

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 Emanuel Santos Araújo – 202302374735

Professor orientador: Vitor Emmanuel

2025

Salvador/BA

Sumário

1.	DIAGNÓSTICO E TEORIZAÇÃO	3
1.1.	Identificação das partes interessadas e parceiros	3
1.2.	Problemática e/ou problemas identificados	3
1.3.	Justificativa	3
1.4.	Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)	3
1.5.	Referencial teórico (subsídio teórico para propositura de ações da extensão)	4
2.	PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO	5
2.1.	Plano de trabalho (usando ferramenta acordada com o docente)	5
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.	7
2.3.	Grupo de trabalho (descrição da responsabilidade de cada membro).....	9
2.4.	Metas, critérios ou indicadores de avaliação do projeto	10
2.5.	Recursos previstos	11
2.6.	Detalhamento técnico do projeto.....	13
3.	ENCERRAMENTO DO PROJETO.....	Error! Bookmark not defined.
3.1.	Relatório Coletivo (podendo ser oral e escrita ou apenas escrita)	Error! Bookmark not defined.
3.2.	Avaliação de reação da parte interessada.....	Error! Bookmark not defined.
3.3.	Relato de Experiência Individual.....	Error! Bookmark not defined.
3.1.	CONTEXTUALIZAÇÃO.....	Error! Bookmark not defined.
3.2.	METODOLOGIA	Error! Bookmark not defined.
3.3.	RESULTADOS E DISCUSSÃO:	Error! Bookmark not defined.
3.4.	REFLEXÃO APROFUNDADA.....	Error! Bookmark not defined.
3.5.	CONSIDERAÇÕES FINAIS.....	Error! Bookmark not defined.

1. DIAGNÓSTICO E TEORIZAÇÃO

1.1. 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.

1.2. 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, umidade e pressão) de forma remota, utilizando uma rede Wi-Fi, e enviá-las por meio do protocolo MQTT.

1.3. 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, umidade 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 os integrantes do grupo.

1.4. 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, umidade 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.

1.5. 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.

2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

2.1. Plano de trabalho (usando ferramenta acordada com o docente)

Etapas 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.

- Conectar a placa ao computador via cabo USB.
- Identificar a porta COM correta no Gerenciador de Dispositivos (que pode variar, dependendo da porta).
- Instalar 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, possibilitando apagar e gravar a memória flash.

adafruit-ampy: permite enviar arquivos .py diretamente para a placa.

Etapas 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 COM5 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 COM3 --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

Tudo assim testado, foi nos passado o passo a passo de como realizar a conexão com um novo servidor MQTT com o objetivo de realizar o upload deste PDF no intuito de demonstrar e aplicar aquilo que foi aprendido em sala.

Abrimos um arquivo PDF previamente salvo na memória do NodeMCU, assumindo que o arquivo exemplo.pdf já foi carregado corretamente.

Codificamos o conteúdo do arquivo em base64, para que os dados pudessem ser transmitidos de forma segura via MQTT.

Publicamos os dados codificados no tópico MQTT configurado, tornando-os disponíveis para visualização desse mesmo tópico.

Desconectamos do broker e encerramos o processo, liberando os recursos utilizados pelo sistema.

Esse passo a passo demonstrou a capacidade do NodeMCU, com MicroPython, de realizar tarefas mais avançadas como o envio de arquivos por meio de protocolos leves como o MQTT.

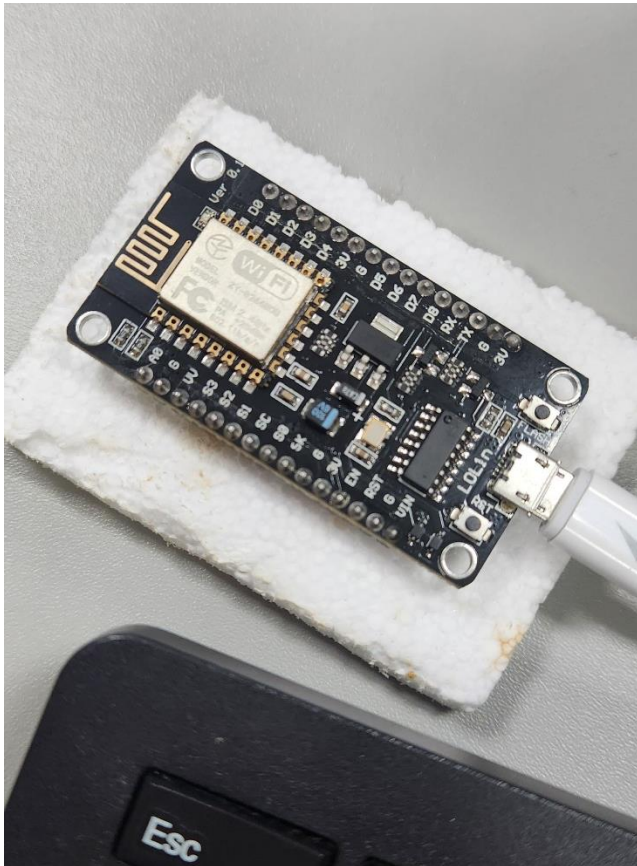
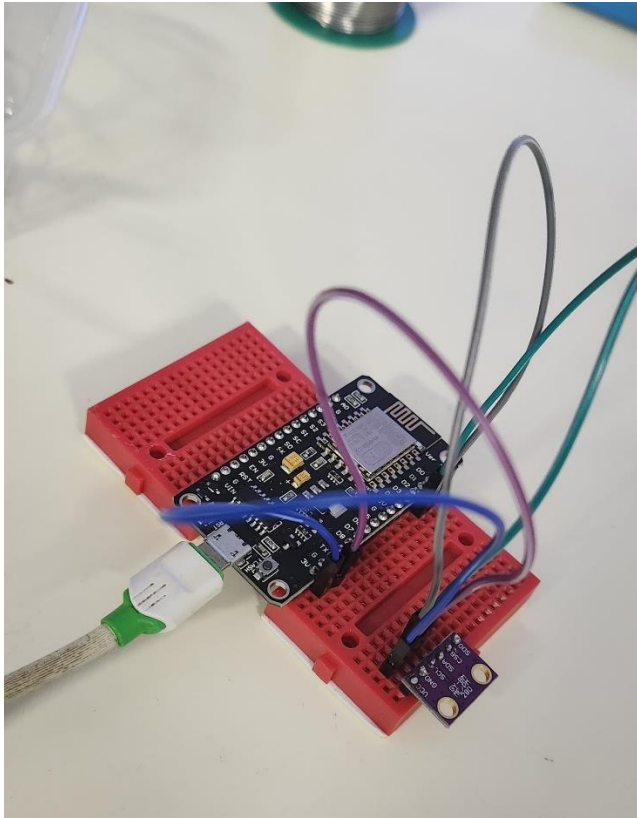
Para acessar e visualizar os códigos do projeto acesse o link:

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

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.

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.



2.3. 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.

2.4. 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, Umidade e Pressão)

Critério de Avaliação: O sistema deve ser capaz de ler e fornecer dados consistentes de temperatura, umidade 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.

2.5. 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, umidade 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-existent 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.

2.6. 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, umidade 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/umidade, /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.