

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES / ENGENHARIA DE CONTROLE E
AUTOMAÇÃO
Trabalho Final de Curso

Gustavo Palmier Carvalho Reis

**Implementação de Roteador de Borda Thread e Rede Matter com
ESP32-C6 e Home Assistant**

Trabalho Final de Curso como parte dos
requisitos para obtenção do título de bacharel
em Engenharia de Controle e Automação
apresentado à Pontifícia Universidade Católica
de Goiás.

BANCA EXAMINADORA:

**Prof. Dr. Antônio Marcos Melo Medeiros – Orientador. Pontifícia
Universidade Católica de Goiás - PUC Goiás.**

**Prof. Dr. Cassio Hideki Fujisawa – Coorientador. Pontifícia
Universidade Católica de Goiás - PUC Goiás.**

**Prof. Me. Marcilon Fonseca de Lima – Coorientador. Pontifícia
Universidade Católica de Goiás - PUC Goiás.**

Goiânia, 04 de Dezembro de 2025.

Implementação de Roteador de Borda Thread e Rede Matter com ESP32-C6 e Home Assistant

Gustavo Palmier Carvalho Reis¹, Antônio Marcos Melo Meideiros², Cassio Hideki Fujisawa³, Marcilon Fonseca de Lima⁴.

¹Pontifícia Universidade Católica de Goiás, Goiânia, Goiás, Brasil, gu.palmier@gmail.com

²Pontifícia Universidade Católica de Goiás, Goiânia, Goiás, Brasil, marcosmelo@pucgoias.edu.br

³Pontifícia Universidade Católica de Goiás, Goiânia, Goiás, Brasil, hideki@pucgoias.edu.br

⁴Pontifícia Universidade Católica de Goiás, Goiânia, Goiás, Brasil, marcilon@pucgoias.edu.br

RESUMO: A fragmentação dos protocolos de comunicação é um dos principais obstáculos para a ampla aceitação da Internet das Coisas (IoT) em casas. Este trabalho apresenta uma implementação prática de uma rede usando o novo padrão *Matter*, que promete uma interoperabilidade mais eficiente. O foco é criar uma infraestrutura acessível e de baixo custo, utilizando o microcontrolador ESP32-C6 como Roteador de Borda *Thread* e a plataforma *open source Home Assistant* como o controlador central. A abordagem foi experimental, aplicando a virtualização do servidor de automação em um ambiente *Windows (VirtualBox)* e gravando o firmware oficial da *Espressif* pelo *Visual Studio Code*. Os resultados mostraram que foi possível fazer o comissionamento dos dispositivos e a comunicação entre as redes Wi-Fi e *Thread* com baixa latência e sem depender de serviços em nuvem. Assim, a combinação do hardware ESP32-C6 com software livre se mostra como uma solução econômica, privada e eficaz para a automação residencial moderna.

Palavras chaves: *Matter, Thread, ESP32-C6, Home Assistant, Roteador de Borda Thread.*

ABSTRACT: The fragmentation of communication protocols is one of the main obstacles to the widespread adoption of the Internet of Things (IoT) in homes. This work presents a practical implementation of a network using the new *Matter* standard, which promises more efficient interoperability. The focus is to create an accessible, low-cost infrastructure using the ESP32-C6 microcontroller as a *Thread* Border Router and the open-source *Home Assistant* platform as the central controller. The approach was experimental, applying server virtualization for home automation in a *Windows environment (VirtualBox)* and flashing *Espressif's* official firmware using *Visual Studio Code*. The results showed that it was possible to commission the devices and enable communication between Wi-Fi and *Thread* networks with low latency and without relying on cloud services. Thus, the combination of ESP32-C6 hardware with free software proves to be an economical, private, and effective solution for modern home automation.

