



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



MILENA LIMA C ACQUESTA
GUSTAVO HENRIQUE DE LIMA E SILVA

OBJETOS INTELIGENTES CONECTADOS
SISTEMA DE MONITORAMENTO DE UMIDADE E TEMPERATURA

SÃO PAULO

2022

Abstract. This article describes the Arduino project with the aim of being used initially to be a versatile project, where we can adapt to different types of situations or usability, becoming the basis for other more complex projects. Examples: Monitoring the temperature and humidity of an air conditioner or a fan, but also even serving as a “thermometer”, where we receive updates on humidity and temperature via Whatsapp for people who have breathing problems, etc. Regarding the construction of the Hardware, we used an ESP32 board (NodeMCU) which in turn has a DHT22 sensor (humidity and temperature sensor) attached. The final result of the project is a device with an attached sensor capable of sending signals over the internet to the end user's whatsapp, using the MQTT protocol that supports the connection of the RED Node and, in turn, to a messaging API, where it has a BOT capable of reporting every time there is any change in the temperature and humidity data captured by the sensor.

Resumo. Este artigo descreve o projeto de Arduino com o objetivo de ser usado de forma inicial para ser um projeto versátil, onde consigamos adaptar para diversos tipos de situações ou usabilidades, se tornando base para outros projetos mais complexos. Exemplos: Monitoramento de temperatura e umidade de um ar condicionado ou de um ventilador, mas também até mesmo servir como um “termômetro”, onde recebemos atualizações sobre a umidade e temperatura via Whatsapp para pessoas que possuem problemas respiratórios, etc. Sobre a construção do Hardware, utilizamos uma placa ESP32 (NodeMCU) que por sua vez tem acoplado um sensor DHT22 (sensor de umidade e temperatura). O resultado final do projeto é um dispositivo com um sensor acoplado capaz de enviar sinais através da internet para o whatsapp do usuário final, utilizando o protocolo MQTT que dá suporte à conexão do Node RED e, por sua vez, à uma API de mensageria, onde possui um BOT capaz de reportar toda vez que há alguma modificação nos dados de temperatura e umidade captados pelo sensor.

SUMÁRIO

1.INTRODUÇÃO.....	5
2.MATERIAIS E MÉTODOS.....	6
3.RESULTADOS.....	16
4.CONCLUSÃO.....	21
5.REFERÊNCIAS BLOGRÁFICAS.....	22
6. LISTA DE FIGURAS.....	23

1. INTRODUÇÃO

O Projeto tem o objetivo ser, de forma pragmática, um pontapé inicial para a construção de um projeto mais complexo, onde possui uma placa ESP32 (NodeMCU) que está ligada a um sensor de umidade e temperatura (DHT22). Essa placa, por sua vez, poderia ser unificada a outros equipamentos que fossem necessários ter esse monitoramento (principalmente de forma virtual, conectado à internet e reportando informações de forma periódica). Assim que o sensor nota a alteração de umidade ou temperatura, imediatamente ele envia uma mensagem (através de uma API chamada CallMeBot) para o celular do usuário final. De forma simplificada, o processo de funcionamento acontece da seguinte forma: O Hardware (peças) e o código estão na plataforma Wokwi, que por sua vez estão conectados via MQTT no HiveMQ e MQTT Explorer que conecta ao Node RED. Dentro do Node RED temos uma orquestração para acionar o CallmeBot, que é uma API mensageiro capaz de sinalizar eventos via whatsapp. Sobre os problemas que são similares, a ideia do projeto foi colocada em prática depois de notarmos uma grande quantidade de dispositivos que entram em falha parcial ou total por falta desse tipo de medição... Superaquecimento, por exemplo, é um dos maiores causadores de falhas e de perdas totais de dispositivos eletrônicos atualmente. Alguns exemplos de uso: Monitoramento de temperatura e umidade de um ar condicionado, computador, celular ou de um ventilador, como também até mesmo servir como um “termômetro”, onde recebemos atualizações sobre a umidade e temperatura via Whatsapp para pessoas que possuem problemas respiratórios, etc.

Link para o projeto:

<https://wokwi.com/projects/349305302851519058>

2. MATERIAIS E MÉTODOS

Essa primeira seção contará com a informação dos módulos/hardware que fora utilizado no projeto final.

Demais detalhes sobre o funcionamento individual de cada componente estão descritos abaixo:

a. Placa ESP32 (NodeMCU):



Figura 1: Placa ESP32

O ESP32 é uma placa de desenvolvimento com Wifi e Bluetooth, ideal para desenvolvimento de inúmeras aplicações IoT (Internet of Things ou Internet das Coisas). É adequado para dispositivos inteligentes domésticos, controle sem fio industrial, monitoramento sem fio.

Equipado com CPU Xtensa® Dual-Core 32-bit LX6, antena embutida, interface usb-serial CP2102, e regulador de tensão 3.3V AMS1117. A programação pode ser feita em LUA ou usando a IDE do Arduino através de um cabo micro-usb. Com 4 MB de memória flash, o ESP32 ESP-WROOM-32, é uma solução ideal para aplicativos IoT.

b. Sensor DHT22



Figura 2: Sensor DHT22

O Sensor de Temperatura e Umidade DHT22, também conhecido como Sensor AM2302, é um sensor que faz a medição da temperatura e da umidade com alta precisão, sendo que é permitido fazer leituras de temperaturas entre -40° a 80° Celsius e umidade entre 0 a 100%.

O DHT22 funciona através de um sensor capacitivo de umidade e um termistor para medir o ar circundante, todos enviando informações para um microcontrolador de 8 bits que responde com um sinal digital para outro microcontrolador, como Arduino ou Pic. Ele trabalha com tensão de 3,5 a 5,5V e possui um baixo consumo de corrente, por volta de 1mA a 1,5mA, sendo que em stand by é de 40µA a 50µA, sem contar que sua precisão para medição de umidade chega a aproximadamente 2% RH e a de temperatura é de mais ou menos 0,5°C, tendo um excelente custo benefício para estudantes, amantes e profissionais da área de eletrônica.

