendizado mais completo.

### 3.2.5. CONSIDERAÇÕES FINAIS

Para continuar o projeto, penso em incluir uma bateria para torná-lo portátil, testar sensores que também medem umidade, como o BME280, e desenvolver um filtro para suavizar as variações da altitude. Também gostaria de explorar ferramentas para analisar e limpar os dados coletados, como a biblioteca pandas do Python. Esse trabalho mostrou o quanto a IoT pode criar soluções inteligentes que unem sensores, microcontroladores e nuvem de forma eficiente e prática.

## 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

Aluna: Letícia Medeiros Costa Gonçalves - 202308705296

### 3.2.1. CONTEXTUALIZAÇÃO

Minha participação neste projeto foi ativa e essencial em todas as etapas do desenvolvimento. Desde o início, estive envolvida na parte de fazer a alimentação elétrica e nas conexões do NodeMCU e sensores. Como o microcontrolador NodeMCU ESP8266 e o sensor BMP280, com o propósito de realizar a leitura de dados ambientais (temperatura, pressão e cálculo da altitude) e enviá-los via protocolo MQTT. Além da parte técnica, participei dos testes práticos para verificar a estabilidade da conexão, a frequência das leituras e a integridade dos dados enviados. Com base nos resultados, realizei ajustes no código para garantir o funcionamento correto do sistema.

### 3.2.2. METODOLOGIA

A metodologia adotada neste projeto seguiu uma abordagem prática e experimental, dividida em etapas que permitiu eu e minha equipe a integração dos componentes físicos, configuração da comunicação e testes. As principais etapas foram para:

* Compreender a arquitetura do NodeMCU e os sensores envolvidos, além de estudar as especificações e capacidades do hardware.
* Realizar a configuração do hardware, garantindo que o NodeMCU e os sensores estejam funcionando corretamente.
* Acompanhar a interação entre o software e o hardware para garantir uma integração eficiente.

### 3.2.3. RESULTADOS E DISCUSSÃO:

De início pensei que a comunicação entre o node e o uso do protocolo fosse uma etapa tranquila, já que o NodeMCU é um microcontrolador bastante acessível. No entanto, surgiram desafios, especialmente na parte de garantir a alimentação elétrica estável e na configuração dos sensores. Foi necessário fazer ajustes nas conexões e realizar testes contínuos para garantir que as leituras estavam sendo feitas. Durante o processo, percebi que a interação entre o hardware e o software exigia um cuidado extra. Embora o código tivesse sido escrito corretamente, a comunicação entre o NodeMCU e o servidor MQTT, pois um processo de muita aprendizagem e também de frustração, especialmente quando o node não funcionava como esperado. No entanto, quando finalmente conseguimos fazer o sistema funcionar, foi uma sensação de grande realização. A colaboração dentro do grupo foi fundamental para superar os obstáculos.

### 3.2.4. REFLEXÃO APROFUNDADA

Esse projeto foi uma maneira de colocar em prática tudo o que aprendi na disciplina de IoT. Usei o NodeMCU ESP8266 para capturar dados de temperatura e pressão com um sensor e enviar essas informações pela rede Wi-Fi usando o protocolo MQTT. O principal objetivo foi entender como os dispositivos se comunicam com a internet, o que é fundamental para aplicações de IoT no mundo real.

A parte mais interessante foi ver como a comunicação entre o sensor, o microcontrolador e a rede funciona na prática. Configurar o MQTT foi um desafio, mas me ajudou a perceber como um protocolo eficiente é crucial para a troca de dados. Também enfrentei alguns problemas com a conectividade, já que a transmissão dos dados via Wi-Fi precisava ser estável para garantir que tudo funcionasse direitinho.

### 3.2.5. CONSIDERAÇÕES FINAIS

Essa experiência foi extremamente enriquecedora para minha formação, pois me permitiu aplicar de forma prática os conceitos aprendidos na disciplina de IoT em Python. Pude desenvolver habilidades importantes como programação embarcada, comunicação em rede e resolução de problemas técnicos. Além disso, este projeto ampliou meu interesse pela área de Internet das Coisas e me preparou melhor para futuros desafios acadêmicos e profissionais relacionados à automação e sistemas conectados.

## 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

Aluna: Roberta Sued Nascimento Gomes De Santana - 202308425986

### 3.2.1. CONTEXTUALIZAÇÃO

A experiência foi desenvolvida no contexto da disciplina de Internet das Coisas (IoT), com foco na criação de uma aplicação prática envolvendo sensores e envio de dados para a nuvem. O projeto consistiu na conexão de um microcontrolador ao sensor BMP280, responsável pela coleta de dados de temperatura, pressão atmosférica e cálculo da altitude, com envio dessas informações para a plataforma Adafruit IO.

### 3.2.2. METODOLOGIA

O projeto foi desenvolvido em ambiente universitário, com reuniões presenciais e por videochamada. Os envolvidos incluíram os membros do grupo e o professor-orientador. A experiência teve duração de duas semanas e envolveu as seguintes etapas:

* Levantamento dos componentes necessários e estudo da documentação do sensor BMP280 e da plataforma Adafruit IO;
* Configuração da rede Wi-Fi e do ambiente de desenvolvimento;
* Escrita do código de coleta de dados e envio à nuvem utilizando MicroPython;
* Testes práticos com leituras reais de temperatura e pressão;
* Discussão dos resultados e refinamento do código para redução de erros e aumento da precisão.

