UNIVERSIDADE DE ITAÚNA

DAVI VENTURA CARDOSO PERDIGÃO
ERIC HENRIQUE DE CASTRO CHAVES

SISTEMAS EMBARCADOS CONECTADOS A INTERNET DAS COISAS

Conceito, Arquitetura, Tecnologias, Desafios, Tendências Futuras, Segurança e Privacidade

ITAÚNA
2023

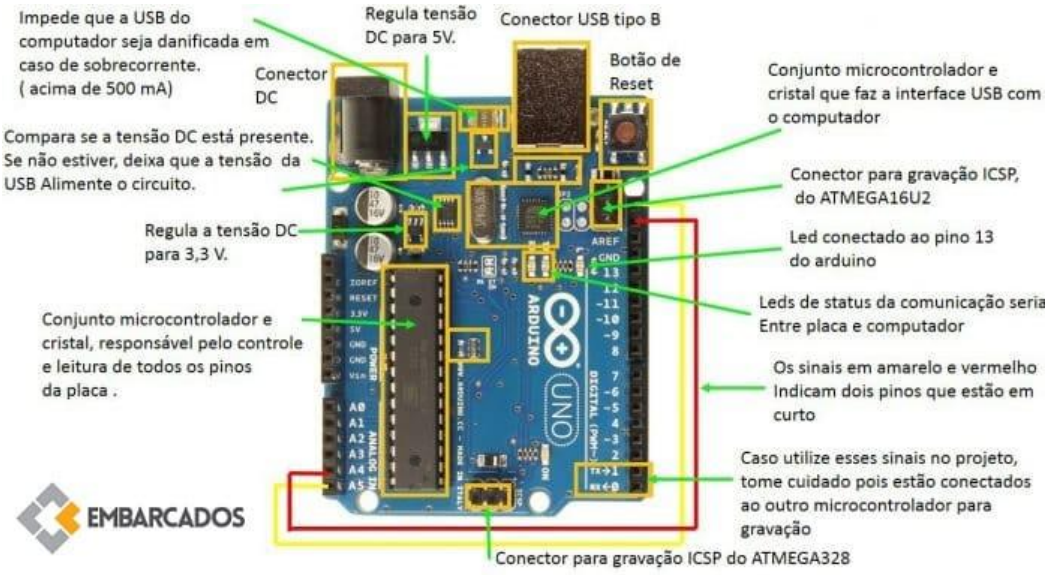
SUMÁRIO

1. INTRODUÇÃO

O conceito de Internet das Coisas (IoT) ganha cada vez mais destaque na era digital. Trata-se da interconexão de dispositivos físicos, como sensores, atuadores e microcontroladores, à Internet, possibilitando a troca de informações e o controle remoto desses dispositivos. Nesse contexto, os sistemas embarcados desempenham um papel fundamental.

Os sistemas embarcados são dispositivos eletrônicos com capacidade de processamento, armazenamento e comunicação, incorporados em outros sistemas maiores, como por exemplo, o conhecido Arduino (**Figura 1**). Eles são projetados para executar funções específicas e estão presentes em uma ampla gama de dispositivos, desde eletrodomésticos inteligentes até veículos autônomos.

Figura 1 - Arduino como exemplo de sistema embarcado.



Fonte: Embarcados, 2014.

A integração de sistemas embarcados com a IoT permite a criação de redes de dispositivos inteligentes interconectados, capazes de coletar dados em tempo real, transmiti-los pela Internet e realizar ações com base nessas informações. Isso possibilita a automação de processos, o monitoramento

remoto, a análise de dados em larga escala e a tomada de decisões inteligentes.

No entanto, a interconexão de sistemas embarcados com a IoT também apresenta desafios, como a segurança dos dados, a privacidade, a escalabilidade e a interoperabilidade entre diferentes dispositivos e protocolos de comunicação. Esses desafios exigem soluções e abordagens adequadas para garantir a integridade, a confidencialidade e a disponibilidade dos dados transmitidos e armazenados.