- c. Descrição dos protocolos de comunicação, serviços utilizados, e outros detalhes que sejam importantes para o entendimento das técnicas utilizadas para o uso do protocolo MQTT:

Configurações necessárias:

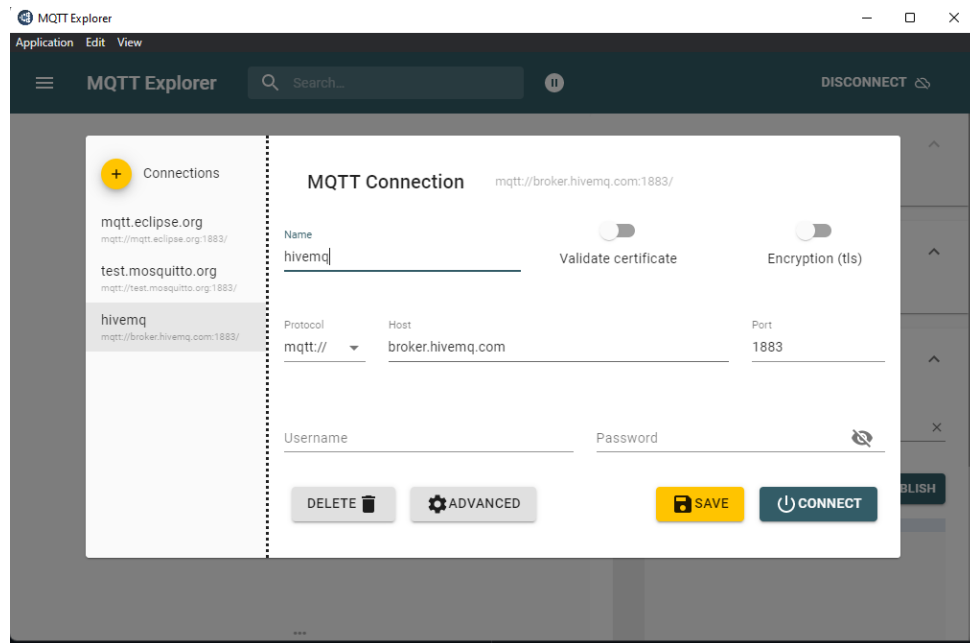


Figura 3: Configuração MQTT Explorer - Etapa I

Clicar no botão Advanced, add o tópicos exclusivo mackenzie20504039/#

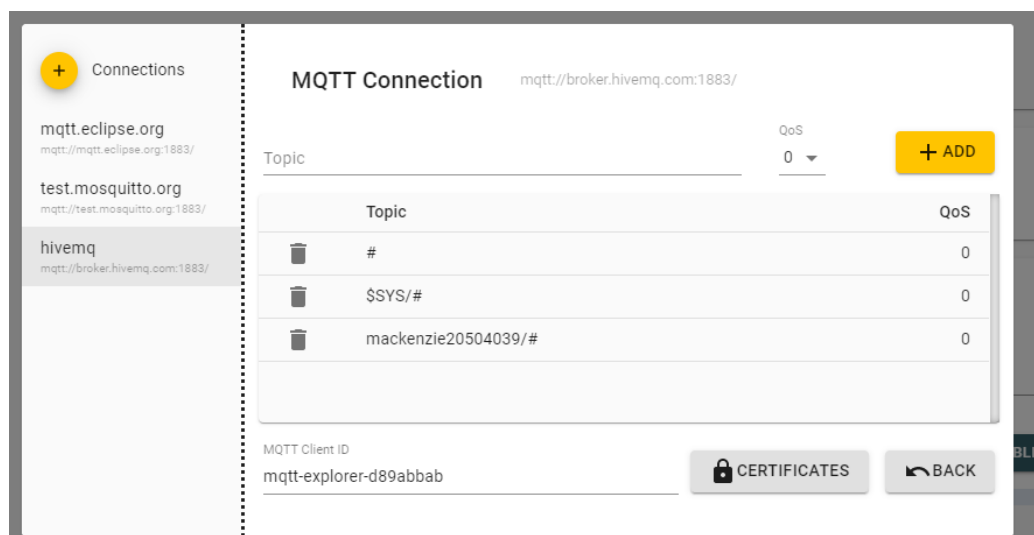


Figura 4: Configuração MQTT Explorer - Etapa II

Em seguida clique no botão de conectar

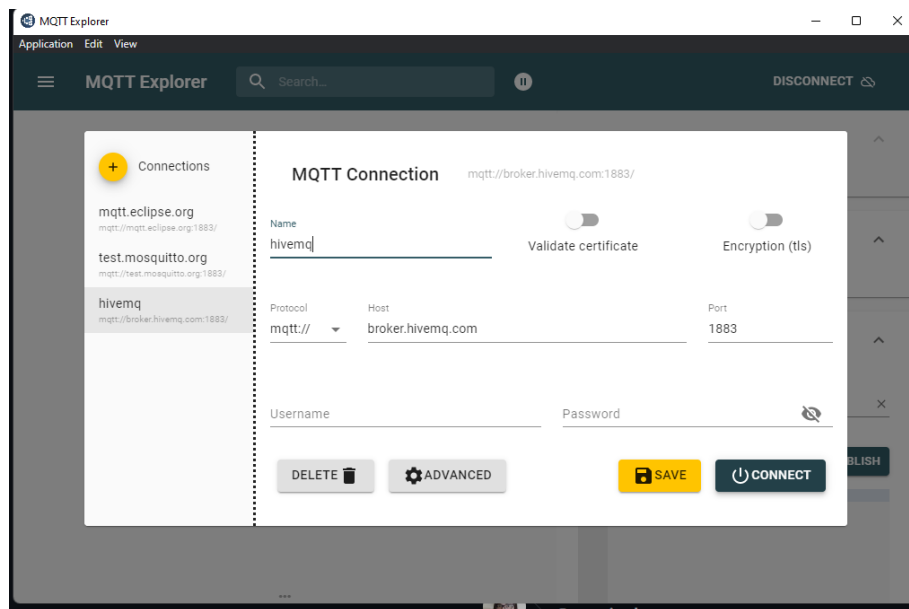


Figura 5: Configuração MQTT Explorer - Etapa III

Após conectar, volte ao projeto no Wokwi e execute:

