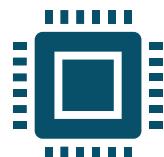
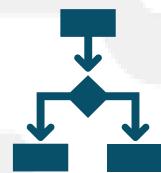


Roberto Higino Pereira da Silva

Joésia Moreira Julião Pacheco

Joelma Monteiro de Carvalho

**Organizadores**



## **Curso de Capacitação Profissional Técnica**

**Hardware e Firmware IoT com IMPC HTNB32L**

**Módulo 4: Organização dos Projetos Finais do Curso de  
Capacitação da Haná**



**Manaus-AM**

**2025**

## EQUIPE TÉCNICA

**Diretor:**

Raimundo Cláudio Souza Gomes

**Coordenador Geral**

Roberto Higino Pereira da Silva

**Coordenadores Pedagógicos:**

Joésia Moreira Julião Pacheco

Joelma Monteiro de Carvalho

**Professores**

Celso Barbosa Carvalho

Rafael Facioni Scalabrin

**Apoio Técnico Hana Electronics**

Weslley Fábio Ferreira Santos

# **SUMÁRIO**

<b>Apresentação .....</b>	<b>4</b>
<b>Módulo 4:.....</b>	<b>5</b>
<b>Organização dos Projetos Finais do Curso de Capacitação da Hana .....</b>	<b>5</b>
<b>Referências.....</b>	<b>20</b>

## APRESENTAÇÃO

O Curso de Extensão “Gerenciamento de Energia, Modos de Operação de Baixo Consumo, Segurança e Técnicas de Localização do iMCP HTNB32L”, oferecido pela Universidade do Estado do Amazonas (UEA), teve como propósito proporcionar formação técnica e aplicada em sistemas embarcados de baixo consumo energético. Com carga horária de 30 horas, a capacitação foi desenvolvida pelos pesquisadores do Hub Tecnologia e Inovação, centro de pesquisa vinculado à Escola Superior de Tecnologia da UEA, situada em Manaus/AM.

A base de conhecimento, centrou-se na implementação de rotinas de interrupção (ISR) com comunicação via filas, promovendo a compreensão prática da interação entre hardware e software em sistemas de tempo real. Os estudantes foram conduzidos por um percurso de aprendizado que incluiu o estudo dos modos de economia de energia em sistemas embarcados — com ênfase em Sleep, Deep Sleep e Power Saving Mode (PSM) —, além da configuração de temporizadores e geração de sinais PWM para controle de intensidade luminosa de LEDs.

A proposta pedagógica culminou na elaboração de um Projeto Final, onde os estudantes aplicaram os conhecimentos adquiridos de forma integrada. As atividades envolveram o uso de ferramentas de controle de versão, como o Git, além da documentação de código e a revisão geral dos tópicos abordados. Cada grupo teve a oportunidade de desenvolver até quatro projetos, os quais foram apresentados à banca de professores e à comunidade acadêmica, demonstrando domínio técnico, criatividade e capacidade de inovação.

O curso teve como objetivo não apenas formar profissionais com competências técnicas específicas, mas também estimular o protagonismo estudantil e a inserção qualificada no mercado de trabalho. A iniciativa reafirma o compromisso social da Universidade do Estado do Amazonas com a formação de excelência e com o desenvolvimento tecnológico da região amazônica.

Professor Roberto Higino Pereira da Silva

Coordenador Geral

## Módulo 4:

# Organização dos Projetos Finais do Curso de Capacitação da Hana

Chega-se agora a um dos momentos mais empolgantes e inspiradores deste Curso de Capacitação da Hana: o Módulo de Organização dos Projetos Finais! Se os capítulos anteriores foram acompanhados, consolidando-se os conceitos e as tecnologias, prepara-se, então, para a materialização de todo esse conhecimento em soluções reais e inovadoras.

Este módulo convida à imersão nas mentes brilhantes dos alunos. Aqui, não apenas será compreendido como os projetos foram estruturados e organizados, mas também ter-se-á a oportunidade única de explorar as ideias, os desafios superados e as conquistas daqueles que percorreram o mesmo caminho.

Cada projeto final representa uma história de dedicação, criatividade e aplicação prática. Ao conhecerem-se essas iniciativas, não só o próprio aprendizado será consolidado, mas também uma fonte inesgotável de inspiração para futuros desenvolvimentos será encontrada. Observa-se, assim, como a teoria se transforma em dispositivos inteligentes, sistemas conectados e soluções que impactam o dia a dia.

