

INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DA PARAÍBA

CAMPUS: CAMPINA GRANDE

DISCIPLINA: IOT

CURSO: ENGENHARIA DE COMPUTAÇÃO

GRUPO: CINCO

ALUNOS: BRÁULIO MATHEUS

IALY SOUSA

JORGE VICTOR

MILENA AGUIAR

RAFAEL MUNIZ

PROJETO IOT 2023.2 - CONTROLADOR DE CLIMATIZAÇÃO

1 INTRODUÇÃO

Com o avanço da Internet das Coisas (IoT), as possibilidades de automação e otimização de sistemas tornaram-se uma realidade tangível. Nesse contexto, apresentamos um projeto inovador de IoT focado na gestão eficiente da climatização de ambientes, integrando um Node Sensor, um Node Atuador e um Gateway.

O principal objetivo desse projeto é proporcionar um ambiente climatizado de maneira inteligente e prática, adaptando-se dinamicamente às condições ambientais e às preferências do usuário. Para alcançar esse objetivo, são utilizados três componentes fundamentais: o Node Sensor, responsável por coletar dados ambientais cruciais; o Node Atuador, encarregado de efetuar ajustes no sistema de climatização; e o Gateway, que atua como o cérebro central, coordenando a comunicação entre os nós e possibilitando a tomada de decisões baseadas nos dados coletados.

O Node Sensor é equipado com um módulo sensor DHT22, capaz de medir a temperatura. Esse dado é essencial para compreender o ambiente circundante e tomar decisões informadas sobre o controle da climatização. O Node Atuador, por sua vez, é responsável por executar as ações necessárias para ajustar o sistema de climatização.

O papel crucial do Gateway reside na integração e processamento eficiente dos dados provenientes dos Nodes Sensor e Atuador. Ele se comunica bidirecionalmente com esses nós, garantindo uma troca contínua de informações. Além disso, o Gateway pode ser conectado à Internet, permitindo o acesso remoto e a monitorização do sistema de climatização, bem como a implementação de atualizações de software.

Ao unir esses três componentes em um sistema coeso, nosso projeto busca não apenas aperfeiçoar o conforto térmico, mas também reduzir o consumo de energia. A implementação dessa solução de IoT representa um passo significativo na direção de ambientes mais inteligentes, eficientes e adaptáveis às necessidades dos usuários.

1.1 Objetivo Geral

Desenvolver um sistema integrado de Internet das Coisas (IoT) para controle inteligente de climatização em ambientes, empregando Nodes Sensor e Atuador, um Gateway e uma Aplicação. O objetivo é criar uma solução eficaz que, por meio da coleta precisa de dados ambientais, permita ajustes dinâmicos no sistema de climatização, proporcionando conforto térmico, otimização energética e possibilitando a interação remota por meio de uma interface intuitiva.

1.2 Tecnologias Utilizadas

- **Arduíno:** É uma plataforma de hardware e software responsável por receber e ler os dados transmitidos pelo sensor e transmitir dados para o infravermelho;
- **Sensor DHT22:** é um sensor capaz de medir a temperatura do ambiente;
- **Bluetooth:** Rede de comunicação sem fio que realiza a transferência de dados e a conexão dos dispositivos.
- **LED Infravermelho:** transmite informações para o ar condicionado;
- **MQTT:** Protocolo responsável por realizar a troca de mensagens;
- **BeagleBone Black (BBB):** Dispositivo principal que realiza a intercomunicação entre os Nodes e a rede funcionando assim como um Gateway.
- **Node.JS:** é um ambiente de execução para JavaScript do lado do servidor. Ele utiliza o motor V8 do Google Chrome para executar o código JavaScript.
- **NPM (Node Package Manager):** O NPM é o sistema de gerenciamento de pacotes para o Node.js.
- **ESP32:** é um microcontrolador de baixo custo e de alto desempenho que integra Wi-Fi e Bluetooth em um único chip.
- **Receptor Infravermelho:** é um dispositivo eletrônico que capta sinais infravermelhos transmitidos por um emissor infravermelho.

2 ARQUITETURA

Node Sensor: (ESP32 com Módulo DHT22 e Conectividade Bluetooth usando MQTT): Realiza leitura dos dados do sensor, encapsula esses dados em mensagens MQTT e os envia para o Gateway através do módulo Bluetooth.

Node Atuador (Arduino Mega com Módulo LED Infravermelho, receptor infravermelho, módulo Bluetooth HC-05, usando MQTT): Recebe comandos MQTT via Bluetooth, interpreta sinais infravermelhos do ar condicionado, e controla dispositivos infravermelhos, como o ar condicionado.

Gateway (BeagleBone Black com Conectividade Bluetooth e Protocolo MQTT): Recebe dados MQTT dos Nodes Sensores, toma decisões de controle, envia comandos via MQTT para os Nodes Atuadores, realiza o envio dos dados para a Aplicação e armazena informações no banco de dados.

Aplicação: Realiza testes para garantir que o Gateway esteja configurado para encaminhar solicitações da Aplicação para a API Spring Boot.