Keywords: *Matter, Thread, ESP32-C6, Home Assistant, Thread Border Router.*

I. Introdução

A Internet das Coisas (IoT) vem mudando radicalmente a forma como lidamos com ambientes residenciais e industriais, permitindo o monitoramento e o controle de dispositivos à distância. Porém, o mercado de automação residencial tem um grande desafio pela frente: a fragmentação dos protocolos de comunicação. Muitas vezes, dispositivos de diferentes fabricantes não conseguem se comunicar de forma nativa, fazendo com que seja necessário vários aplicativos e hubs próprios. É aí que entra o protocolo *Matter*, um padrão aberto e unificado criado pela *Connectivity Standards Alliance (CSA)*, que usa o protocolo IP para conectar dispositivos de casas inteligentes, garantindo segurança, confiabilidade e interoperabilidade entre os ecossistemas [1]. Este trabalho se concentra em como implementar essa tecnologia usando o protocolo de rede *Thread*, que é uma rede *mesh* de baixa potência e baixa latência, baseada no padrão IEEE 802.15.4 [2].

Mas, apesar do que o *Matter* promete em termos de unificação, ainda há dificuldades na hora de montar uma infraestrutura de rede local que suporte o protocolo *Thread*, especialmente por causa do hardware necessário para o Roteador de Borda (*Border Router*). Muitas vezes, essa parte depende de equipamentos comerciais que custam caro. Em contrapartida, a *Espressif Systems* apresentou o *SoC ESP32-C6*, que junta rádio Wi-Fi 6 e 802.15.4, oferecendo uma solução de baixo custo para esse problema [3].

Para responder a essa pergunta, o objetivo geral deste trabalho é implementar e validar uma rede *Matter* local com microcontroladores da linha ESP32 e software de código aberto. Os objetivos específicos para alcançar esse propósito incluem: configurar o firmware oficial da *Espressif* para transformar o ESP32-C6 em um *Thread Border Router* [4]; implementar o controlador *Matter* na plataforma *Home Assistant*, rodando em uma máquina virtual e aproveitando a capacidade de processar dados

localmente [5]; e, por fim, fazer o pareamento e controle de dispositivos finais na rede que criamos, checando a estabilidade da comunicação.

A importância deste estudo se justifica pela crescente demanda de democratizar o acesso a tecnologias avançadas em IoT. Usar o ESP32-C6 como um Roteador de Borda oficial permite que desenvolvedores e estudantes criem *gateways* robustos sem gastar muito, evitando assim ficar preso a um único fabricante. Além disso, validar um ambiente com *Home Assistant* reforça a importância da privacidade e da soberania dos dados do usuário, que são questões centrais na especificação do protocolo *Matter*.

Finalmente, o trabalho está estruturado da seguinte forma: a seção 2 traz a Fundamentação Teórica sobre os protocolos *Matter* e *Thread*; a seção 3 descreve os Materiais e Métodos usados na montagem do experimento; a seção 4 apresenta os Resultados e Discussão sobre como a rede funcionou; e, por último, a seção 5 traz as Considerações Finais.

II. Referencial teórico

A. Internet das Coisas e o Desafio da Interoperabilidade

A popularização da Internet das Coisas (IoT) trouxe uma gama de dispositivos inteligentes para nossas casas. Mas um dos problemas é que o mercado se dividiu em vários protocolos de comunicação que não se conversam, como Zigbee, Z-Wave, Wi-Fi, Thread e Bluetooth LE, conforme **Fig 1**. Isso resulta em "silos" tecnológicos, fazendo com que o usuário tenha que usar vários aplicativos e hubs de diferentes marcas só para controlar os dispositivos em casa. Por isso, a interoperabilidade virou um dos maiores obstáculos para que as casas inteligentes se tornem comuns.