O avanço contínuo nesse campo promete transformar a maneira como interagimos com o mundo físico, impulsionando a conectividade e a inteligência dos dispositivos, e abrindo caminho para um futuro cada vez mais interconectado.

2. ARQUITETURA

Vamos explorar a arquitetura da Internet das Coisas, seus componentes, ferramentas e protocolos que compõem essa tecnologia.

Para isso, vamos utilizar um exemplo prático para entendermos os passos envolvidos na criação de uma solução, desde a identificação do problema até a prototipação e a criação final.

2.1. Identificando o problema

O primeiro passo para a aplicação de IoT, é identificar o problema a ser resolvido, e a partir desta identificação, devemos pensar em como coletar dados que ajudem na tomada de decisão para sua resolução, como iremos trafegar e armazenar os dados coletados, como trabalhar para transformar este dado em informação, e fazer a tomada de decisão.

2.2. Exemplos de plataformas

Existem diversos tipos de plataformas que podem ser utilizadas para a coleta dos dados pois possuem conectividade com diversos tipos de sensores. Abaixo exemplos de algumas delas.

- Arduino - É uma plataforma de prototipagem que possui um microcontrolador onde podemos conectar os sensores. Desenvolvimento feito em C/C++;
- MCU – É um microcontrolador embarcado que possui chip único, de sistema operacional real time. Muito utilizado em soluções para uso médico ou militar, pois atua em sistemas de alta confiabilidade, de missão crítica;
- Raspberry Pi – É uma versão pocket do computador como conhecemos. Possui processador, memória, entradas e saídas, e possui grande capacidade para executar software. Pode executar sistemas operacionais convencionais, o que facilita a execução em diversas linguagens de programação;
- Smartphones – Sim, muitas vezes não percebemos, mas temos em nossas mãos a todo momento uma grande ferramenta para fazer a coleta de dados. Nossos telefones atualmente são mais poderosos que muitos computadores, com sensores para vários tipos de atividades, como GPS, giroscópio, sensor de luminosidade, temperatura ambiente, acelerômetro entre outros.

2.3. O protocolo de comunicação

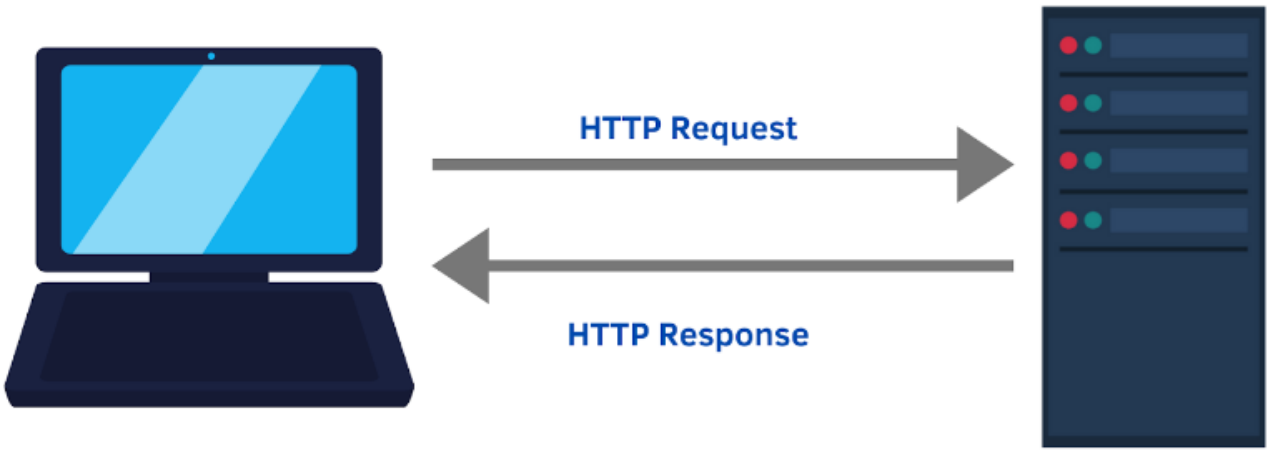
No início dos anos 90, a IBM criou o protocolo MQTT (Message Queue Telemetry Transport) que é um protocolo M2M (machine to machine) com foco específico para IoT. MQTT é um protocolo cliente servidor, onde o cliente pode tanto se inscrever no servidor para receber informações de determinado assunto que o interesse, quanto postar informações para que o servidor notifique aos inscritos naquele assunto específico. Ele é baseado no paradigma Publish-subscribe.