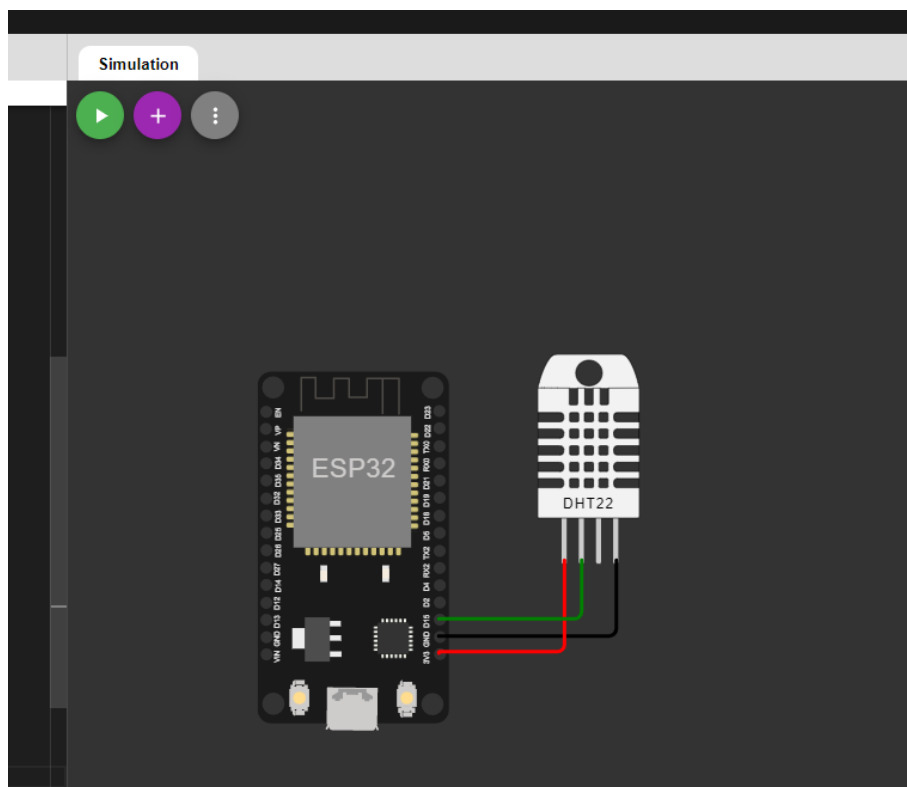


Figura 6: Execução do Projeto Wokwi

Após alguns segundos é conectado ao mqtt-explorer, onde podemos ver o histórico de temperatura e umidade do ar.

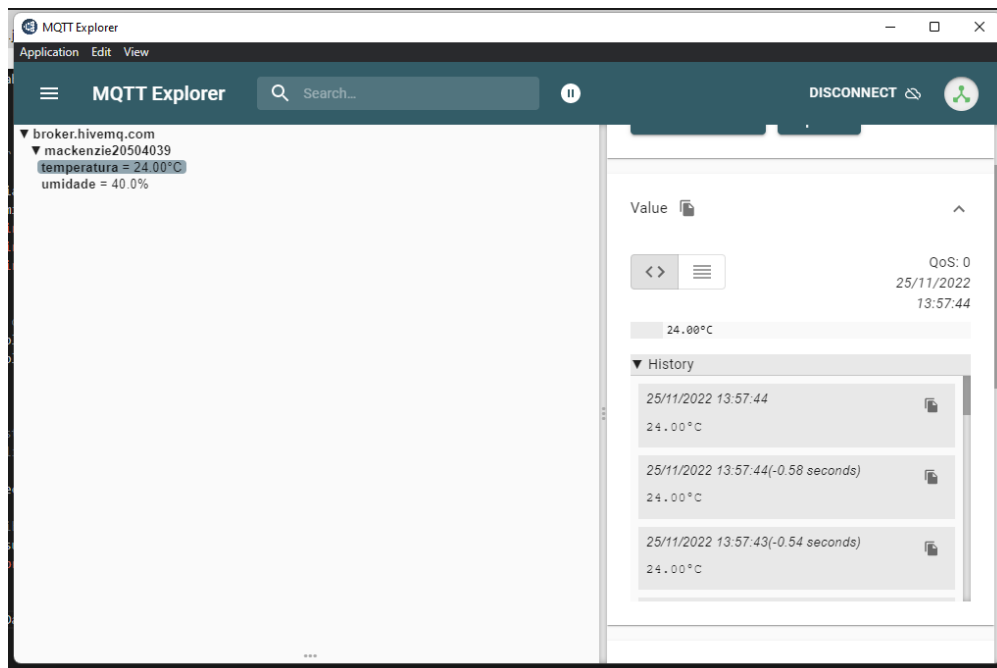


Figura 7: Visualização do Output MQTT Explorer

Agora precisar fazer a comunicação com o node-red, baixe o programa no link <https://nodered.org/> Importante! siga o passo a passo para instalação e execução do node-red.