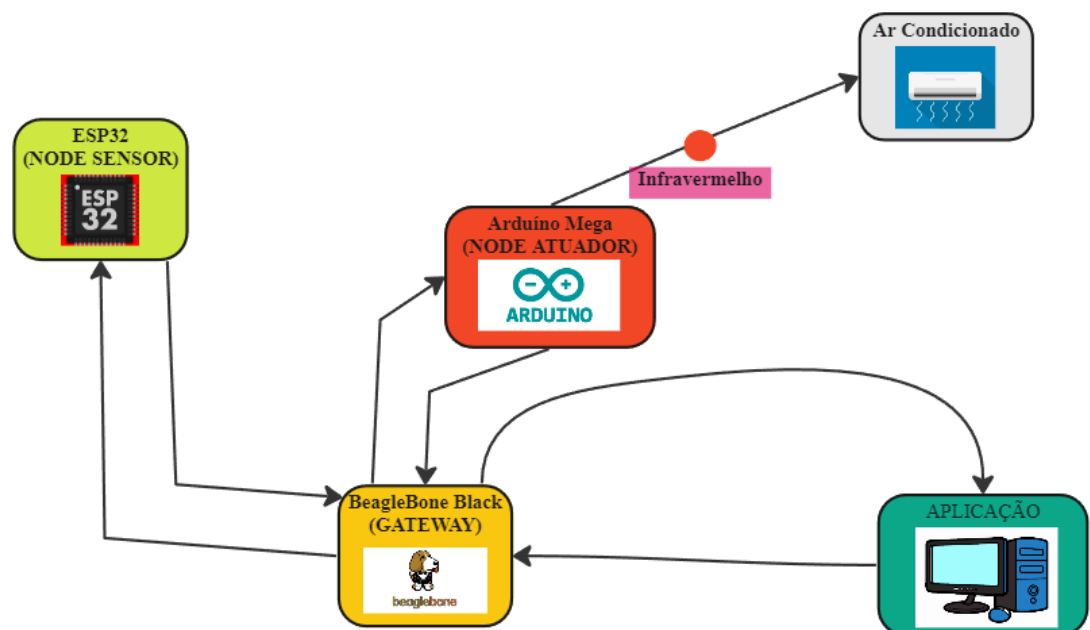


Figura 1 - Arquitetura Geral do Projeto

Essa arquitetura visa criar um ecossistema coeso, permitindo a coleta precisa de dados ambientais, ajustes automáticos no sistema de climatização e interação remota via Aplicação, proporcionando um controle eficiente e adaptável às condições climáticas.

Todas as conexões entre o Gateway e os Nodes ocorrem por meio da conexão Bluetooth, já a conexão entre o Gateway e a Aplicação ocorre por meio da conexão de rede Wi-Fi.

O GITHUB deste projeto se encontra no Link: <https://github.com/RafaelMunizz/IoT-2023.2>

2.1 Node Sensor (ESP32 com Módulo DHT22 e Conectividade Bluetooth usando MQTT)

A arquitetura de um Node Sensor com um ESP32, Módulo DHT22, conectividade Bluetooth e MQTT na plataforma Arduino envolve a integração de hardware e software para coleta de dados, comunicação e interação com outros dispositivos.

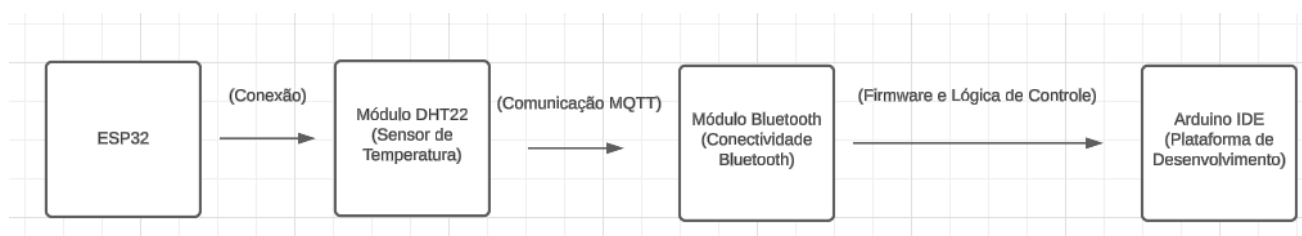


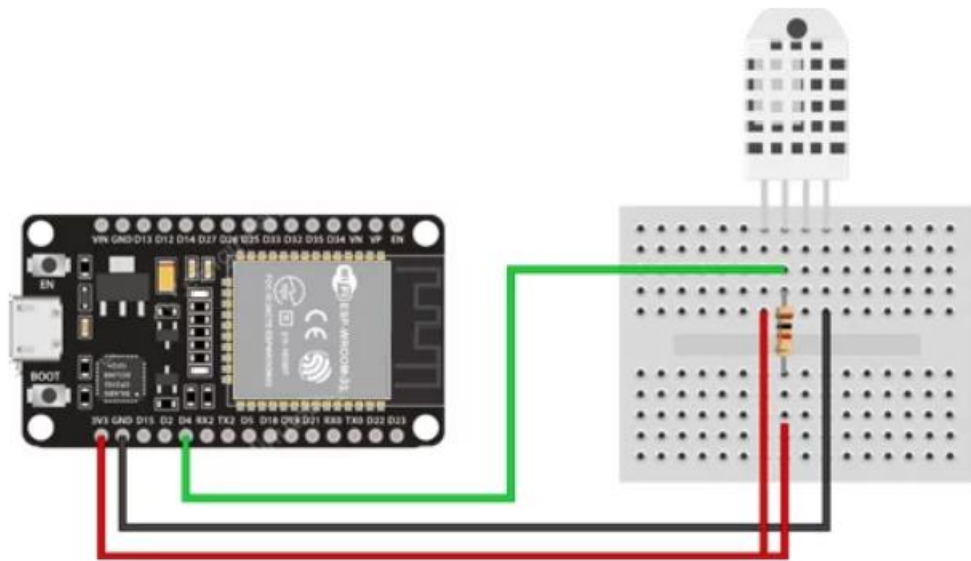
Figura 2 - Diagrama de Bloco (Nó Sensor)

Aqui estão as principais componentes da arquitetura do Node Sensor:

2.1.1 Hardware

- ESP32: O ESP32 atua como a unidade principal de processamento e comunicação. Ele é programado usando a plataforma Arduino IDE;

- Módulo DHT22: Este é o sensor de temperatura e umidade que está conectado ao ESP32 para coletar dados ambientais;
- Módulo Bluetooth: Para a conectividade Bluetooth, um módulo BLE (Bluetooth Low Energy) pode ser utilizado. O ESP32 já possui suporte integrado para BLE.