➔ Publish-subscribe

Para efeito de comparação e entendimento sobre o paradigma Publish-subscribe, vamos compará-lo a um velho conhecido nosso: O paradigma Request-response, representado pelo protocolo HTTP.

Em Request-response, a comunicação se dá de forma síncrona, onde o cliente envia ao servidor uma requisição, e fica aguardando sua resposta. A conexão entre os dois é feita de forma temporária, apenas durante o período da troca de mensagens.

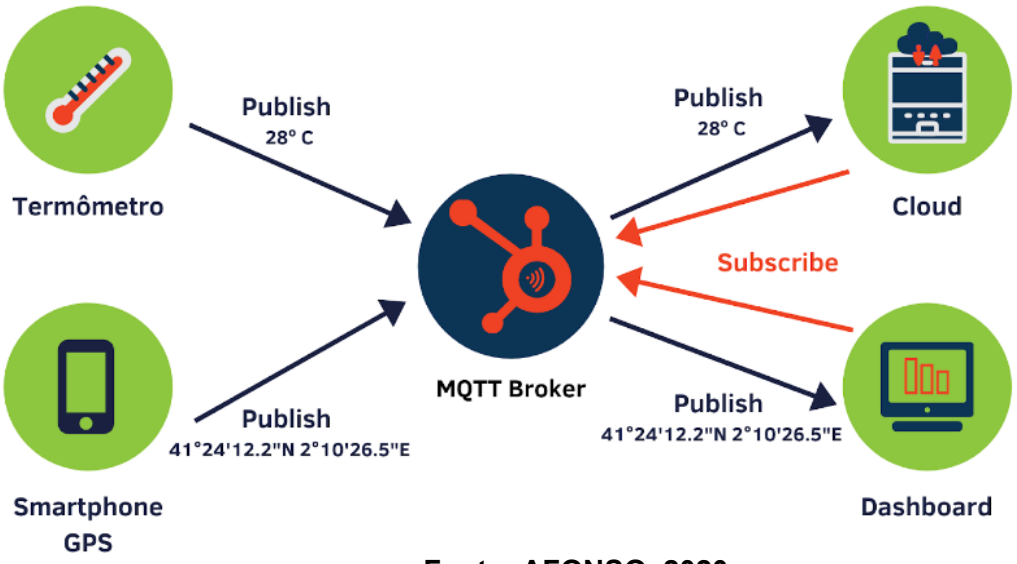
Figura 2 - Comunicação em Request-response.



Fonte: AFONSO, 2020.

Já no paradigma Publish-subscribe a comunicação se dá de forma assíncrona, onde as mensagens são recebidas e distribuídas aos respectivos destinatários através de um broker. O broker é um moderador que recebe as mensagens, e as entrega à todos que estão inscritos para recebê-la.

Figura 3 - Comunicação em Publish-subscribe.



Fonte: AFONSO, 2020.