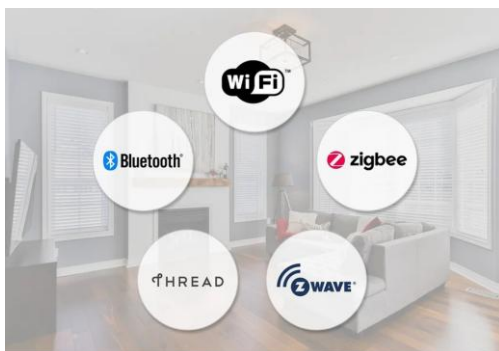


Fig. 1: Diversidade de protocolos de comunicação sem fio no ambiente residencial **Fonte:** IOTICS[6].

B. O Protocolo Matter e a Rede Thread

Para enfrentar a fragmentação, a *Connectivity Standards Alliance (CSA)* criou o padrão *Matter*. Ao contrário de outros padrões anteriores, o *Matter* não é só um protocolo de rádio; ele é uma camada de aplicação que funciona sobre *IP (Internet Protocol)*, permitindo que

dispositivos de marcas distintas se comuniquem localmente, sem precisar da nuvem [7]. Para a camada de transporte, a solução adota o protocolo *Thread* [8]. A **Fig. 2** detalha a anatomia da arquitetura implementada, evidenciando a interação entre os diferentes segmentos da rede.

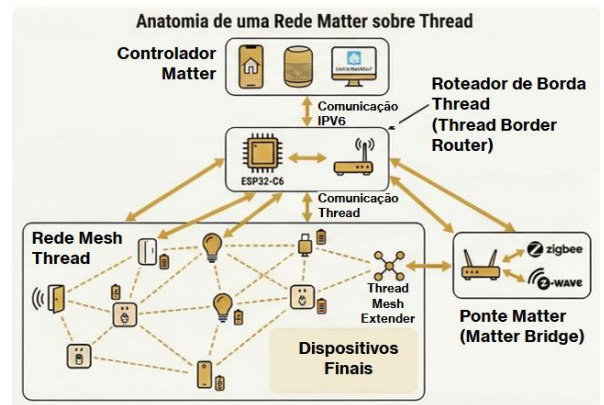


Fig. 2: Arquitetura e topologia de uma rede *Matter* sobre *Thread*
Fonte: Baseado em threadgroup [9].

No nível superior, encontram-se os Controladores Matter (como smartphones e o servidor *Home Assistant*), que operam na rede local convencional (Wi-Fi/Ethernet) e se comunicam via IPv6. O elemento central da topologia é o Roteador de Borda Thread (Thread Border Router), função desempenhada nesta pesquisa pelo microcontrolador ESP32-C6. Ele atua como uma ponte transparente, traduzindo os pacotes IPv6 entre a interface de alta velocidade (Wi-Fi) e a interface de baixa potência (IEEE 802.15.4).

Na parte inferior do diagrama, observa-se a Rede Mesh Thread, composta pelos Dispositivos Finais (como lâmpadas, tomadas e sensores a bateria). Estes dispositivos formam uma malha auto-regenerável, permitindo que a comunicação se estenda por longas distâncias sem depender de um ponto central único. O esquema também demonstra a extensibilidade do ecossistema através de uma Ponte Matter (Matter Bridge), capaz de integrar dispositivos legados de outros protocolos, como Zigbee e Z-Wave, à mesma infraestrutura unificada."

C. O Microcontrolador ESP32-C6

O equipamento escolhido para ser o Roteador de Borda neste projeto é o ESP32-C6, fabricado pela *Espressif Systems*. Esse hardware é um *System-on-Chip (SoC)* que utiliza a arquitetura RISC-V de 32 bits, sendo desenvolvido especialmente para o padrão *Matter*. O grande diferencial dele é o suporte nativo a *dual-mode*: ele tem rádios para Wi-Fi 6 (2.4 GHz) e para IEEE 802.15.4 [4]. Essa característica permite que um único chip ESP32-C6 faça a conexão entre a rede Wi-Fi da casa e a rede *Thread* dos sensores, rodando a *stack* completa de um *Thread Border Router* oferecida

pelo SDK da *Espressif* [10]. Para o experimento foi usado uma ESP32-C6-DevKitC-1, como mostra a **Fig. 3**.