Ligação do sensor de umidade e temperatura DHT22 com ESP32

2.1.2 Software

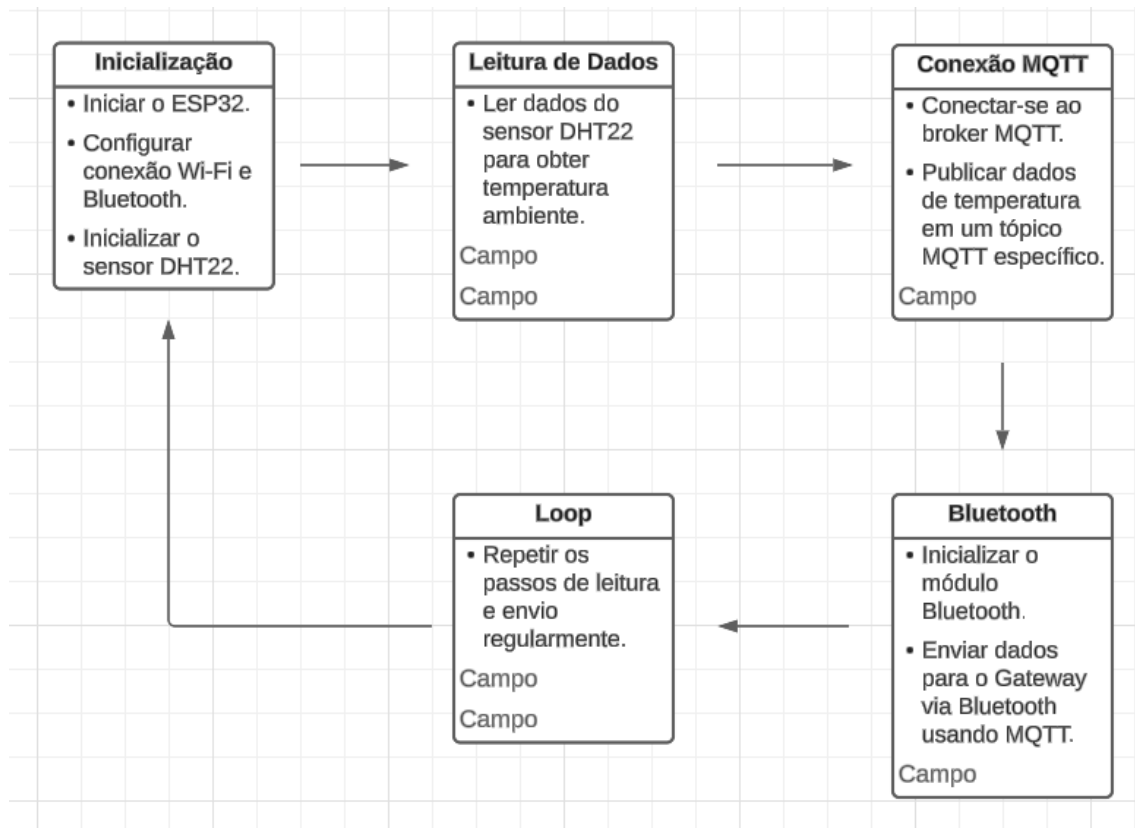


Figura 3 - Fluxograma de Softwares do node Sensor

2.1.1.1 Firmware

O firmware é o código que é carregado no ESP32 e é responsável por:

- Inicializar o ESP32, configurar a conexão Wi-Fi e Bluetooth;
- Ler dados do sensor DHT22 para medir a temperatura;
- Estabelecer conexões MQTT para publicar dados;
- Lidar com eventos Bluetooth, se aplicável.

2.1.1.2 Bibliotecas Arduino

Utilização de bibliotecas específicas do Arduino, como:

- DHT: Para interagir com o sensor de temperatura e umidade DHT22.
- PubSubClient: Para implementar a comunicação MQTT.
- BLEPeripheral ou BLE: Para a implementação da conectividade Bluetooth Low Energy.

2.1.1.3 Comunicação MQTT

O Node Sensor utiliza o protocolo MQTT para enviar dados para um servidor MQTT (broker). O firmware inclui lógica para:

- Conectar-se ao broker MQTT.
- Publicar dados do sensor em tópicos MQTT específicos.
- Pode ser necessário configurar a autenticação e autorização no broker MQTT para garantir a segurança da comunicação.

2.1.1.4 Conectividade Bluetooth

Caso a comunicação local via Bluetooth seja necessária, o firmware deve lidar com a inicialização do BLE e implementar a lógica para interações Bluetooth, como a exposição de serviços e características GATT.

2.1.1.5 Arduino IDE

A plataforma Arduino IDE é utilizada para desenvolver e carregar o firmware no ESP32. Os desenvolvedores podem aproveitar a simplicidade da IDE Arduino e a vasta biblioteca disponível.

2.1.3 Funcionamento

O funcionamento do Node Sensor ocorre por meio dos seus componentes da seguinte forma:

- O sensor DHT22 faz a leitura dos dados de temperatura do ambiente;
- O STM32 lê os dados do sensor DHT22;
- Os dados são encapsulados em mensagens MQTT;
- Através do módulo Bluetooth, os dados são enviados para o Gateway usando o protocolo MQTT.