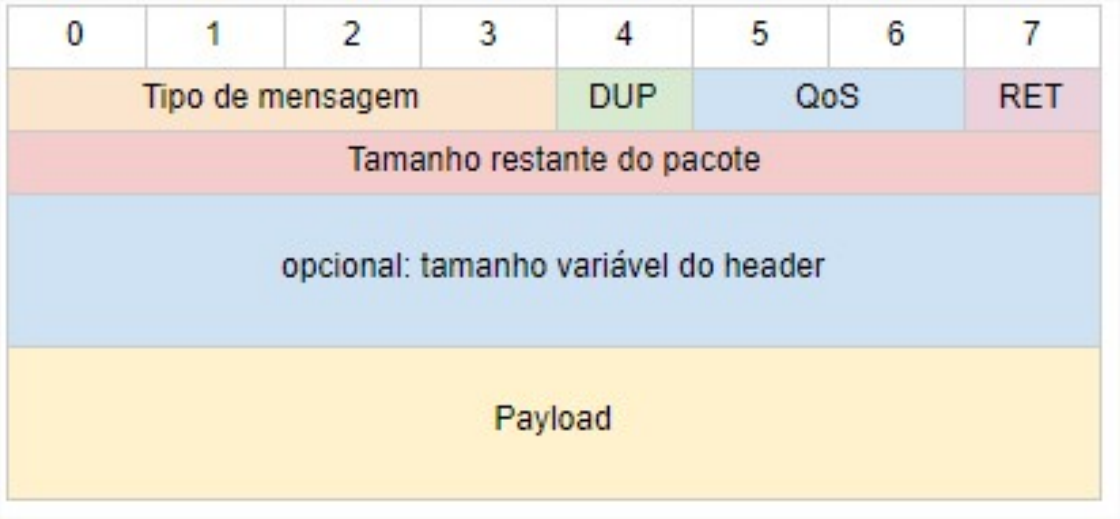
➔ MQTT

O protocolo MQTT, apesar de trazer a palavra queue no nome, não trabalha com um sistema de filas, e sim com um sistema de tópicos. Os tópicos são estruturas hierarquicamente criadas para tratar determinado assunto. Os clientes têm liberdade de efetuar a criação de um novo tópico, publicar mensagens em um tópico, e de se subscrever em um tópico.

Para efetuar a transmissão de dados, o MQTT faz a utilização de outro protocolo, o TCP. A mensagem MQTT segue um padrão, onde existe um header com uma quantidade específica de bytes que pode variar entre 2 e 5. Apenas o primeiro destes bytes é obrigatório.

Dentro dele, as informações são divididas da seguinte forma: os 4 primeiros bits referem-se ao tipo de mensagem. o 5º bit refere-se ao indicador de mensagem duplicada, os dois bits seguintes para identificar o QoS (quality of service) do pacote, e o último bit indica se a mensagem deve ser retida ou não, para que ao entrar um novo subscriber, ele receba a última mensagem publicada no tópico. Os próximos 4 bytes irão definir o tamanho do restante do pacote, variando de 0 a 268 335 455 bits.

Figura 4 - Representação gráfica da mensagem MQTT.



Fonte: AFONSO, 2020.

Figura 5 - Tipos de mensagem MQTT.

Valor	Nome	Direção	Descrição
0	Reservado	Proibido	Reservado
1	CONNECT	Cliente para Servidor	Requisição do cliente para conectar ao servidor
2	CONNACK	Servidor para Cliente	Reconhecimento da conexão
3	PUBLISH	Cliente para Servidor ou Servidor para cliente	Publicar mensagem
4	PUBACK	Cliente para Servidor ou Servidor para cliente	Reconhecimento da publicação
5	PUBREC	Publicação recebida	Publicação recebida(parte 2 do QoS=1)
6	PUBREL	Cliente para Servidor ou Servidor para cliente	Publicação lançada(parte 2 do QoS=2)
7	PUBCOMP	Cliente para Servidor ou Servidor para cliente	Publicação completa(parte 3 do QoS=2)
8	SUBSCRIBE	Cliente para Servidor	Pedido de inscrição
9	SUBACK	Servidor para cliente	Reconhecimento de inscrição
10	UNSUBSCRIBE	Cliente para Servidor	Pedido de desinscrição
11	UNSUBACK	Servidor para cliente	Reconhecimento de desinscrição
12	PINGREQ	Requisição	Requisição PING
13	PINGRESP	Servidor para cliente	Resposta PING
14	DISCONNECT	Cliente para Servidor	Cliente está desconectado
15	Reservado	Proibido	Reservado

Fonte: AFONSO, 2020.