Fig. 3: Vista isométrica da placa de desenvolvimento ESP32-C6-DevKitC-1 **Fonte:** *Espressif Systems* [11].

D. Home Assistant e Virtualização

O gerenciamento da rede é feito pelo *Home Assistant*, uma plataforma de automação de código aberto que prioriza a privacidade e o controle local. Ao contrário de assistentes comerciais que dependem da nuvem para processar comandos, o *Home Assistant* funciona localmente, o que diminui o tempo de resposta e aumenta a segurança (ver **Fig.4**). Neste projeto, o software roda em uma máquina virtual (*VirtualBox*) sobre *Windows*, usando a integração oficial *Matter* (BETA) [5] para gerenciar os dispositivos conectados via *Border Router* ESP32-C6.

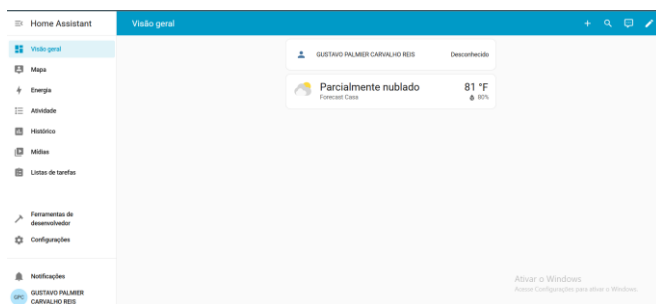


Fig. 4: interface do *Home Assistant* **Fonte:** elaborada pelo próprio autor

III. Metodologia

A. Materiais e Ferramentas

Para este projeto, é necessário uma série de recursos de hardware e software, que estão descritos a seguir. A escolha dos componentes levou em conta a disponibilidade e o suporte da documentação oficial das tecnologias *Matter* e *Thread*. O projeto foi desenvolvido com base na documentação técnica oficial dos fabricantes (*Espressif Systems* e *Connectivity Standards Alliance*), e consiste na configuração de uma máquina virtual no *Windows* (*VirtualBox*) para rodar o servidor de automação (*Home Assistant*) e no uso do *Visual Studio Code* para compilar e gravar o firmware no microcontrolador ESP32-C6.

1). Hardware

- **Microcontrolador ESP32-C6 (*Espressif Systems*):** Um SoC (*System on Chip*) que tem suporte nativo a Wi-Fi 6 (2.4 GHz) e IEEE 802.15.4, como mostra a **Fig. 5** que é essencial para que funcione como Roteador de Borda *Thread*, permitindo que ela funcione sem precisar de outro SoC, apesar de fazer com que ele não possa usar Wi-Fi e *Thread* ao mesmo tempo [10].

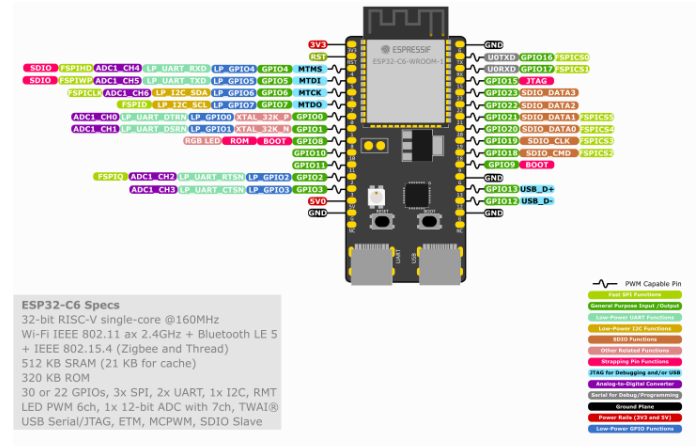


Fig. 5: Layout da ESP32-C6 DevKitC-1 **Fonte:** *Espressif Systems* [11].

- **Computador Pessoal (Host):** Um notebook que roda *Windows 10/11*, utilizado para a virtualização do servidor e a programação do microcontrolador.