2.2 Node Atuador (Arduino com Módulo LED Infravermelho, Receptor Infravermelho e Conectividade Bluetooth usando MQTT):

O Node Atuador baseia-se em um Arduino Mega com módulo LED infravermelho, receptor infravermelho, módulo Bluetooth HC-05 e conectividade MQTT, envolve a integração de hardware e software para permitir que o Node Atuador receba comandos remotamente e realize ações específicas.

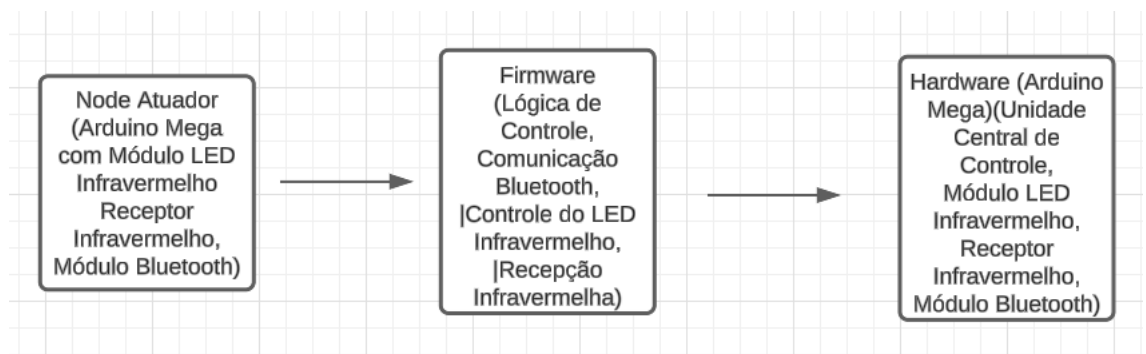


Figura 4 - Diagrama de Bloco (Nó Atuador)

Aqui estão os principais componentes dessa arquitetura do Node Atuador:

2.2.1 Hardware

- Arduino Mega: Atua como a unidade central de processamento e controle. O Arduino Mega é escolhido por sua capacidade de E/S (entrada/saída) e recursos de memória;
- Módulo LED Infravermelho: Este módulo é usado para emitir sinais infravermelhos, permitindo a comunicação remota;

- Receptor Infravermelho: Permite ao Node Atuador receber sinais infravermelhos enviados por um dispositivo de envio IR;
- Módulo Bluetooth HC-05: Oferece conectividade Bluetooth para comunicação sem fio.