Configuração do Node-red:

Primeiro vamos utilizar o node-red que recebe informações e configurar com nosso tópico exclusivo mackenzie20504039

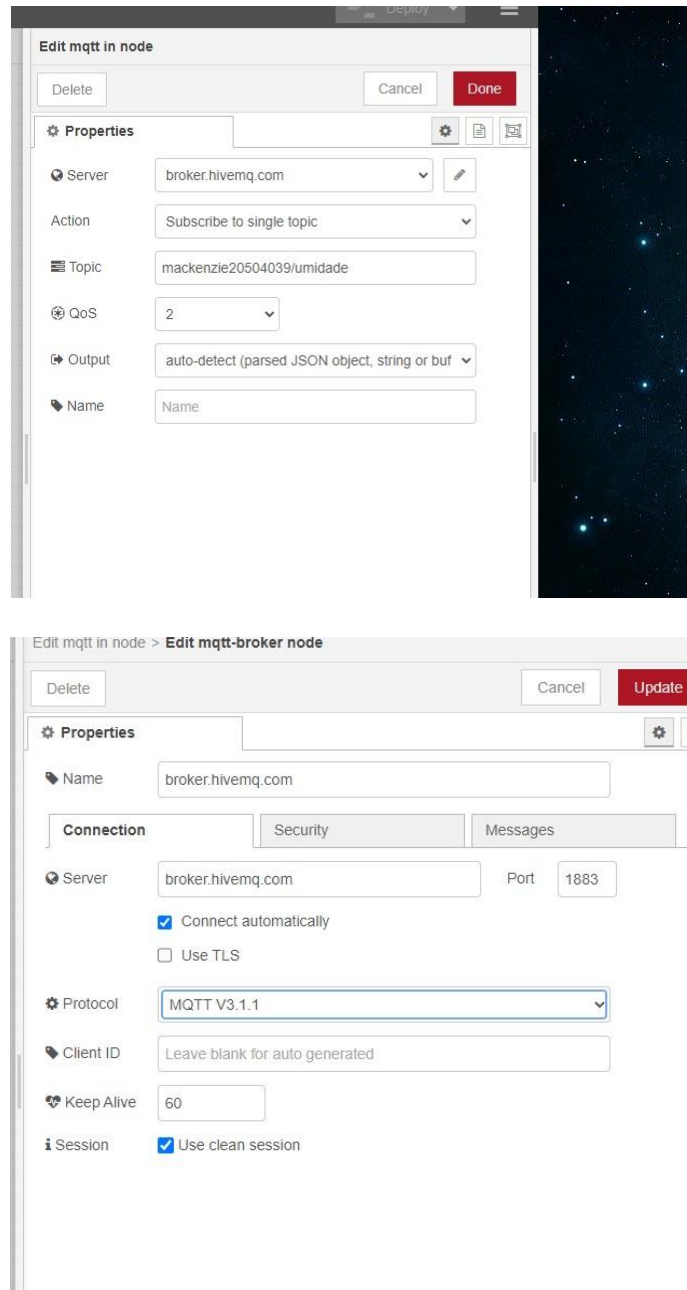


Figura 8 e 9: Configuração do Node Red

Menu > Manager Palette

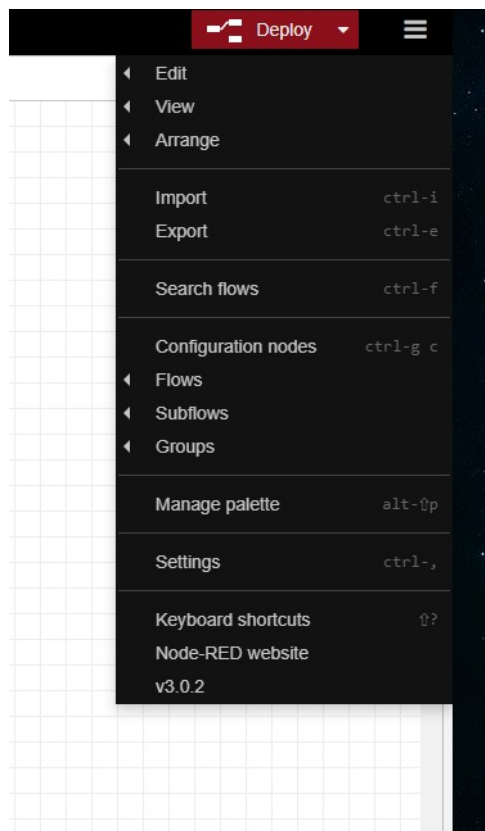


Figura 10: Print do Menu > Manage Palette

Depois vamos incluir a biblioteca do whatsapp para enviar mensagens

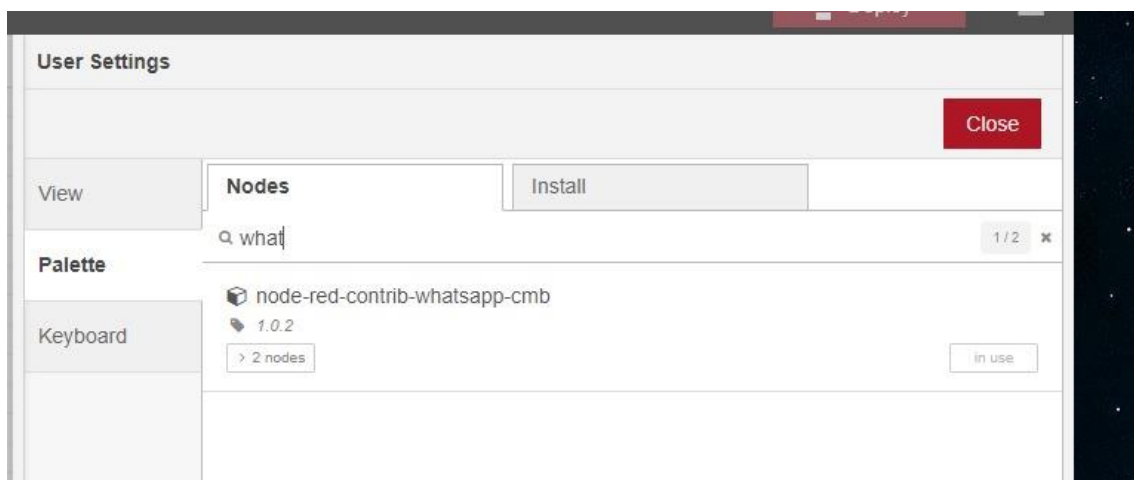


Figura 11: Inclusão da biblioteca do Whatsapp no Node Red

Siga o modelo, clique 2x no campo para abrir a configuração

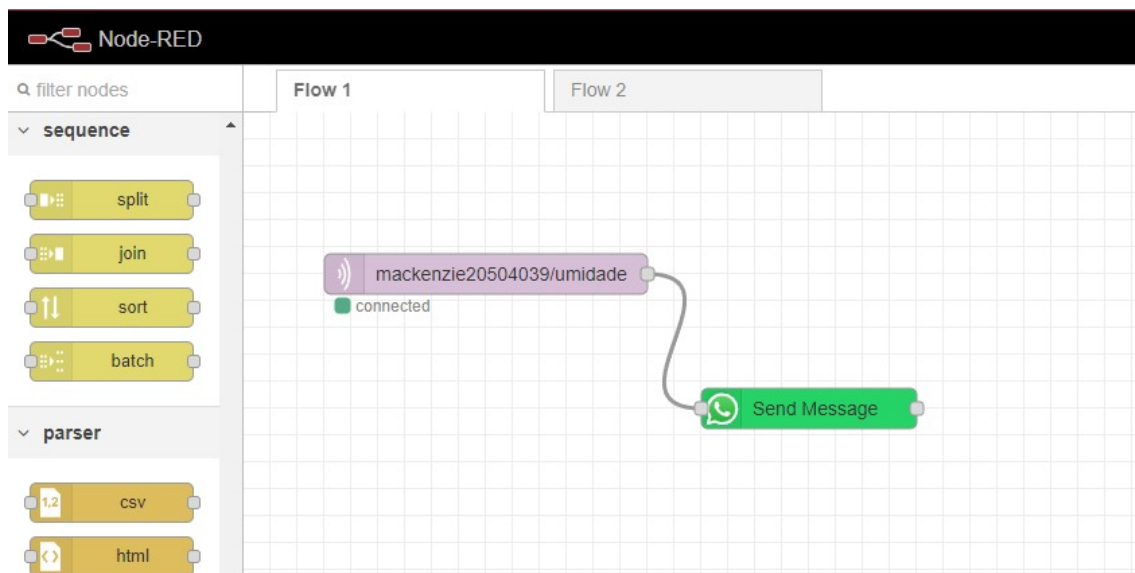


Figura 12: Workflow do Node-Red

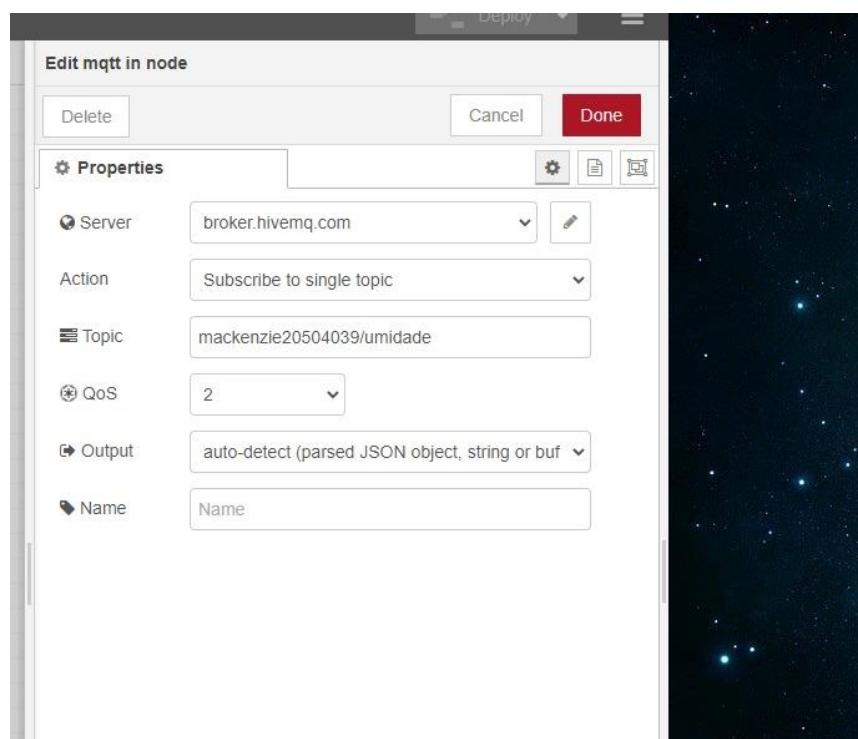


Figura 13: Configuração do Node Red II

The screenshot shows the 'Edit mqtt in node > Edit mqtt-broker node' configuration window in Node-RED. At the top, there are three buttons: 'Delete', 'Cancel', and 'Update'. Below these is a 'Properties' section with a gear icon and a document icon. The 'Name' field is set to 'broker.hivemq.com'. There are three tabs: 'Connection', 'Security', and 'Messages'. The 'Security' tab is active, showing a 'Username' field with the value '5511987358397' and an empty 'Password' field.

Figura 14: Configuração do Node Red III

Ao clicar 2x no campo de mensagem whats App tem a configuração, vamos usar agora o CallMeBot para enviar as mensagens, seguir os passos e depois preencher as informações como API-KEY na configuração do node-red

<https://www.callmebot.com/blog/free-api-whatsapp-messages/>

Depois de ter feito todas as etapas o projeto já está pronto para realizar os testes.

d. Fluxograma do que foi implementado:

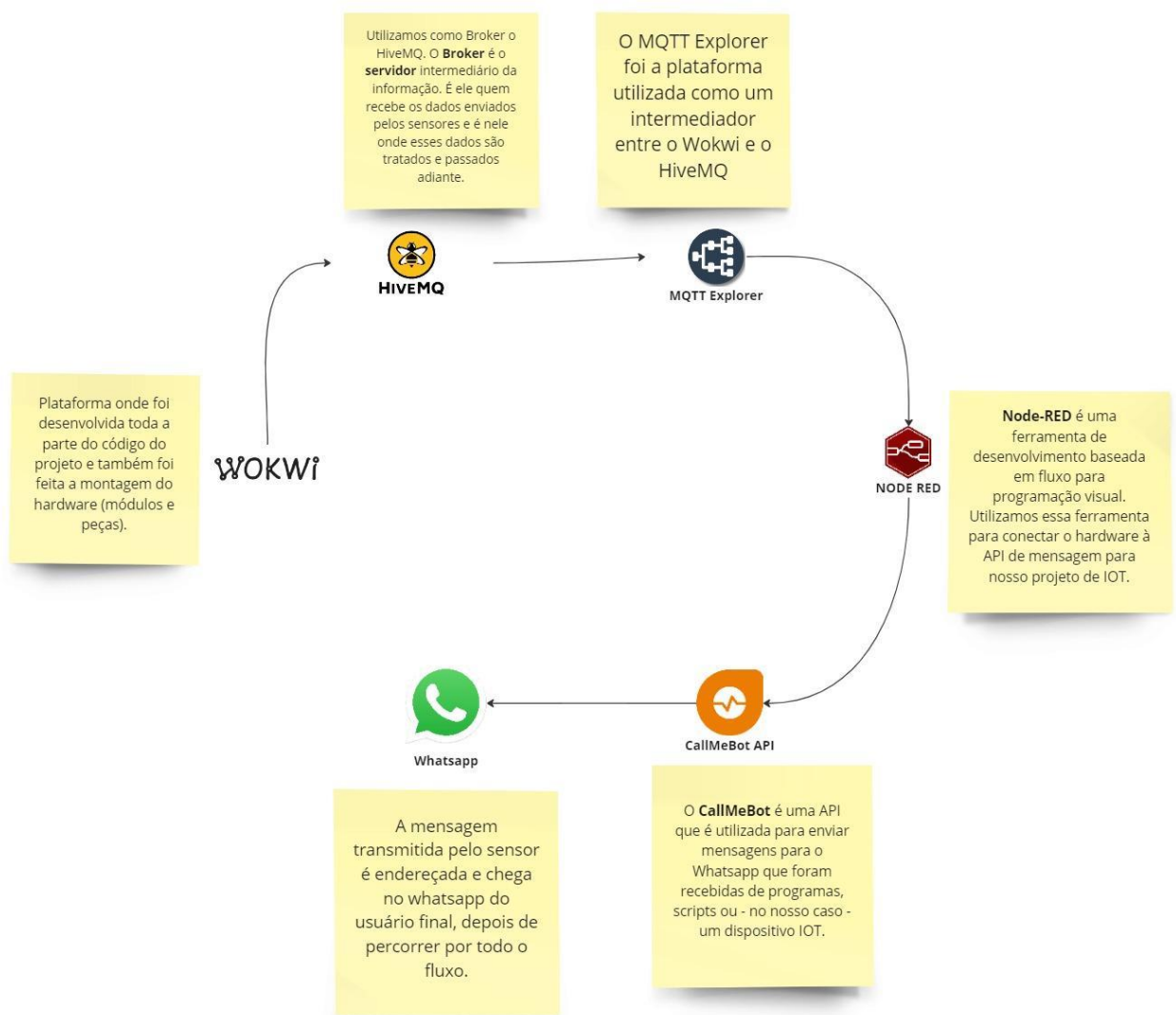


Figura 15: Fluxograma do funcionamento do projeto

3. RESULTADOS

O Projeto será desenvolvido pelo simulador Wokwi a conexão com a internet será feita pelo simulador MQTT, através do modulo wireless ESP32.

Wokwi - <https://wokwi.com/projects/349305302851519058>

a. Imagens do projeto e descrições do funcionamento:

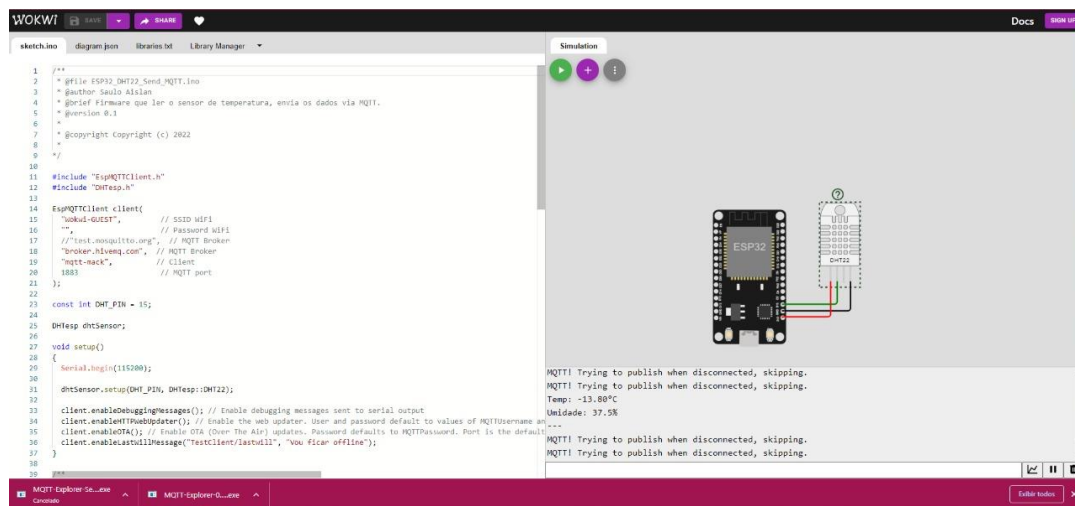


Figura 16: Protótipo em Execução

Essa imagem representa o protótipo do nosso projeto em execução. O console contém os dados obtidos através do sensor em funcionamento.

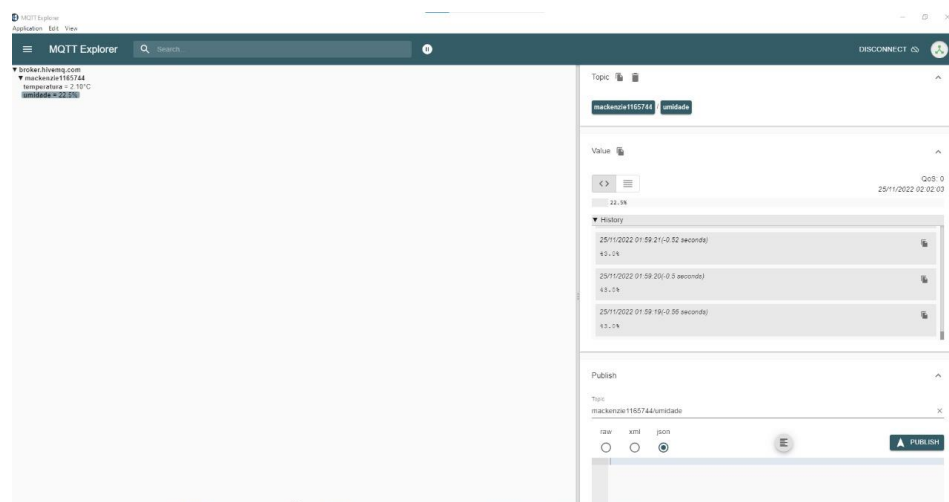


Figura 17: Frontpage do MQTT Explorer

Os dados são enviados para o MQTT Explorer e é gerado um histórico com o intervalo de tempo entre uma mensagem e a outra.