- **Cabo USB-C:** Para alimentar e transferir dados (flash) do microcontrolador.

2). Software

- **Oracle VM *VirtualBox*:** Um software de virtualização usado para rodar o sistema do controlador, mostrado na **fig. 6**. Ele usa uma licença PUEL, para uso doméstico e escolar.

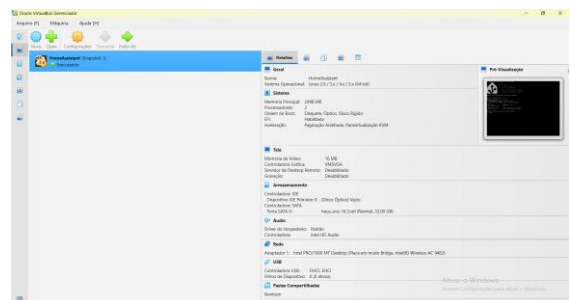


Figura 6: Interface do *Virtual Box* **Fonte:** elaborada pelo próprio autor

- ***Home Assistant OS (HAOS)*:** Sistema operacional *open-source* focado em automação, instalado na máquina virtual para atuar como Controlador *Matter* [5]. Vale ressaltar que este sistema é nativo para arquiteturas como o *Raspberry Pi*, mas optou-se pela virtualização no PC para eliminar a necessidade de aquisição de hardware dedicado adicional.

• ESP-IDF (*Espressif IoT Development Framework*): O ambiente de desenvolvimento oficial da *Espressif* (ver Fig.7), instalado no Visual Studio Code. Ele serve para compilar e gravar o firmware no ESP32-C6 [10].

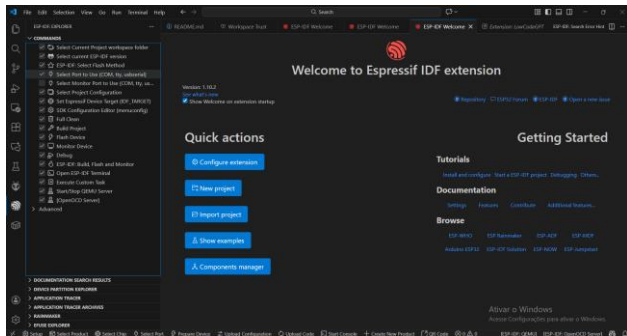


Fig. 7: Extensão ESP-IDF no Visual Code **Fonte:** elaborada pelo próprio autor.

• SDK *esp-thread-br*: Um kit de desenvolvimento de software específico para funcionar como Roteador de Borda *Thread*, distribuído pela própria *Espressif* [10].

Procedimentos Experimentais

1). Configuração do Controlador (Home Assistant)

A primeira etapa foi criar o ambiente do controlador central. Como o Windows não consegue rodar o *Home Assistant* nativamente com todas as funcionalidades do Supervisor necessárias para o Add-on *Matter*, foi necessário a máquina virtual [5].

1. A ferramenta Oracle VM VirtualBox foi instalada no computador host.

2. A imagem oficial do Home Assistant OS (formato VDI) foi baixada para máquinas virtuais.

3. Na VirtualBox, criou-se uma nova máquina virtual que suportasse EFI (Extensible Firmware Interface), configurando o adaptador de rede em modo "Bridge" (Ponte), Isso permite que o *Home Assistant* recebesse um endereço IP válido na nossa rede local, se comunicando diretamente com o ESP32-C6.

4. Após executada a máquina virtual, aparece o ip local do home assistant, como na Fig. 8. No navegador deve ser digitado ip do home assistant: 8123 para aparecer do software.