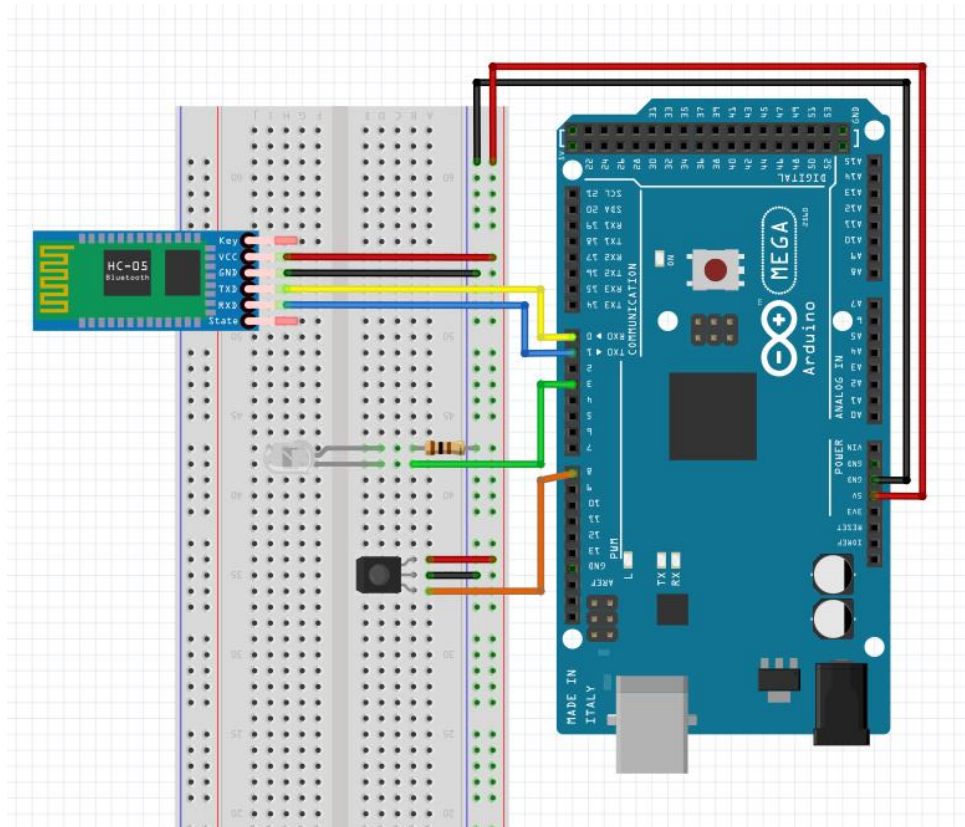


Figura 5 - Esquema de Montagem do Node Atuador

2.2.2 Software

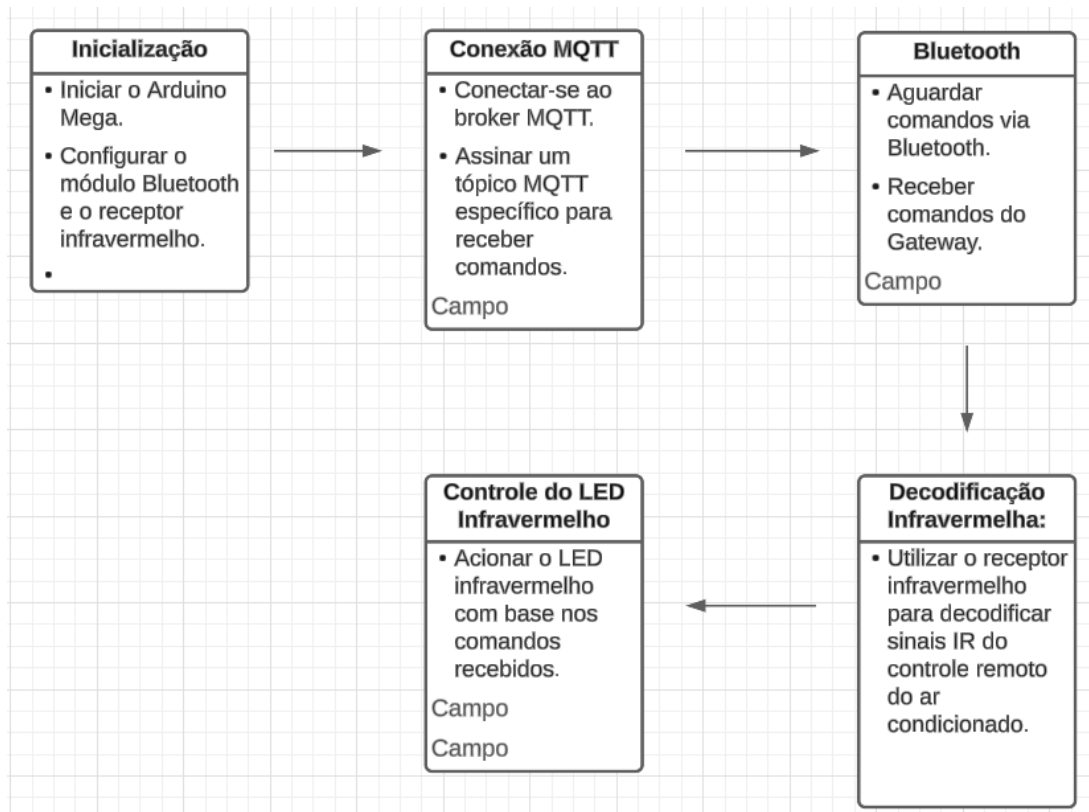


Figura 6 - Fluxograma de Funcionamento de Softwares para o Atuador

2.2.2.1 Firmware:

O firmware é o código que é carregado no Arduino Mega e é responsável por:

- Inicializar o Arduino Mega, configurar as portas necessárias e definir a lógica de controle;
- Monitorar a comunicação Bluetooth HC-05 para receber comandos;
- Processar comandos recebidos via MQTT;
- Controlar o módulo LED infravermelho com base nos comandos recebidos.

2.2.2.2 Bibliotecas Arduino:

Utilização de bibliotecas específicas do Arduino, como:

- **IRremote:** Para interagir com o receptor infravermelho;
- **SoftwareSerial:** Para comunicação com o módulo Bluetooth HC-05;
- **PubSubClient:** Para implementar a comunicação MQTT.

2.2.2.3 Comunicação MQTT

O Node Atuador utiliza o protocolo MQTT para receber comandos. A lógica do firmware inclui:

- Conectar-se ao broker MQTT;
- Assinar um tópico MQTT específico para receber comandos.

2.2.2.4 Comunicação Bluetooth

O módulo Bluetooth HC-05 é configurado para permitir a comunicação com outro dispositivo Bluetooth, como o ESP32 ou o BeagleBone Black (Gateway). O firmware lida com a recepção de comandos via Bluetooth.

2.2.2.5 Controle do LED Infravermelho

A lógica do firmware deve incluir a capacidade de controlar o módulo LED infravermelho. Dependendo dos comandos recebidos via MQTT ou Bluetooth, o Node Atuador aciona o LED infravermelho para realizar ações específicas.

2.2.2.6 Receptor Infravermelho

O receptor infravermelho (IR) é um dispositivo eletrônico capaz de receber sinais infravermelhos transmitidos por um emissor infravermelho, neste caso o ar condicionado. Neste projeto o receptor é utilizado na decodificação do sinal infravermelho do ar condicionado, a partir da leitura deste sinal, é feita a decodificação para que o LED infravermelho envie os comandos corretos para o ar condicionado.

2.2.3 Funcionamento

O funcionamento do Node Atuador ocorre por meio dos seus componentes da seguinte forma:

- O Arduino recebe comandos MQTT através do módulo Bluetooth;
- Receptor infravermelho recebe sinais do aparelho de ar condicionado para decodificá-los e gerar os códigos corretos por meio do LED Infravermelho;
- Dependendo dos comandos recebidos, o LED Infravermelho é acionado para controlar dispositivos IR (por exemplo, ligar/desligar um ar-condicionado).

2.3 Gateway (BeagleBone Black com Conectividade Bluetooth e Protocolo MQTT)

O Gateway envolve a integração de hardware e software para facilitar a comunicação entre dispositivos IoT, como Nodes Sensores e Nodes Atuadores, e sistemas externos.