➔ QoS

O QoS, ou Quality of Service é dividido em 3 categorias, e devem ser informados no header de cada mensagem. São eles:

- **QoS 0 (No máximo uma vez)** - Conhecido como fire and forget (atirar e esquecer), nesse QoS a mensagem é enviada apenas uma vez e não haverá passos seguintes, dessa forma a mensagem não será armazenada, nem haverá um feedback para saber se ela chegou ao destinatário. Esse modo de transferência é o mais rápido, porém o menos seguro já que a mensagem será perdida caso o envio falhe ou o cliente esteja desconectado;

- **QoS 1 (Pelo menos uma vez) -** Nesse modo de transferência, a mensagem é entregue pelo menos uma vez, havendo uma espera da recepção de feedback da entrega da mensagem, o chamado Puback. Não recebendo o Puback, a mensagem continuará sendo enviada até que haja o feedback. Nesse QoS pode acontecer da mensagem ser enviada diversas vezes e ser processada diversas vezes. Para que haja o envio da mensagem mais de uma vez, a mensagem precisa ser armazenada. Ela será excluída do receptor após ter recebido o feedback de confirmação do envio;

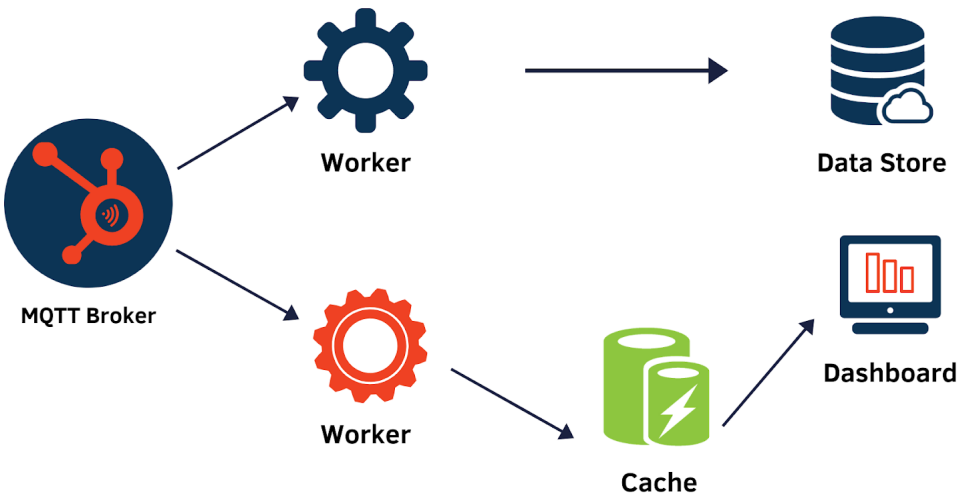
- **QoS 2 (Exatamente uma vez) -** Nesse modo de transferência, a mensagem é entregue exatamente uma vez, necessitando que seja armazenada localmente no emissor e no receptor até que seja processada. Para garantir a segurança desse QoS, é necessário o envio de 2 pares de request-response(chamado de four-part handshake), onde temos o envio da mensagem(Publish), a resposta de recepção (Pubrec), o aviso do recebimento do Pubrec (Pubrel) e a confirmação de que o processo foi concluído e pode ser feita a exclusão (Pubcomp). Após o recebimento do Pubrel, o receptor pode excluir a mensagem e ao publicador receber o Pubcomp ele poderá excluir a mensagem.

2.4. A nuvem

A escalabilidade oferecida pela nuvem se torna uma grande aliada para aplicações de IoT em larga escala. Imagine aplicativos que capturam a posição geográfica de milhares, talvez milhões de veículos simultaneamente, utilizando GPS. Assim como no ambiente local, precisamos receber e processar os dados coletados.Isso é feito por meio de workers, que são serviços na nuvem que se inscrevem para receber as informações do nosso intermediário (broker).

No exemplo abaixo, temos dois workers recebendo informações do mesmo tópico e trabalhando com elas de maneiras diferentes. O primeiro worker recebe dados sobre o uso de energia e temperatura ambiente e os armazena em um banco de dados. O segundo worker recebe os mesmos dados, mas os envia para uma área de cache onde apenas a última informação é relevante. Esses dados podem ser acessados por um serviço web ou mobile, que disponibiliza o status atual dos dispositivos monitorados.