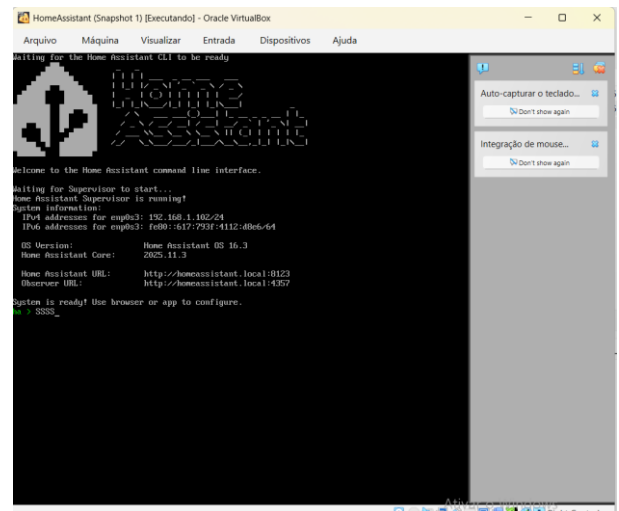


Fig. 8: Print da interface do *Home Assistant* após Instalação **Fonte:** Elaborada pelo próprio autor

5. Depois de criar um login e senha e fazer as configurações iniciais, o Add-on "*Matter Server*" (BETA) foi instalado pela loja de complementos do *Home Assistant*, ativando o suporte ao protocolo.

2). Implementação do Thread Border Router (ESP32-C6)

A segunda etapa focou na preparação do hardware. Para compilar e gravar o firmware, usamos a IDE *Visual Studio Code (VS Code)* com a extensão oficial da *Espressif* e suas dependências. Essa ferramenta facilitou bastante o gerenciamento das bibliotecas e a configuração das partições de memória do dispositivo [10].

1. No *VS Code*, foi importado o projeto exemplo *esp-thread-br* do SDK oficial [12].

2. Usando a interface da extensão, acessou-se o menu de configuração (*SDK Configuration Editor*), como na Fig. 9, para definir as credenciais da rede Wi-Fi e os parâmetros da rede *Thread*.

3.

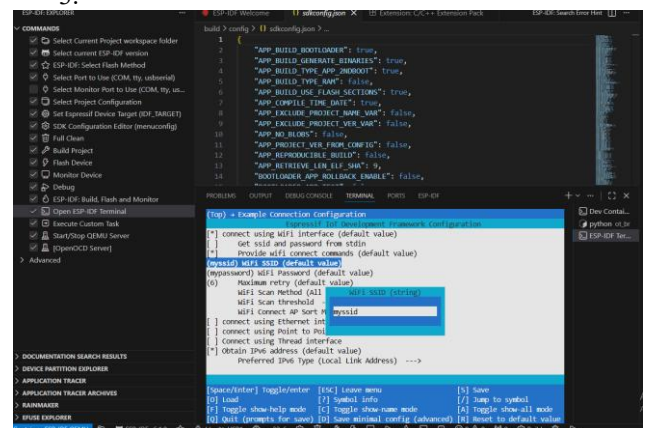


Fig. 9: IDF Terminal **Fonte:** Elaborada pelo próprio autor

4. O código foi compilado e gravado no ESP32-C6 via USB, diretamente pela interface do *VS Code*, que também permitiu monitorar os logs de inicialização através do terminal serial integrado.

3). *Comissionamento e Validação*

Com a infraestrutura no lugar, agora só falta o pareamento. O *Home Assistant*, através da integração *Matter*, reconhece a presença da rede *Thread*. Dispositivos compatíveis podem ser colocados em modo de pareamento e adicionados à interface do controlador, seja pela leitura de um *QR code* ou pela inserção manual dos códigos de pareamento, validando a comunicação bidirecional entre o servidor virtualizado e o dispositivo final via Roteador de Borda ESP32-C6.