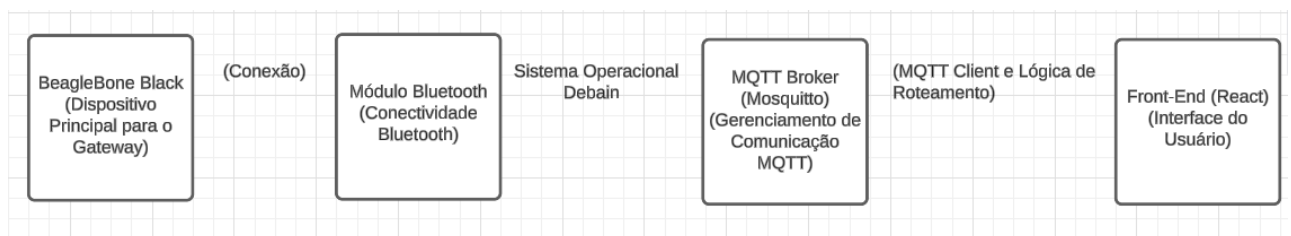


Figura 7 - Diagrama de Blocos Gateway

Aqui estão os principais componentes do Gateway:

2.3.1 Hardware

BeagleBone Black: Atua como o principal hardware para o Gateway. A BeagleBone Black é uma placa de desenvolvimento de código aberto que oferece conectividade e recursos de E/S.

2.3.2 Software

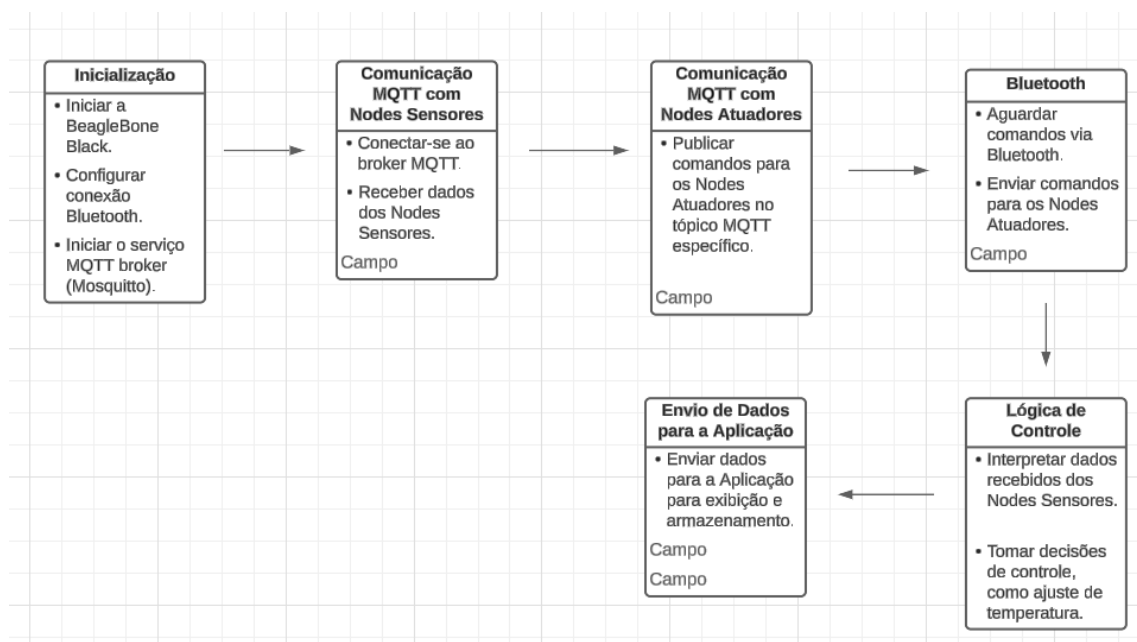


Figura 8 - Fluxograma de Software do Gateway

2.3.2.1 Sistema Operacional Debian

Os passos para instalar o Sistema Operacional Debian no BeagleBone Black foram os seguintes:

A escolha da Imagem Debian:

Escolha uma imagem do Debian adequada para a BeagleBone Black. Foi utilizada a versão AM335x 11.7 2023-09-02 4GB eMMC IoT Flasher.

Preparação do Cartão microSD:

Foi formatado um cartão microSD em FAT32. A ferramenta [SD Card Formatter](#) foi utilizada no processo.

Gravação da Imagem Debian no Cartão microSD:

Utilizou-se a ferramenta [Etcher](#) para gravar a imagem Debian no cartão microSD. No Etcher, foi selecionada a imagem Debian, o dispositivo microSD e iniciar o processo de gravação.

Conectar a BeagleBone Black ao Computador:

Foi conectada a BeagleBone Black ao computador usando um cabo microUSB. Isso permitiu o acesso a BeagleBone Black via SSH.

Acessar a BeagleBone Black via SSH:

Assim que o cartão microSD estiver pronto, foi conectada a BeagleBone Black à alimentação e ao computador. Aguarde até que a placa inicialize e, em seguida, acesse-a via SSH. O endereço padrão de conexão é o de IP 192.168.7.2.

2.3.2.2 Bluetooth

A BeagleBone Black deve estar equipada com módulos Bluetooth ou adaptadores para permitir a comunicação Bluetooth. O suporte ao Bluetooth no Debian é crucial para interagir com dispositivos que utilizam essa tecnologia, como Nodes Sensores ou Atuadores.

2.3.2.3 Middleware - MQTT Broker

Um servidor MQTT (broker) é instalado no Gateway para gerenciar a comunicação MQTT entre os dispositivos IoT. Existem várias opções de brokers MQTT disponíveis para Debian, neste caso utilizamos o Mosquitto.

Os passos para a instalação e atualização do Mosquitto MQTT são os seguintes:

```
# Atualização dos Repositórios
sudo apt update

# Instalação do Mosquitto
sudo apt install mosquitto

# Instalação do Cliente de Linha de Comando (Opcional)
sudo apt install mosquitto-clients

# Iniciar o Serviço Mosquitto
sudo systemctl start mosquitto
```

```
# Garantir que o Mosquitto seja iniciado automaticamente na
inicialização
sudo systemctl enable mosquitto

# Teste do Serviço Mosquitto
mosquitto_pub -h localhost -t test_topic -m "Hello, MQTT!"
mosquitto_sub -h localhost -t test_topic
```

Todos os passos descritos devem ser executados com a permissão de administrador.