Figura 6 - Gerenciando dados de IoT com workers: exemplo de armazenamento e processamento.



Fonte: AFONSO, 2020.

3. TECNOLOGIAS

➔ CONECTIVIDADE

A conectividade desempenha um papel fundamental nos sistemas embarcados conectados à Internet das Coisas (IoT), permitindo a comunicação entre os dispositivos e a transmissão de dados. Algumas das principais tecnologias de comunicação utilizadas na IoT incluem:

- **Wi-Fi:** Tecnologia amplamente utilizada em redes domésticas e empresariais. Oferece alta velocidade de transmissão de dados, mas consome mais energia;
- **Bluetooth:** Tecnologia de curto alcance adequada para aplicações que requerem comunicação entre dispositivos próximos. É eficiente em termos de energia e é amplamente utilizada em dispositivos portáteis;
- **Zigbee:** Protocolo de comunicação sem fio de baixo consumo de energia, projetado para aplicações de controle e monitoramento em redes de sensores sem fio;
- **LoRa (Long Range):** Tecnologia de comunicação de longo alcance que permite a transmissão de dados em grandes distâncias com baixo consumo de energia. É adequada para aplicações em áreas rurais e urbanas com baixa densidade de dispositivos.

Além disso, existem diversos **protocolos de comunicação** utilizados na IoT, como MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol) e HTTP (Hypertext Transfer Protocol), que são projetados para permitir a troca de dados entre dispositivos e sistemas.

Cada tecnologia possui vantagens e limitações. É importante considerar o alcance, a largura de banda, o consumo de energia e a segurança ao escolher a tecnologia de conectividade mais adequada para um sistema embarcado na IoT.

➔ SOFTWARE

O desenvolvimento de software é essencial para o funcionamento dos sistemas embarcados conectados à IoT. Alguns aspectos relevantes nesse contexto incluem:

- **Plataformas de desenvolvimento:** Arduino, Raspberry Pi, ESP32, entre outras. Essas plataformas oferecem um ambiente de desenvolvimento e bibliotecas que facilitam a programação dos dispositivos;
- **Linguagens de programação:** Linguagens como C/C + +, Python e Java são amplamente utilizadas no desenvolvimento de software para sistemas embarcados;
- **Frameworks e bibliotecas:** Existem frameworks e bibliotecas específicas para o desenvolvimento de aplicações na IoT, que fornecem funcionalidades adicionais e simplificam o processo de programação. Exemplos incluem o Node-RED, que permite a criação de fluxos de dados visualmente, e o TensorFlow Lite, que facilita a implementação de modelos de aprendizado de máquina em sistemas embarcados;
- **Sistemas operacionais embarcados:** Os sistemas operacionais embarcados, como o FreeRTOS, o Contiki e o TinyOS, são projetados para executar em dispositivos com recursos limitados. Eles oferecem recursos de gerenciamento de tarefas, escalonamento, comunicação e acesso a periféricos.

A escolha da plataforma de desenvolvimento, linguagem de programação, frameworks e bibliotecas depende das necessidades e requisitos do sistema embarcado na IoT. É importante considerar a eficiência de recursos, facilidade de desenvolvimento e suporte à conectividade e segurança.

➔ **GERENCIAMENTO DE ENERGIA**

O gerenciamento de energia é um aspecto crítico em sistemas embarcados conectados à IoT, especialmente quando se trata de dispositivos alimentados por bateria. Algumas considerações relevantes incluem:

- **Otimização do consumo de energia:** É importante projetar sistemas que sejam eficientes em termos de energia. Isso pode incluir o uso de sensores de baixo consumo de energia, algoritmos de economia de energia e estratégias de desligamento de componentes quando não estão em uso;

- **Estratégias de gerenciamento de energia:** O gerenciamento eficiente da energia pode ser alcançado por meio de técnicas como hibernação, escalonamento dinâmico de frequência e controle inteligente de energia com base na carga de trabalho;

- **Eficiência energética na IoT:** Os sistemas embarcados na IoT podem se beneficiar da coleta e análise de dados relacionados ao consumo de energia. Isso permite identificar padrões de uso, detectar desperdícios e otimizar o consumo em tempo real.