Prepara-se para a surpresa, para aprender com a experiência alheia e, quem sabe, para o despertar daquela ideia que faltava para o próximo grande projeto! Desvenda-se, juntos, o poder da inovação *made in Hana!*

## 4.1. Organização dos Projetos Finais do Curso de Capacitação da Hana

Nº	Nome do Projeto	Função Principal
1	<b>SmartDoor</b>	Monitorar porta e iluminação; acionar buzzer via MQTT
2	<b>SenseClima</b>	Enviar temperatura e umidade periodicamente via MQTT
3	<b>AirControl</b>	Controlar ar-condicionado com comandos via MQTT e IR
4	<b>CoreHub</b>	Centralizar dados, decidir e controlar os demais sistemas

- **SmartDoor:** Esse projeto será responsável por monitorar necessidade de ligar o ar-condicionado e monitorar abertura e fechamento da porta. O monitoramento para ligar o ar-condicionado será feito a partir da iluminação. O dispositivo deverá conter um sensor de luminosidade para detectar quando a luz está ligada. Também deverá monitorar se a porta está aberta ou fechada. Essas informações de luz ligada e porta fechada deverão ser enviadas para o broker MQTT. Esse projeto também deverá receber a informação para acionamento de um buzzer através de um tópico MQTT. O monitoramento da porta será feito através de um reed switch. Deverá obrigatoriamente utilizar o HTNB32L como microcontrolador. O sensor de luz será o BH1750-FVI / Sensor de Luminosidade Lux Digital.

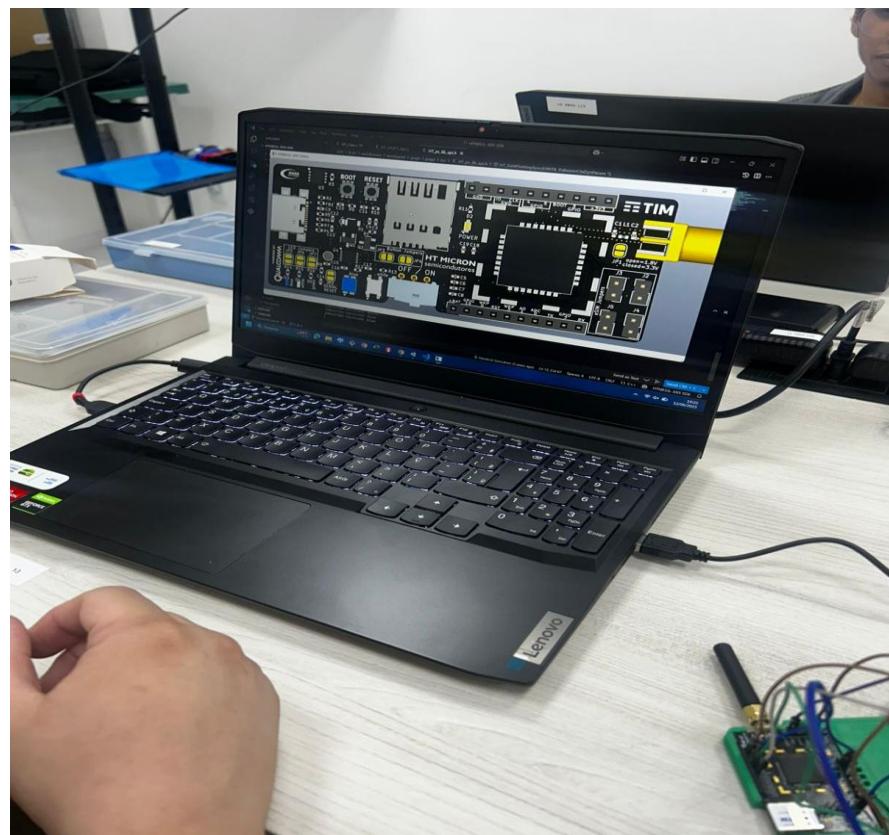
- **SenseClima:** Esse projeto será responsável por desenvolver um dispositivo que fará leituras periódicas de temperatura e umidade do ambiente e enviará essas informações para tópicos MQTT. A leitura de temperatura e umidade será realizada utilizando um sensor DHT22. Deverá obrigatoriamente utilizar o HTNB32L como microcontrolador.