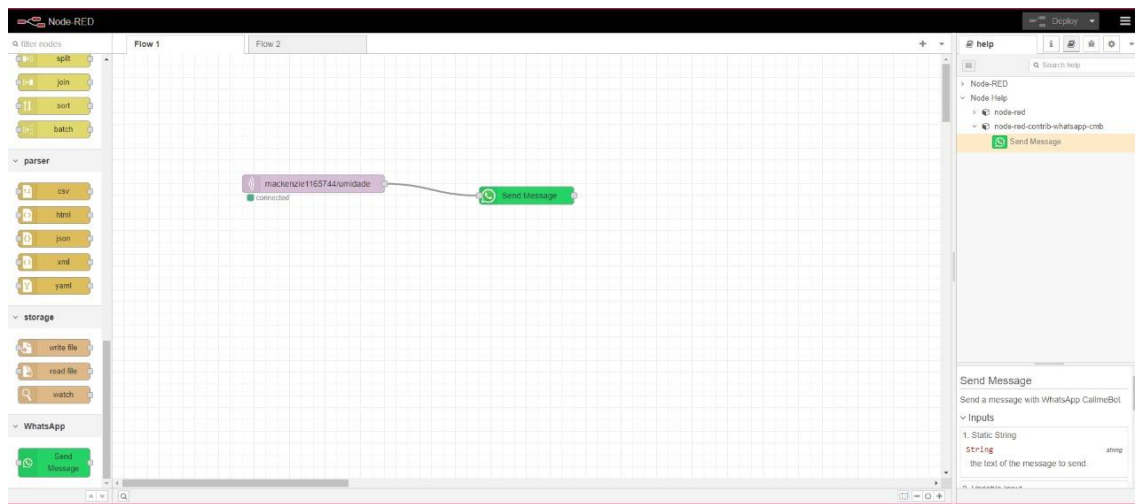


Figura 18: Workflow no Node-Red

Essa imagem representa o Node-Red, que recebe as informações do MQTT Explorer e com a extensão do Whatsapp ele consegue via CallMeBot API enviar mensagens com os dados recebidos através do sensor via Whatsapp.

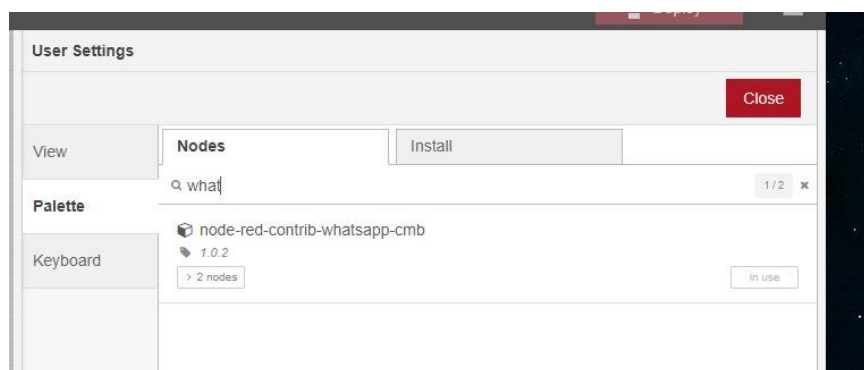


Figura 19: Extensão do Whatsapp Instalada

Extensão instalada do Whatsapp no Node-red.

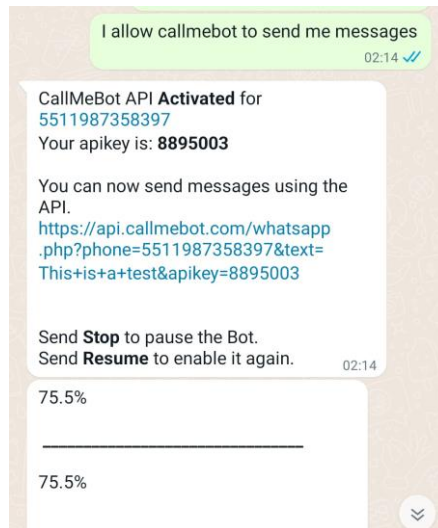


Figura 20: Credenciais para conexão da API Callmebot

APIKey necessária para configurar a extensão do whatsapp no Node-Red.

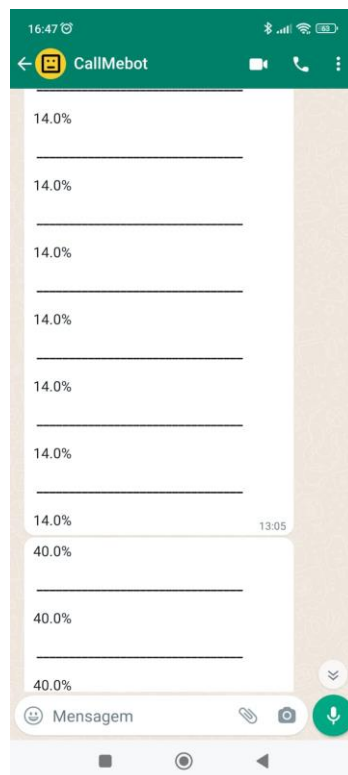


Figura 21: Mensagem recebida via Whatsapp

Mensagem recebida via Whatsapp.