O gerenciamento de energia eficaz é essencial para garantir a vida útil da bateria, a confiabilidade do sistema e a redução do impacto ambiental.

4. SEGURANÇA E PRIVACIDADE

A segurança e a privacidade são preocupações fundamentais nos sistemas embarcados conectados à Internet das Coisas devido à natureza sensível dos dados coletados e trocados. Aqui estão algumas considerações importantes em relação à segurança e privacidade em sistemas IoT:

- **Autenticação e Autorização:** É crucial garantir a autenticidade dos dispositivos IoT e dos usuários que interagem com esses dispositivos. A autenticação forte, como o uso de chaves criptográficas, ajuda a verificar a identidade dos dispositivos e usuários antes de permitir o acesso aos recursos. A autorização adequada deve ser implementada para garantir que somente dispositivos e usuários autorizados possam acessar e controlar os dispositivos e os dados coletados;

- **Criptografia de Dados:** A comunicação entre os dispositivos IoT, os gateways e a nuvem deve ser criptografada para garantir a confidencialidade dos dados transmitidos. A criptografia assegura que os dados não possam ser lidos ou modificados por terceiros não autorizados durante o transporte. Algoritmos criptográficos robustos e práticas de gerenciamento de chaves adequadas devem ser empregados;

- **Atualizações de Segurança:** É essencial manter os dispositivos IoT atualizados com as últimas correções de segurança. Isso inclui a aplicação regular de patches e atualizações de firmware para corrigir vulnerabilidades conhecidas. Os fabricantes devem fornecer mecanismos de atualização seguros e simples para garantir que os dispositivos IoT permaneçam protegidos contra ameaças emergentes;

- **Monitoramento e Detecção de Anomalias:** Sistemas de monitoramento contínuo devem ser implementados para detectar atividades suspeitas ou comportamento anômalo nos dispositivos IoT. Isso pode envolver a análise de padrões de tráfego, detecção de intrusões e análise de dados em tempo real para identificar possíveis ameaças e violações de segurança;

- **Privacidade dos Dados:** A proteção da privacidade dos dados coletados pelos dispositivos IoT é fundamental. As informações pessoais dos usuários devem ser tratadas com cuidado e de acordo com as regulamentações de privacidade aplicáveis, como o Regulamento Geral de Proteção de Dados (GDPR) na União Europeia. As práticas de coleta,

armazenamento e uso de dados devem ser transparentes e informadas aos usuários, com opções claras de consentimento e controle sobre o compartilhamento e uso dos dados;

- **Segurança Física:** Além das medidas técnicas, a segurança física dos dispositivos IoT também é importante. Os dispositivos devem ser protegidos contra acesso não autorizado, seja fisicamente ou por meio de proteções de rede, como firewalls e sistemas de detecção de intrusões.

5. DESAFIOS E TENDÊNCIAS FUTURAS

- **Segurança:** A segurança continua sendo um dos desafios mais críticos para os sistemas IoT. Com o aumento da conectividade e do número de dispositivos interconectados, a superfície de ataque também aumenta. Garantir a segurança desses dispositivos, das redes e dos dados coletados é essencial para evitar violações de segurança e proteger a privacidade dos usuários;

- **Escalabilidade:** Com o crescimento exponencial do número de dispositivos IoT, a escalabilidade se torna um desafio importante. Os sistemas precisam ser projetados para lidar com grandes volumes de dados, comunicação em tempo real e processamento de alto desempenho, garantindo ao mesmo tempo a eficiência e a capacidade de resposta;

- **Interoperabilidade:** A falta de padronização e a diversidade de tecnologias e protocolos podem dificultar a integração de dispositivos de diferentes fabricantes. Tendências futuras incluem o desenvolvimento de padrões e protocolos comuns para facilitar a interoperabilidade entre dispositivos IoT;