IV. Resultados e Discussão

A. Inicialização do Roteador de Borda

A gravação do *firmware* no ESP32-C6 foi um sucesso através do *Visual Studio Code*. Assim que o dispositivo recebeu energia, ele iniciou a *stack* de rede dupla (Wi-Fi e 802.15.4). Usando o monitor serial, foi confirmado que a ESP32-C6 se conectou à rede Wi-Fi local, pegou um endereço IPv6 global e iniciou a rede *Thread*, assumindo as funções de líder e roteador de Borda. A extensão do *VS Code* se mostrou muito útil em comparação com os métodos tradicionais de linha de comando, já que proporcionou uma visão clara do consumo de memória da partição e facilitou a alteração de parâmetros da rede sem a necessidade de recompilar todo o projeto [13].

B. Integração com o Controlador Home Assistant

No ambiente virtual (*VirtualBox*), o *Home Assistant* funcionou de maneira estável. A configuração de rede em modo *Bridge* permitiu que a máquina virtual fosse reconhecida como um dispositivo parte da rede local (LAN), o que é fundamental para a descoberta de serviços mDNS (*Multicast DNS*) usados pelo protocolo *Matter*. Depois de instalar o complemento *Matter Server*, o sistema detectou a rede *Thread* criada pelo ESP32-C6. O pareamento dos dispositivos foi realizado com sucesso, prova de que a "ponte" entre a rede Wi-Fi (onde está o *Home Assistant*) e a rede *Thread* (onde estão os sensores/atuadores) funcionou.

C. Validação da Operação Local e Independência da Nuvem

A conexão independente com a internet (nuvem) foi validada de forma interna durante a fase de comissionamento. A rede foi configurada para funcionar apenas em ambiente local (LAN) com suporte a IPv6 e sem conexão ativa com a WAN (Internet). Nessas condições de isolamento, todas as funcionalidades de pareamento, monitoramento e controle através do *Home Assistant*

funcionaram normalmente. Isso confirmou que o tráfego *Matter* sobre *Thread* ocorre apenas dentro da infraestrutura local, assegurando que o serviço continue mesmo sem um provedor de internet. O teste foi conduzido em um ambiente controlado e com visada direta (sem obstáculos físicos significativos).

D. Discussão sobre Custo

A análise financeira mostra que a solução proposta é acessível. Os hubs comerciais que suportam *Matter* e *Thread*, como o Apple HomePod mini ou o Google Nest Hub 2ª Geração, custam entre R\$ 600,00 e R\$ 1.000,00 no Brasil. Por outro lado, o microcontrolador ESP32-C6, que usamos como Roteador de Borda, teve um custo entre R\$70 e R\$100. Isso significa que uma economia de quase 90% na construção da infraestrutura central da rede, o que comprova que dá pra democratizar o acesso a tecnologias de automação avançada com hardware de baixo custo.

V. Considerações finais

Este trabalho teve como foco principal a implementação e validação de uma infraestrutura de automação residencial seguindo o padrão *Matter*, usando o microcontrolador ESP32-C6 como Roteador de Borda *Thread* e o *Home Assistant* como controlador central. Com base nos resultados, é possível afirmar que o objetivo foi atingido.

Ficou demonstrado que a fragmentação dos protocolos de IoT pode ser superada com soluções abertas. O ESP32-C6 se mostrou uma alternativa viável e econômica em comparação aos hubs comerciais fechados, tornando a tecnologia de redes mesh IPv6 (*Thread*) mais acessível. A integração com o *VS Code* facilitou a configuração do *firmware*, permitindo que até quem não tem muita prática com linhas de comando conseguisse mexer.

A escolha de utilizar o *Home Assistant* no Windows também se provou eficaz, garantindo que o processamento dos dados acontecesse localmente. Isso apoia a ideia de que é possível criar uma casa inteligente que seja interoperável e privada, sem precisar depender de servidores na nuvem.

Para projetos futuros, seria interessante expandir a rede com mais dispositivos para realizar testes de estresse e analisar o consumo energético da bateria dos sensores *Thread* a longo prazo. Além disso, explorar recursos avançados do *Matter*, como o Multi-Admin (controle simultâneo por várias plataformas), pode render novos estudos.