- **AirControl:** Esse projeto será responsável por desenvolver um dispositivo para controlar o acionamento e temperatura dos equipamentos de ar-condicionado. Deverá ter um dispositivo por aparelho de ar-condicionado. O valor da temperatura e o estado do ar-condicionado serão recebidos através de tópicos MQTT. Os comandos serão enviados do dispositivo para o equipamento de ar-condicionado.

através do Módulo Emissor Infravermelho IR KY-005. Deverá obrigatoriamente utilizar o HTNB32L como microcontrolador. Não terá PCB.

- **CoreHub:** Será responsável por receber, processar e interpretar todos os dados recebidos dos sensores via MQTT. Também será responsável por implementar o sistema de controle e comandar os dispositivos que controlarão os aparelhos de ar-condicionado e será responsável por decidir o momento de acionar o buzzer caso a porta fique aberta por muito tempo. Todo esse sistema deverá obrigatoriamente ser desenvolvido no HTNB32L. Não terá PCB.

Figura 1: aula prática do curso



**Fonte:** Lab 4.1 (Equipe técnica do projeto)

## 1. SmartDoor

Nome completo: SmartDoor – Monitoramento de Acesso e Iluminação

### 1.1. Objetivo

Desenvolver um dispositivo embarcado capaz de monitorar a abertura de uma porta e o nível de luminosidade do ambiente, enviar essas informações via MQTT e responder a comandos para ativar um buzzer.

### 1.2. Componentes Utilizados

Microcontrolador: iMCP HTNB32L

Sensor de luminosidade: BH1750-FVI (I2C)

Sensor de porta: Reed Switch (GPIO)

Atuador: Buzzer (GPIO)

Conectividade: NB-IoT (via modem interno do HTNB32L)

Broker MQTT:

- IP: 131.255.82.115;
- Porta: 1883;
- PCB personalizada: obrigatória para montagem do dispositivo final.

### 1.3. Requisitos Funcionais

#### 1.3.1. Leitura do sensor BH1750-FVI

- Capturar o valor da luminosidade do ambiente periodicamente.
- Interpretar valor lido para saber se a luz está acesa ou apagada.
- Quando o estado da lâmpada mudar de apagado para aceso, publicar uma vez o valor “ON”.
- Quando o estado da lâmpada mudar de aceso para apagado, publicar uma vez o valor “OFF”.

#### 1.3.2. Leitura do reed switch

- Detectar abertura e fechamento da porta.

- Quando o estado da porta mudar de fechado para aberto, publicar uma vez o valor "OPEN".
- Quando estado da porta mudar de aberto para fechado, publicar uma vez o valor "CLOSED".

### 1.3.3. Envio via MQTT

- Publicar os seguintes dados nos tópicos padronizados conforme a seção;
- Frequência dos Estado: a cada mudança de estado da porta ou da lâmpada.

### 1.3.4. Re却bimento via MQTT

- Escutar tópico para controle do buzzer conforme indicado na seção 1.4.
- Se receber "ON": ativar buzzer.
- Se receber "OFF": desativar buzzer.

## 1.4. Tópicos MQTT Padronizados

**IMPORTANTE:** Cada grupo deverá substituir <ambiente> pelo nome do local (ex: lab1, salaaula, auditorio) em letras minúsculas e sem espaços.

Finalidade	Tópico MQTT	Direção	Tipo de dado
Iluminação	hana/<ambiente>/smartdoor/light	publish	"ON" ou "OFF"
Porta	hana/<ambiente>/smartdoor/door	publish	"OPEN" ou "CLOSED"
Buzzer	hana/<ambiente>/smartdoor/buzzer	subscribe	"ON" ou "OFF"

## 1.5. Desenvolvimento de um PCB

- A placa deve integrar todos os componentes citados.
- Deve conter conector para alimentação.
- Deverá ter marcações serigrafadas identificando pinos e funções.
- Deve ser projetada usando software apropriado (ex: EasyEDA, KiCad).
- O layout deverá ser entregue em formato .pdf e.gerber.
- O esquemático deverá ser entregue em .pdf.

## 1.6. Observações Técnicas

- Sensor BH1750-FVI comunica via I2C. Verifique o endereço padrão.
- Reed switch deve possuir resistor de pull-up.
- O buzzer utilizado no projeto permite o controle direto via GPIO.

- Implemente reconexão MQTT automática em caso de falha.