- **Gerenciamento de Dados:** A quantidade de dados gerada pelos dispositivos IoT é enorme e crescente. Lidar com o gerenciamento,

armazenamento, processamento e análise desses dados de forma eficiente e escalável é um desafio significativo. Técnicas avançadas de análise de dados, como aprendizado de máquina e inteligência artificial, serão cada vez mais importantes para extrair informações úteis dos dados coletados;

- **Eficiência Energética:** Muitos dispositivos IoT são alimentados por baterias ou fontes de energia limitadas. Garantir a eficiência energética dos dispositivos é essencial para prolongar a vida útil da bateria e reduzir a necessidade de manutenção frequente. O desenvolvimento de técnicas de baixo consumo de energia e o uso de técnicas de otimização energética serão tendências futuras nesse campo;

- **Integração com a Inteligência Artificial:** A combinação de IoT e inteligência artificial (IA) abre caminho para inúmeras aplicações inovadoras. A capacidade de coletar grandes volumes de dados em tempo real dos dispositivos IoT pode alimentar algoritmos de IA para análises avançadas, previsões e tomada de decisões automatizadas. A IA desempenhará um papel crucial no avanço dos sistemas IoT e na geração de insights valiosos a partir dos dados coletados.

6. CONCLUSÃO

A interconexão dos sistemas embarcados à IoT cria um ecossistema de dispositivos inteligentes que colaboram entre si, coletam dados em tempo real e permitem a tomada de decisões mais informadas e precisas. Isso resulta em maior eficiência operacional, redução de custos, maior segurança e melhores experiências para os usuários.

No entanto, essa interconectividade também traz desafios significativos, como a segurança dos dados e a privacidade. À medida que mais dispositivos estão conectados à IoT, surgem preocupações sobre a proteção de informações sensíveis e a prevenção de violações de segurança.

Portanto, medidas rigorosas de segurança cibernética e políticas de privacidade devem ser implementadas para mitigar esses riscos.

Além disso, a evolução contínua da IoT requer uma abordagem adaptativa e flexível. É essencial acompanhar as tendências tecnológicas, como a computação em nuvem, a inteligência artificial, a análise de dados em tempo real e a edge computing, para aproveitar ao máximo o potencial da IoT.

Conforme a IoT se expande, a colaboração entre empresas, pesquisadores e governos desempenha um papel fundamental na superação dos desafios e no impulsionamento da inovação. A troca de conhecimentos, melhores práticas e padrões de segurança contribui para o desenvolvimento sustentável e seguro dos sistemas embarcados conectados à IoT.

Em suma, os sistemas embarcados conectados à IoT têm o poder de transformar a maneira como vivemos e trabalhamos. Com a combinação certa de conectividade, software e gerenciamento de energia, podemos criar soluções inteligentes que melhoram a eficiência, a segurança e a qualidade de vida. À medida que avançamos nessa era digital, é crucial explorar o potencial da IoT de forma responsável, garantindo a segurança dos dados e a privacidade, e impulsionando a inovação para criar um futuro mais conectado e inteligente.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- Antonopoulos, A., & Gillam, L. (2017). **The Internet of Things: Living in a Connected World.** Syngress.
- Barros, E., & Farias, M. C. (2017). **Internet das Coisas (IoT): Conceitos, Arquiteturas e Aplicações.** Novatec Editora.
- Gu, Y., Wang, Z., Khandelwal, A., & Wu, W. (2018). **IoT Based Intelligent Embedded Systems: Paradigms, Technologies, and Applications.** Springer.

- Liu, J., & Jiang, H. (2019). **Internet of Things: Advances, Challenges, and Opportunities.** CRC Press.
- Roisenberg, M., Ferreira, D., & Fagundes, L. (2017). **Internet das Coisas: Aplicações e Desafios.** Editora UFSM.
- Silva, R., & Silva, R. (2019). **Internet of Things: Principles and Paradigms.** Wiley.
- Vermesan, O., & Friess, P. (Eds.). (2014). **Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems.** River Publishers.