2.3.2.4 MQTT Client

O Gateway também atua como um cliente MQTT para se comunicar com Nodes Sensores e Atuadores. Ele é responsável por assinar tópicos relevantes para receber dados dos Nodes Sensores e publicar comandos para os Nodes Atuadores.

2.3.2.5 Lógica de Roteamento e Processamento

A lógica no Gateway inclui a capacidade de rotear dados entre os dispositivos IoT e processar mensagens MQTT. Ele deve ser capaz de entender os tópicos MQTT, interpretar dados e encaminhar informações conforme necessário.

2.3.3 Funcionamento

- A BBB recebe dados MQTT dos Nodes sensores;
- Baseando-se nos dados recebidos, o BBB toma decisões de controle (por exemplo, ajuste de temperatura);
- Os comandos resultantes são enviados via MQTT para os Nodes atuadores através da conexão Bluetooth;
- A BBB também realiza o envio dos dados para a Aplicação e esta os guarda no banco de dados.

2.4 Aplicação

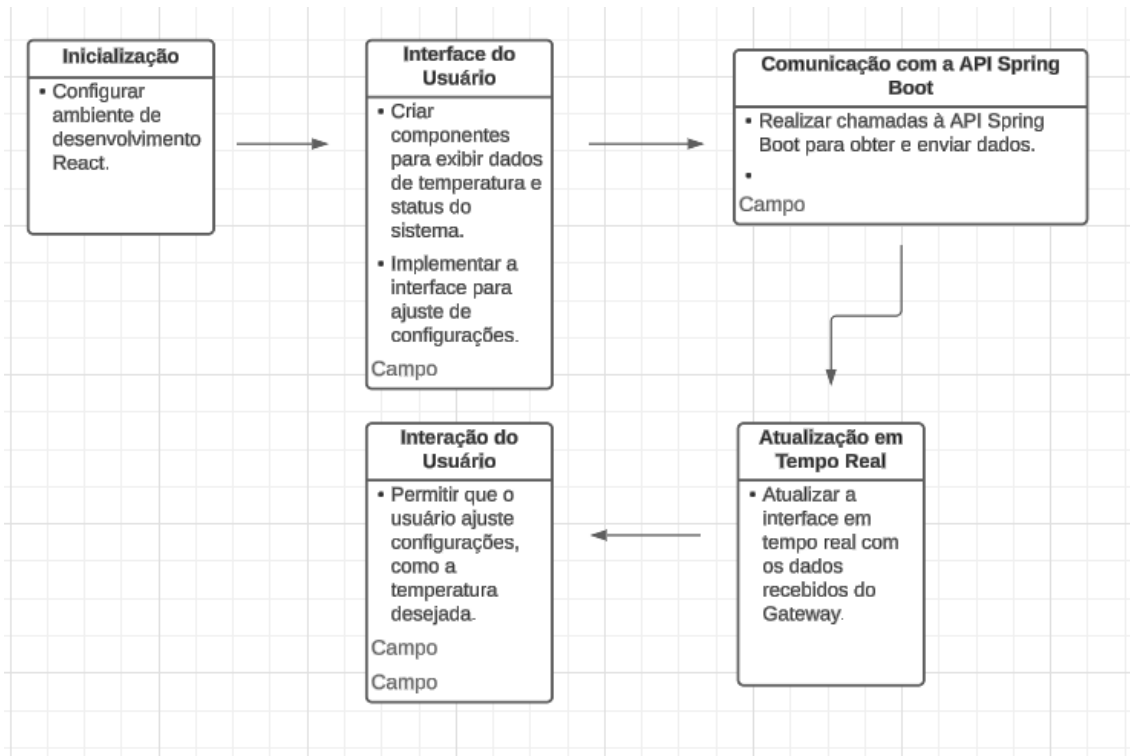


Figura 9 - Fluxograma de Funcionamento da Aplicação

2.4.1 Configuração do ambiente de desenvolvimento

Foi feita a instalação das tecnologias Node.js e npm.

2.4.2 Criação do Projeto React

Criou-se um projeto React usando:

```
npx create-react-app meu-front-end.
```

2.4.3 Desenvolvimento da Aplicação em React

Desenvolveram-se componentes React para a interface do usuário, utilizando ferramentas como React Router e Axios para navegação e requisições HTTP.

2.4.4 Conexão com a API Spring Boot

Configurou-se chamadas à API Spring Boot na Aplicação, lidando com CORS adequadamente. Implemente operações CRUD para interagir com o banco de dados por meio da API.

2.4.5 Integração com o Banco de Dados:

Implementou-se operações CRUD (Create, Read, Update, Delete) na aplicação para interagir com a API Spring Boot e, conseqüentemente, o banco de dados.

2.4.6 Configuração do Gateway

Foram feitos testes para que o gateway esteja configurado para encaminhar solicitações da aplicação para a API Spring Boot.

3 DISCUSSÕES E REFLEXÕES FINAIS

3.1 Desafios Enfrentados

Durante a implementação deste projeto de controle climático utilizando uma arquitetura IoT, enfrentamos diversos desafios que impactaram diretamente o desempenho e a eficácia do sistema. Um dos principais desafios foi a inconsistência na comunicação entre os Nodes (Sensor e Atuador) e o Gateway. A desconexão intermitente prejudicou a coleta precisa de dados de temperatura pelo Node Sensor, resultando em leituras imprecisas e, por vezes, na falta de resposta do Node Atuador aos comandos do Gateway.