### 1.7. Critérios de Avaliação

- Funcionamento correto do envio dos estados de lâmpada e porta via MQTT.
- Funcionamento do buzzer com comandos MQTT.
- Uso correto dos tópicos conforme padronizado na seção 1.4.
- Documentação completa, na wiki do github, com evolução do projeto e dificuldades encontradas durante o desenvolvimento. Exemplo de documentação da wiki: <https://github.com/rafaelfacioni/Hands-On-Linux/wiki>.
- Projeto de PCB funcional e documentado na wiki.
- Apresentação prática do projeto final.
- (Opcional) Adicione na Wiki do repositório um registro pessoal com os principais aprendizados adquiridos ao longo do curso.

## SenseClima

Nome completo: SenseClima – Estação Ambiental MQTT

### 1.8. Objetivo

Desenvolver um dispositivo embarcado capaz de realizar leituras periódicas de temperatura e umidade do ambiente, utilizando o sensor DHT22, e enviar essas informações via MQTT para um broker remoto. O projeto deverá obrigatoriamente utilizar o microcontrolador iMCP HTNB32L e ser alimentado por bateria, adotando práticas de economia de energia para aumentar sua autonomia.

### 1.9. Componentes Utilizados

- Microcontrolador: iMCP HTNB32L
- Sensor de temperatura e umidade: DHT22 (GPIO)
- Conectividade: NB-IoT (via modem interno do HTNB32L)
- Alimentação: Bateria recarregável (ex: Li-Ion 3.7 V)
- Broker MQTT:
- IP: 131.255.82.115;
- Porta: 1883;
- PCB personalizada: obrigatória para montagem do dispositivo final.

### 1.10. Requisitos Funcionais

#### 1.10.1. Leitura do sensor DHT22

- Capturar os valores de temperatura (em °C) e umidade relativa (%).
- Validar os dados lidos; caso inválidos, publicar "NaN".

#### 1.10.2. Envio via MQTT

- Publicar os dados nos tópicos padronizados (ver seção 2.4).
- O envio dos dados deve ser feito em intervalos definidos pelo próprio desenvolvedor.

#### 1.10.3. Recebimento via MQTT

- Escutar tópico MQTT para controle do intervalo de envio.
- Atualizar o intervalo de leitura/transmissão caso receba comandos válidos.

#### 1.10.4. Economia de energia

- O dispositivo deverá utilizar modo low power entre as leituras e transmissões.
- Os desenvolvedores deverão determinar os melhores valores de intervalo entre leitura e envio para garantir o equilíbrio entre economia de energia e funcionalidade do sistema.

#### 1.11. Tópicos MQTT Padronizados

**IMPORTANTE:** Cada desenvolvedor deverá substituir <ambiente> pelo nome do local e <board> por um identificador único para o dispositivo, ambos em letras minúsculas e sem espaços.

Finalidade	Tópico MQTT	Direção	Tipo de dado
Temperatura	hana/<ambiente>/senseclima/<board>/temperatura	publish	Ex: "27.8"
Umidade	hana/<ambiente>/senseclima/<board>/humidity	publish	Ex: "64.2"
Intervalo	hana/<ambiente>/senseclima/<board>/interval	subscribe	Ex: "30"

#### 1.12. Desenvolvimento de uma PCB

- A placa deve integrar o microcontrolador HTNB32L e o sensor DHT22.
- Deve conter conector para bateria e, preferencialmente, circuito de recarga.
- Deverá ter marcações serigrafadas identificando pinos e funções.
- Deve ser projetada usando software apropriado (ex: EasyEDA, KiCad).
- O layout deverá ser entregue em formato .pdf e .gerber.
- O esquemático deverá ser entregue em .pdf.

#### 1.13. Observações Técnicas

- O sensor DHT22 requer estabilização após energização.
- Evitar leituras em sequência muito próximas para preservar a bateria.
- Implementar reconexão automática ao broker MQTT em caso de falha.
- Priorizar modos de baixo consumo do HTNB32L entre ciclos.
- Os intervalos de operação devem ser testados e definidos conforme o caso de uso.