### 3.2.3. RESULTADOS E DISCUSSÃO:

Antes de iniciar, acreditava que a integração entre sensores e conectividade Wi-Fi seria um grande desafio. No entanto, com o apoio do grupo e o estudo aprofundado das bibliotecas disponíveis, o processo tornou-se viável e até mesmo estimulante.

Facilidades: a estrutura do código em Python foi clara, o que facilitou a leitura e a adaptação das funções ao nosso contexto.

Dificuldades: enfrentamos erros de sintaxe relacionados às limitações do MicroPython, como a inicialização incorreta do método \_\_init\_\_, além de instabilidades na conexão com a internet.

Aprendizados: adquiri conhecimentos sobre a comunicação via I2C, envio de dados por meio de API REST, além da importância de tratar exceções e modularizar bem o código. Compreendi também a relevância do planejamento das etapas do projeto e o impacto positivo da colaboração em equipe.

Foi especialmente gratificante ver os dados sendo coletados, processados e enviados com sucesso — e observar as leituras sendo exibidas em tempo real na nuvem.

### 3.2.4. REFLEXÃO APROFUNDADA

A experiência prática foi um complemento essencial ao conteúdo teórico discutido em sala de aula. Conceitos como sensores, comunicação digital e uso de plataformas em nuvem foram aplicados de forma concreta e eficiente. A lógica de cálculo da altitude a partir da pressão atmosférica evidenciou a interdisciplinaridade entre computação, física e matemática.

A vivência reforçou a importância dos testes contínuos, da modularização do código e da adoção de boas práticas de desenvolvimento, mesmo em projetos de menor escala. Além disso, contribuiu para o desenvolvimento de habilidades técnicas e de trabalho em equipe que serão valiosas para projetos futuros.

### CONSIDERAÇÕES FINAIS

Para projetos futuros, seria interessante explorar a utilização de diferentes sensores e a integração de gráficos em tempo real, facilitando a visualização e análise dos dados. Também seria válido comparar plataformas de IoT, como Blynk e ThingSpeak, ou até mesmo desenvolver um aplicativo móvel para consumir os dados via API.

Esse tipo de projeto tem potencial para expansão, tanto em pesquisas acadêmicas quanto em iniciativas de extensão, como o monitoramento ambiental em comunidades. Alternativamente, poderíamos explorar soluções locais de armazenamento e visualização, com dashboards personalizados hospedados em servidores próprios.

## 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

Aluno: Thiago Emanoel Santos Araújo - 202302374735

### 3.2.1. CONTEXTUALIZAÇÃO

Este projeto teve como objetivo o desenvolvimento de um sistema embarcado utilizando o microcontrolador NodeMCU integrado ao sensor BMP280, responsável pela medição de temperatura e pressão atmosférica. A partir dessas variáveis, foi possível calcular a altitude por meio de uma equação física implementada no código. As informações obtidas foram transmitidas em tempo real para a plataforma Adafruit IO, utilizando conexão Wi-Fi, permitindo o monitoramento remoto dos dados. A montagem foi realizada com o auxílio de uma protoboard, facilitando os testes e a organização do circuito. Todas as atividades foram conduzidas no contexto da disciplina de Internet das Coisas (IoT).

### 3.2.2. METODOLOGIA

O desenvolvimento do projeto ocorreu tanto no laboratório quanto em ambiente doméstico, com suporte do grupo de trabalho e sob orientação do professor responsável. As etapas seguidas incluíram: instalação do firmware MicroPython no NodeMCU; configuração da conexão Wi-Fi; leitura dos dados do sensor BMP280; cálculo da altitude com base nas medições; envio dos dados para a plataforma Adafruit IO por meio de requisições HTTP; e visualização dos dados em painéis (dashboards) dinâmicos. Durante o processo, foram realizados diversos testes com o intuito de assegurar a precisão das leituras e a estabilidade da comunicação com a rede.

### 3.2.3. RESULTADOS E DISCUSSÃO:

A configuração inicial do NodeMCU apresentou certo grau de complexidade, principalmente nas etapas de instalação e comunicação com o ambiente de desenvolvimento. Contudo, após essa fase inicial, o processo evoluiu de maneira mais fluida. Destaca-se a flexibilidade proporcionada pelo uso do MicroPython, que permitiu a coleta e o registro contínuo dos dados em tempo real, contribuindo significativamente para o monitoramento eficaz das variáveis ambientais.

### 3.2.4. REFLEXÃO APROFUNDADA

A experiência prática foi fundamental para consolidar os conhecimentos teóricos abordados em sala de aula, especialmente no que se refere ao funcionamento do sensor BMP280 e ao cálculo de altitude baseado nas variáveis físicas. O contato com hardware real evidenciou desafios que não são perceptíveis na teoria, como interferências, ruídos e instabilidades na comunicação. A integração entre conceitos de física e programação resultou em um aprendizado mais abrangente e significativo.

### CONSIDERAÇÕES FINAIS

Como proposta de continuidade do projeto, destaca-se a possibilidade de implementação de uma fonte de alimentação por bateria, conferindo portabilidade ao sistema. Sugere-se ainda o desenvolvimento de filtros para suavizar variações bruscas na medição da altitude, bem como o uso de bibliotecas como o Pandas (Python) para análise e tratamento dos dados coletados. Este trabalho evidenciou o potencial da Internet das Coisas (IoT) na criação de soluções inteligentes, integrando sensores, microcontroladores e plataformas em nuvem de forma eficiente e prática. Além disso, reforça a importância da coleta sistemática de dados para análises futuras sobre as variações climáticas locais, como as observadas na cidade de Salvador.