Outro desafio notável foi a complexidade da integração entre as diferentes tecnologias, como MQTT, Bluetooth e a interação com o hardware específico de cada Node. As nuances na configuração e comunicação dessas tecnologias contribuíram para a instabilidade da rede, afetando diretamente a confiabilidade do sistema como um todo.

A falta de uma camada robusta de segurança também foi identificada como um desafio significativo. A ausência de autenticação adequada na comunicação MQTT pode comprometer a integridade e a confidencialidade dos dados, sendo uma área crítica que precisa ser aprimorada para garantir a segurança do sistema em ambientes do mundo real.

3.2 Sugestões para Melhorias Futuras

Para mitigar os desafios identificados, propomos diversas sugestões para melhorias futuras, visando aprimorar a confiabilidade, segurança e eficiência do sistema:

3.2.1 Melhoria na Comunicação e Estabilidade da Rede:

Explorar algoritmos de reconexão automática para Nodes, garantindo uma comunicação mais estável.

Implementar protocolos de handshaking para verificar a disponibilidade e a integridade da conexão entre Nodes e Gateway.

3.2.2 Reforço na Segurança da Comunicação:

Introduzir autenticação e criptografia na comunicação MQTT para proteger a integridade e confidencialidade dos dados transmitidos.

Implementar medidas de segurança adicionais, como firewalls e filtros de pacotes, para proteger contra possíveis ameaças externas.

3.2.3 Aprimoramento na Interface do Usuário (Front-End):

Desenvolver uma interface de usuário mais intuitiva e informativa, permitindo uma compreensão clara das condições climáticas monitoradas.

Incluir alertas visuais ou notificações para indicar a desconexão entre os Nodes e o Gateway.

3.2.4 Testes e Simulações Extensivas:

Realizar testes extensivos em ambientes simulados para avaliar a robustez do sistema em condições diversas.

Simular cenários de estresse para identificar potenciais pontos fracos na arquitetura e implementação.

3.2.5 Investigação de Tecnologias Alternativas:

Explorar tecnologias alternativas de comunicação e protocolos mais eficientes para otimizar a troca de dados entre os dispositivos IoT.

Avaliar a viabilidade de implementação de tecnologias de comunicação de baixa potência para otimizar o consumo de energia.

Essas sugestões buscam endereçar as lacunas identificadas, garantindo um sistema mais confiável, seguro e eficiente. O contínuo aprimoramento e a atenção a esses pontos críticos são essenciais para a implementação bem-sucedida de soluções de IoT em ambientes do mundo real.

4 CONCLUSÃO

Na conclusão deste projeto, é crucial ressaltar que, apesar dos avanços significativos na integração de um Node Sensor, Node Atuador, Gateway e Aplicação para o controle climático, enfrentamos desafios relacionados à falha de funcionamento causada pela ausência de conexão entre os componentes e o Gateway.

A captura dos valores de temperatura pelo Node Sensor representa um ponto central para o sucesso deste projeto, proporcionando dados cruciais para a tomada de decisões e otimização do ambiente climático. No entanto, a desconexão entre o Node Sensor, Node Atuador e o Gateway comprometeu a eficácia dessa captação, resultando em uma limitação significativa no monitoramento em tempo real.

Além disso, a conexão com o ar condicionado, realizada pelo Node Atuador, é uma parte vital da funcionalidade do sistema. A falta de comunicação eficiente entre o Node Atuador e o Gateway impede a implementação de ajustes automáticos no ar condicionado com base nas leituras do Node Sensor, comprometendo a capacidade do sistema de resposta dinâmica às condições climáticas.

Esses desafios ressaltam a importância crítica de uma comunicação robusta entre os dispositivos para garantir a efetividade do sistema como um todo. Para futuras iterações, será essencial direcionar esforços para fortalecer a integração e a troca de dados entre os componentes, assegurando que a captação de valores de temperatura e a conexão com o ar condicionado sejam confiáveis e contínuas.

Adicionalmente, ao considerar as melhorias futuras, a capacidade de aprimorar a interação entre a Aplicação e os demais componentes do sistema é crucial. Melhorias na interface do usuário devem proporcionar uma compreensão mais clara das condições climáticas e facilitar o controle eficaz do ar condicionado, mesmo em situações de conectividade desafiadoras.

Assim, ao alinhar a captação precisa dos valores de temperatura com uma conexão robusta entre os dispositivos, este projeto se tornará não apenas uma solução eficaz de controle climático, mas também uma base sólida para inovações futuras e aprimoramentos contínuos na experiência do usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

ESP32. Espressif Systems. Disponível em:

<https://www.espressif.com/en/products/socs/esp32>. Acesso em: [10/12/2023].

Receptor Infravermelho. Vishay. Disponível em: <https://www.vishay.com/ir-receiver-modules/>. Acesso em: [10/12/2023].

Node.js. Node.js. Disponível em: <https://nodejs.org/>. Acesso em: [10/12/2023].

NPM (Node Package Manager). npm. Disponível em: <https://www.npmjs.com/>. Acesso em: [10/12/2023].

Mosquitto MQTT. Eclipse Mosquitto. Disponível em: <https://mosquitto.org/>. Acesso em: [10/12/2023].

Arduino. Arduino. Disponível em: <https://www.arduino.cc/>. Acesso em: [10/12/2023].

React. React. Disponível em: <https://reactjs.org/>. Acesso em: [10/12/2023].

Spring Boot. Spring Boot. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: [10/12/2023].

BeagleBone Black. BeagleBoard.org. Disponível em: <https://beagleboard.org/black>. Acesso em: [10/12/2023].

DHT22 Sensor. Adafruit. Disponível em: <https://www.adafruit.com/product/385>. Acesso em: [10/12/2023].