#### 1.14. Critérios de Avaliação

- Funcionamento correto do envio dos dados de temperatura e umidade via MQTT.
- Controle remoto dos intervalos de leitura e envio de dados.
- Uso correto dos tópicos MQTT conforme padronizado na seção 2.4.
- Documentação completa, na wiki do github, com evolução do projeto e dificuldades encontradas durante o desenvolvimento. Exemplo de documentação da wiki: <https://github.com/rafaelfacioni/Hands-On-Linux/wiki>.
- Projeto de PCB funcional e documentado na wiki.
- Apresentação prática do projeto final.
- (Opcional) Adicione na Wiki do repositório um registro pessoal com os principais aprendizados adquiridos ao longo do curso.

## 2. AirControl

Nome completo: AirControl – Controle Inteligente de Climatização

### 2.1. Objetivo

Desenvolver um dispositivo embarcado capaz de controlar o acionamento e a temperatura de equipamentos de ar-condicionado por meio de comandos recebidos via MQTT. O dispositivo deverá interpretar os comandos e emitir sinais infravermelhos apropriados utilizando o módulo IR KY-005. Cada unidade do AirControl será responsável por controlar um único equipamento de ar-condicionado. O microcontrolador obrigatório é o iMCP HTNB32L.

### 2.2. Componentes Utilizados

- Microcontrolador: iMCP HTNB32L
- Módulo Emissor IR: KY-005 (GPIO)
- Conectividade: NB-IoT (via modem interno do HTNB32L)
- Broker MQTT:
  - IP: 131.255.82.115
    - Porta: 1883
- PCB personalizada: não aplicável neste projeto

### 2.3. Requisitos Funcionais

#### 2.3.1. Recebimento de comandos via MQTT

- O dispositivo deverá escutar dois tópicos MQTT distintos: um para o estado do ar- condicionado (ligar/desligar) e outro para o valor da temperatura desejada.
- Se receber "ON" no tópico de estado, deverá enviar o sinal IR correspondente para ligar o ar- condicionado.
- Se receber "OFF", deverá enviar o sinal IR para desligar.
- Se receber um valor numérico (ex: "24") no tópico de temperatura, deverá emitir o sinal IR correspondente à configuração da temperatura.

#### 2.3.2. Emissão de sinal IR

- Utilizar o módulo IR KY-005 para emitir os sinais.

- Os sinais IR devem ser compatíveis com o protocolo do modelo de ar-condicionado utilizado.
- O dispositivo deve garantir que os comandos IR sejam enviados de forma robusta (ex: repetição, delay entre comandos, verificação de sucesso opcional).

### **2.3.3. Modo de operação**

- O dispositivo deverá funcionar continuamente, aguardando comandos MQTT.
- É desejável que haja uma rotina de reconexão MQTT automática em caso de perda de conexão.

## **2.4. Tópicos MQTT Padronizados**

**IMPORTANTE:** Cada grupo deverá substituir <ambiente> pelo nome do local (ex: lab1, salaaula) e

<equipamento> por um identificador único do ar-condicionado (ex: ac1, ac2), ambos em letras minúsculas e sem espaços.

Finalidade	Tópico MQTT	Direção	Tipo de dado
Estado	hana/<ambiente>/aircontrol/<equipamento>/power	subscribe	"ON" ou "OFF"
Temperatura	hana/<ambiente>/aircontrol/<equipamento>/temperature	subscribe	"22", "24", "26"

## **2.5. Desenvolvimento de PCB**

Não se aplica. Este projeto não exige o desenvolvimento de uma placa personalizada.