b. Video-demonstração:

[Link do vídeo-demonstração](#)

c. Medição técnica do sensor e recebimento dos dados

Núm. medida	Sensor/atuador	Tempo de resposta
1	DHT22 - Umidade	16:55:05 (-0,6s)
2	DHT22 - Umidade	16:55:05 (-0,57s)
3	DHT22 - Umidade	16:55:06 (-0,67s)
4	DHT22 - Umidade	16:55:06 (-0,67s)
5	DHT22 - Umidade	16:55:07

Núm. medida	Sensor/atuador	Tempo de resposta
1	DHT22 - Temperatura	16:55:05 (-0,59s)
2	DHT22 - Temperatura	16:55:05 (-0,71s)
3	DHT22 - Temperatura	16:55:06 (-0,67s)
4	DHT22 - Temperatura	16:55:07 (-0,57s)
5	DHT22 - Temperatura	16:55:07

Figura 22: Testes do Sensor de Umidade e Temperatura

d. Repositório Github

Link de acesso: <https://github.com/milena-caravaggio/MackenzieArduino>

Dos itens solicitados, segue os caminhos para encontrar os arquivos:

I) Uma breve descrição do funcionamento e uso para quem quiser reproduzir.

Essa parte está presente no arquivo "README.md"

II) O software desenvolvido e a documentação de código.

O software desenvolvido está presente no "Mackenzie Milena e Gustavo.zip" e a documentação juntamente com o código estão no arquivo "CodArduino.txt"

III) A descrição do hardware utilizado (plataformas de desenvolvimento, sensores, atuadores, impressão 3D de peças, medidas de peças e caixas etc.)

Está presente neste documento, que também consta no Github.

IV) A documentação das interfaces, protocolos e módulos de comunicação.

Essa parte está presente no arquivo “README.md”

V) O projeto deve possuir comunicação/controlado via internet (TCP/IP) e uso do Protocolo MQTT.

O software desenvolvido está presente no “Mackenzie Milena e Gustavo.zip” e a documentação juntamente com o código estão no arquivo “CodArduino.txt” e contém a comunicação via protocolo MQTT.

e. Últimas considerações

Tivemos alguns problemas na conexão do projeto com o protocolo MQTT. Iniciamos o projeto utilizando uma ferramenta online chamada Tinkercad, porém no decorrer do projeto identificamos que a mesma não possuía suporte para alguns dos principais temas que deveriam compor o trabalho, então nós migramos para a plataforma Wokwi e refizemos o trabalho, pesquisando muito sobre as conexões MQTT, seu funcionamento e então adaptando e utilizando o MQTT Explorer juntamente com o HiveMQ.

4. CONCLUSÕES

Os objetivos propostos foram alcançados?

Concluimos que todos os objetivos propostos foram alcançados com sucesso. Criamos um protótipo útil e eficaz que cumpre o que promete e que serve como pontapé inicial para o desenvolvimento de vários outros projetos mais complexos e melhor estruturados, mas que já de imediato seria utilizado facilmente para auxiliar em diversas frentes. É um projeto simples, mas que cumpre todos os requisitos solicitados e que apresenta um funcionamento excelente, tanto em questão de hardware e código como também em conexão e funcionamento dos protocolos de conexão virtual e de comunicação/mensageria.

Quais são as vantagens e desvantagens do projeto?

Entendemos que a vantagem do projeto é ser um projeto simples, útil e prático, que qualquer pessoa com conhecimento moderado poderia desenvolver de forma autônoma e, por ser extremamente maleável/adaptativo, pode ser aplicado em qualquer frente que faça sentido, sem dúvidas nenhuma cumpriria facilmente seu papel que é reportar dados de temperatura e umidade. Já a desvantagem do projeto que identificamos é sua dependência de o usuário final estar conectado à rede para acessar as informações via Whatsapp, ou seja, se

o usuário estiver sem internet o mesmo não conseguirá ter acesso aos relatórios de mudança de temperatura e umidade transmitidos pelo dispositivo.

O que deveria/poderia ser feito para melhorar o projeto?

Poderíamos incluir outros componentes correlatos que aumentem ainda mais a versatilidade e a usabilidade do projeto, mas identificamos que para a proposta que sugerimos, o mesmo atende completamente.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Figura 1: Placa ESP32, Disponível em: <[Placa de Desenvolvimento DOIT ESP32 ESP-WROOM-32 WiFi Bluetooth - Saravati Materiais Técnicos](#)>, pesquisa: Placa ESP32. Acesso em: 20 de novembro de 2022

Figura 2: Sensor DHT22, Disponível em: <[Sensor Temperatura e Umidade DHT22 | Componentes Eletrônicos e Arduino \(curtocircuito.com.br\)](#)>, pesquisa: Sensor DHT. Acesso em: 20 de novembro de 2022

6. LISTA DE FIGURAS

Figura 1: Placa ESP32	6
Figura 2: Sensor DHT22	7
Figura 3: Configuração MQTT Explorer - Etapa I	8
Figura 4: Configuração MQTT Explorer - Etapa II	8
Figura 5: Configuração MQTT Explorer - Etapa III	9
Figura 6: Execução do Projeto Wokwi	9
Figura 7: Visualização do Output MQTT Explorer	10
Figura 8: Configuração do Node Red	11
Figura 9: Configuração do Node Red	11
Figura 10: Print do Menu > Manage Palette	12
Figura 11: Inclusão da biblioteca do Whatsapp no Node Red	12
Figura 12: Workflow do Node-Red	13
Figura 13: Configuração do Node Red II	13
Figura 14: Configuração do Node Red III	14
Figura 15: Fluxograma do funcionamento do projeto	15
Figura 16: Protótipo em Execução	16
Figura 17: Frontpage do MQTT Explorer	16
Figura 18: Workflow no Node-Red	17
Figura 19: Extensão do Whatsapp Instalada	17
Figura 20: Credenciais para conexão da API Callmebot	18
Figura 21: Mensagem recebida via Whatsapp	18
Figura 22: Testes do Sensor de Umidade e Temperatura	19