A fim de passar do ambiente experimental para uma utilização contínua e definitiva, é possível substituir a máquina virtual por um hardware dedicado, como um Raspberry Pi. Essa mudança reduziria drasticamente o consumo de energia do servidor central. Para tornar o sistema mais amplo, poderia implementar um acesso

remoto seguro com o DuckDNS integrado ao Home Assistant, junto com uma VPN. Com essa configuração, seria possível monitorar e controlar os dispositivos Matter pela internet, de qualquer lugar, mantendo a criptografia e a segurança dos dados, sem expor a rede local diretamente à internet.

VI. Referências bibliográficas

- [1] C. Gomez, S. Madadi-Barough, P. Ruiz-Blanco, J. Lin, and R. Vidal, "Matter: IoT Interoperability for Smart Homes," IEEE Communications Magazine, vol. 63, no. 4, pp. 106-112, abr. 2025.
- [2] F. Hussain et al., "Performance Evaluation of an IEEE 802.15.4-Based Thread Network for Efficient Internet of Things Communications in Smart Cities," Appl. Sci., vol. 13, no. 13, p. 7745, jun. 2023.
- [3] A. Mota, C. Serôdio, and A. Valente, "Matter Protocol Integration Using Espressif's Solutions to Achieve Smart Home Interoperability," Electronics, vol. 13, no. 11, p. 2217, jun. 2024.
- [4] Espressif Systems, "ESP32-C6 Series Datasheet: 2.4 GHz Wi-Fi 6 & Bluetooth 5 (LE) & IEEE 802.15.4 SoC," ver. 1.2, Espressif Systems, Shanghai, China, 2024. [Online]. Disponível em: <https://www.espressif.com/en/products/socs/esp32-c6>. [Acesso em: 23 nov. 2025].
- [5] Home Assistant, "Matter (BETA): Integrations," Home Assistant Documentation, 2025. [Online]. Disponível em: <https://www.home-assistant.io/integrations/matter/> [Acesso em: 23 nov. 2025].
- [6] IOTICS, "Zigbee, Z-Wave, Wi-Fi... Should You Care Which Wireless Technology Your Smart Home Device Uses?," IOTICS Blog, 2022. [Online]. Disponível em: <https://www.iotics.io/> [Acesso em: 23 nov. 2025].
- [7] Connectivity Standards Alliance, "The Foundation for Connected Things: Matter," CSA, Davis, CA, USA, White Paper, 2023. [Online]. Disponível em: <https://csa-iot.org/all-solutions/matter/> [Acesso em: 23 nov. 2025].
- [8] Thread Group, "Thread 1.3.0 Specification," Thread Group, San Ramon, CA, USA, Tech. Spec., 2023. [Online]. Disponível em: <https://threadgroup.org/>. [Acesso em: 23 nov. 2025].
- [9] Thread Group, "Built for IoT: Smart Home," Thread Group Website, 2024. [Online]. Disponível em: <https://threadgroup.org/BUILT-FOR-IOT/Smart-Home> [Acesso em: 23 nov. 2025].
- [10] Espressif Systems, "ESP Thread Border Router SDK," GitHub Repository, 2024. [Online]. Disponível em: <https://github.com/espressif/esp-thread-br> [Acesso em: 23 nov. 2025].
- [11] Espressif Systems, "ESP32-C6-DevKitC-1 User Guide," Espressif Systems, Documentation, 2024. [Online]. Disponível em: <https://docs.espressif.com/> [Acesso em: 23 nov. 2025].
- [12] Espressif Systems, "OpenThread Border Router (OT_BR) Example for ESP-IDF," GitHub Repository, 2024. [Online]. Disponível em: <https://github.com/espressif/esp-idf/> [Acesso em: 23 nov. 2025].
- [13] A Smarter House, "The Shocking Truth About Matter Protocol Nobody Tells You," YouTube, 24 nov. 2025. [Video Online]. Disponível em: <http://www.youtube.com/watch?v=YwSEyPCWzyA> [Acesso em: 26 nov. 2025].