## **2.6. Observações Técnicas**

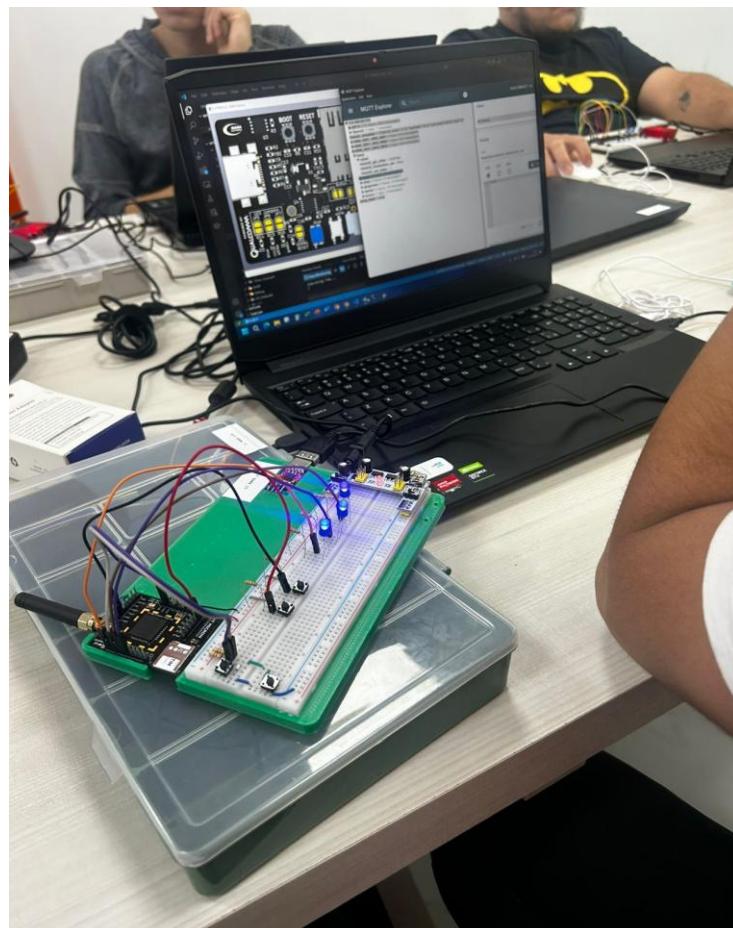
- O módulo KY-005 deverá ser conectado a um pino digital do HTNB32L, com controle via PWM ou pulso digital conforme o protocolo IR exigido.
- A biblioteca de controle IR utilizada deverá permitir emissão com protocolo compatível com o ar-condicionado (ex: NEC, LG, Samsung).
- Em alguns casos, pode ser necessário capturar previamente os códigos IR reais do controle original.
- Recomenda-se testar o tempo entre comandos e uso de repetição para maior confiabilidade.

- Implementar reconexão MQTT automática.

### 3.7 Critérios de Avaliação

- Funcionamento correto do recebimento dos comandos via MQTT.
- Funcionamento correto da emissão de sinal IR para controle dos aparelhos de ar- condicionado.
- Uso correto dos tópicos MQTT conforme padronizado na seção 3.4.
- Documentação completa, na wiki do github, com evolução do projeto e dificuldades encontradas durante o desenvolvimento. Exemplo de documentação da wiki: <https://github.com/rafaelfacioni/Hands-On-Linux/wiki>.
- Apresentação prática do projeto final.
- (Opcional) Adicione na Wiki do repositório um registro pessoal com os principais aprendizados adquiridos ao longo do curso.

Figura 2: aula prática embarcados



Fonte: Lab 4.1 (Equipe técnica do projeto)

### 3. CoreHub

Nome completo: CoreHub – Central de Decisão e Automação.

#### 3.1. Objetivo

Desenvolver um dispositivo embarcado capaz de atuar como uma central de automação e controle, responsável por processar dados provenientes de sensores publicados por outros dispositivos via MQTT, tomar decisões com base em regras definidas e publicar comandos para atuadores (como ar- condicionado e buzzer). O projeto deverá obrigatoriamente utilizar o microcontrolador iMCP HTNB32L e não exige o desenvolvimento de uma placa personalizada.

#### 3.2. Componentes Utilizados

- Microcontrolador: iMCP HTNB32L
- Conectividade: NB-IoT (via modem interno do HTNB32L)
- Broker MQTT:
- IP: 131.255.82.115
- Porta: 1883
- PCB personalizada: não aplicável neste projeto

#### 3.3. Requisitos Funcionais

##### 3.3.1. Subscrição de dados via MQTT

- O dispositivo deve escutar os seguintes dados enviados por outros projetos:
- Estado da porta (OPEN / CLOSED) – do projeto SmartDoor
- Temperatura e umidade – do projeto SenseClima
- Estado de funcionamento do ar-condicionado – do projeto AirControl

##### 3.3.2. Lógica de decisão automatizada

Monitorar o tempo que a porta permanece aberta:

- Se a porta permanecer aberta por mais de um intervalo definido (ex: 60 segundos), publicar comando "ON" para acionar o buzzer.
- Caso a porta seja fechada antes desse tempo, não acionar o buzzer.

Controlar o ar-condicionado:

- Se a temperatura lida for superior a um limite definido (ex: 28 °C), publicar "ON" para ligar o ar-condicionado.
- Se a temperatura cair abaixo de um limite mínimo (ex: 24 °C), publicar "OFF" para desligar ar-condicionado.

### 3.3.3. Publicação de comandos via MQTT

Enviar comandos MQTT para dispositivos AirControl e SmartDoor utilizando os tópicos padronizados.

#### 3.4. Modo de operação

- O dispositivo deverá funcionar continuamente como central de decisão.
- Implementar reconexão automática ao broker MQTT em caso de falha.

#### 3.5. Tópicos MQTT Padronizados

**IMPORTANTE:** Substituir <ambiente>, <board> e <equipamento> conforme o padrão geral.  
O CoreHub deve escutar e publicar em múltiplos tópicos.

Finalidade	Tópico MQTT	Direção	Tipo de dado
Iluminação	hana/<ambiente>/smartdoor/light	subscribe	"ON" / "OFF"
Porta	hana/<ambiente>/smartdoor/door	subscribe	"OPEN" / "CLOSED"
Buzzer	hana/<ambiente>/smartdoor/buzzer	publish	"ON" / "OFF"
Temp. Amb.	hana/<ambiente>/senseclima/<board>/temperature	subscribe	Ex: "27.8"
Umidade	hana/<ambiente>/senseclima/<board>/humidity	subscribe	Ex: "64.2"
Estado Ar	hana/<ambiente>/aircontrol/<equipamento>/power	publish	"ON" ou "OFF"
Temp. Ar	hana/<ambiente>/aircontrol/<equipamento>/temperature	publish	"22", "24", "26"

#### 3.6. Desenvolvimento de PCB

Não se aplica. Este projeto não exige o desenvolvimento de uma placa personalizada.

#### 3.7. Observações Técnicas

- O tempo de porta aberta pode ser controlado com timers no firmware.
- A decisão do ar-condicionado deve considerar atualizações periódicas de temperatura.
- O sistema deve garantir robustez em ambientes com desconexões temporárias.
- Implementar reconexão automática ao broker MQTT.

### 3.8. Critérios de Avaliação

- Funcionamento correto da lógica de decisão baseada nos dados MQTT.
- Publicação de comandos funcionais para dispositivos remotos (buzzer, ar-condicionado).
- Subscrição adequada e eficiente aos tópicos MQTT de sensores.
- Documentação completa, na wiki do github, com evolução do projeto e dificuldades encontradas durante o desenvolvimento. Exemplo de documentação: <https://github.com/rafaelfacioni/Hands-On-Linux/wiki>.
- Apresentação prática do projeto demonstrando o funcionamento da automação.
- (Opcional) Adicione na Wiki do repositório um registro pessoal com os principais aprendizados adquiridos ao longo do curso.

#### Github dos Projetos

SmartDoor: <https://github.com/rafaelfacioni/smart-door>

SenseClima: <https://github.com/rafaelfacioni/sense-clima>

AirControl: <https://github.com/rafaelfacioni/air-control>

CoreHub: <https://github.com/rafaelfacioni/core-hub>

## Referências

ESPRESSIF. Documentação oficial da biblioteca WiFi do ESP32 (Arduino). [S. I.], [2020]. Disponível em: <https://github.com/espressif/arduino-esp32/tree/master/libraries/WiFi>. Acesso em: 28 jul. 2025.

ESPRESSIF. Documentação oficial do ESP32. [S. I.], [2020]. Disponível em: <https://docs.espressif.com>. Acesso em: 28 jul. 2025.

MQTT.ORG. Site oficial do protocolo. [S. I.], [2020]. Disponível em: <https://mqtt.org>. Acesso em: 28 jul. 2025.

RANDOM NERD TUTORIALS. Tutorial: conectando o ESP32 ao Wi-Fi e realizando uma requisição HTTP. [S. I.], [2020]. Disponível em: <https://randomnerdtutorials.com/esp32-http-get-post-arduino/>. Acesso em: 28 jul. 2025.

RANDOM NERD TUTORIALS. Tutorial prático MQTT com ESP32 + Mosquitto + MQTT.fx. [S. I.], [20--?]. Disponível em: <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>. Acesso em: 28 jul. 2025.

SCIKIT-LEARN. Documentação do Scikit-learn: biblioteca de machine learning em Python. [S. I.], [2020]. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 28 jul. 2025.

THE THINGS NETWORK. Comunidade e plataforma aberta para LoRaWAN. [S. I.], [2020]. Disponível em: <https://www.thethingsnetwork.org>. Acesso em: 28 jul. 2025.