UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DO TOCANTINS - CAMETÁ
FACULDADE DE SISTEMAS DE INFORMAÇÃO

DANIEL CARDOSO ESTUMANO

COMPUTAÇÃO EM NÉVOA: OS PRINCIPAIS BENECIFÍCIOS DA ARQUITETURA PARA APLICAÇÕES IOT SENSÍVEIS À LATÊNCIA

Cametá/Pará
2026

UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DO TOCANTINS - CAMETÁ
FACULDADE DE SISTEMAS DE INFORMAÇÃO

COMPUTAÇÃO EM NÉVOA: OS PRINCIPAIS BENECIFÍCIOS DA ARQUITETURA PARA APLICAÇÕES IOT SENSÍVEIS À LATÊNCIA

Trabalho de Conclusão de Curso apresentado ao curso de sistemas de informação, da UNIVERSIDADE FEDERAL DO PARÁ, como requisito parcial para a Obtenção do grau de Bacharel em sistemas de informação.

Orientador: Dr. Allan Barbosa Costa

Cametá/Pará
2026

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

E79c Estumano, Daniel Cardoso.
COMPUTAÇÃO EM NÉVOA : os principais benefícios da arquitetura para aplicações IoT à sensíveis a latência / Daniel Cardoso Estumano. — 2026.
43 f. : il. color.

Orientador(a): Prof. Dr. Allan Barbosa Costa
Trabalho de Curso (Graduação) - Universidade Federal do Pará, Campus Universitário de Cametá, Curso de Sistemas de Informação, Cametá, 2026.

1. Computação em Névoa. 2. IoT. 3. Latência. I. Título.

CDD 004.6782

DANIEL CARDOSO ESTUMANO

COMPUTAÇÃO EM NÉVOA: OS PRINCIPAIS BENECIFÍCIOS DA ARQUITETURA PARA APLICAÇÕES IOT SENSÍVEIS À LATÊNCIA

Trabalho de Conclusão de Curso apresentado ao curso de sistemas de informação, da UNIVERSIDADE FEDERAL DO PARÁ, como requisito parcial para a Obtenção do grau de Bacharel em sistemas de informação.

LOCAL: Cametá, 14 de julho de 2026.

BANCA EXAMINADORA

Prof. Dr. Allan Barbosa Costa (Orientador)
Universidade Federal do Pará

Prof. Me. Leonardo Nunes Gonçalves (Membro)
Universidade Federal do Pará

Prof. Keventon Rian Guimarães Gonçalves (Membro)
Universidade Federal do Pará

Agredecimentos

Agradeço, primeiramente, a Deus pela força, saúde e sabedoria que me concedeu em todos os momentos de desafio durante esta jornada.

Aos meus pais, pelo amor incondicional, pelo suporte constante e por serem a base de tudo o que sou e conquisto. Obrigado por acreditarem em mim e por incentivarem meus sonhos.

À minha família, pelos conselhos dados e apoio moral, por me incentivarem aos estudos.

Ao meu orientador, pela paciência e pelas orientações preciosas; agradeço por compartilhar seu vasto conhecimento e por sempre incentivar a todos ao estudo durante as aulas ministradas por ele.

Aos meus amigos, por me proporcionarem bons momentos e apoio nas dificuldades enfrentadas durante esta jornada.

Agradeço à Universidade Federal do Pará pela oportunidade.

“É preciso estudar muito para saber um pouco”

Montesquieu

RESUMO

A Internet das Coisas mudou a forma como os dispositivos falam, pegam e trocam dados em tempo real. No entanto, alguns aplicativos essenciais não podem contar tanto com a computação em nuvem e precisam de resposta rápida. A necessidade de resposta rápida cria latência que a computação em nuvem tradicional não consegue atender bem. Por isso, a Computação em Névoa está crescendo, porque coloca o processamento mais perto de onde os dados são gerados e ajuda a reduzir a latência. O objetivo deste trabalho é descobrir os benefícios básicos da computação em névoa em aplicações de IoT que são sensíveis à latência. Para isso, aplicamos uma metodologia descritiva e bibliográfica. Os resultados mostram que a computação em névoa tem um papel essencial. A computação em névoa ajuda a reduzir a latência, a usar menos largura de banda, a aumentar a segurança e a melhorar o desempenho do sistema.

Palavras-chave: IoT; Computação em Névoa; Latência.

ABSTRACT

Internet of Things (IoT) has transformed the way devices communicate, collect, and exchange data in real time. However, some critical applications cannot rely solely on cloud computing and require rapid response times. This need for fast response creates latency issues that traditional cloud computing cannot efficiently address. Therefore, Fog Computing has been gaining importance because it places data processing closer to where the data is generated, helping to reduce latency. The objective of this study is to identify the main benefits of Fog Computing in latency-sensitive IoT applications. To achieve this, a descriptive and bibliographic methodology was applied. The results show that Fog Computing plays an essential role by reducing latency, minimizing bandwidth usage, increasing security, and improving system performance.

Keywords: IoT, Fog Computing, and Latency.

LISTA DE ABREVIATURAS E SIGLAS

ACRN Hipervisor de código aberto leve e de alta segurança

ACK Pacote de confirmação de recebimento

Botnets Rede de computadores e dispositivos conectados infectados por códigos maliciosos

CPS Sistemas Ciber-Físicos

CoAP Protocolo de transporte lógico para aparelhos com limitações críticas

DaaS Data as a Service

DDoS Ataques de negação de serviço

Endpoints Qualquer dispositivo físico conectado à rede

FPS Quadros por Segundo

HMI interface homem-máquina

IBSG *Internet Business Solutions Group*

IoT Internet das Coisas

JITTER Variação do Atraso de Pacotes

Kubernetes Sistema de orquestração de código aberto

M2M Comunicação Máquina-a-Máquina

MANO Gerenciamento e Orquestração Autônomos

MQTT Message Queuing Telemetry Transport

NFV Virtualização de Funções de Rede

NON Mensagens estabelecidas como não confiáveis

QoE Qualidade de Experiência

QoS Qualidade de Serviço

RTT Tempo de ida e volta (*Round-Trip Time*)

SOA Arquitetura Orientada a Serviço

SWaP-C Tamanho, Peso, Consumo de Energia e Custo

TCP Protocolo de Controle de Transmissão

UDP Protocolo de transporte mais leve e sem estado

WSNs Redes de sensores sem fios

YOLOv3 Modelo de deteção de objetos da família YOLO (You Only Look Once)

LISTA DE EQUAÇÕES

Lfila	Tempo de enfileiramento nos roteadores
Lproc	Tempo de processamento do pacote nos nós/dispositivos
Lprop	Tempo de propagação física do sinal no meio de transmissão
Ltotal	Tempo total de atraso na rede
Ltrans	Tempo de transmissão do pacote para o canal (largura de banda)

SUMÁRIO

1 INTRODUÇÃO

A tecnologia da informação e comunicação mudou muito a forma como os usuários usa dispositivos, sistemas e conversa com outras pessoas. Por isso, a Internet das Coisas (IoT) virou um dos modelos mais usados para criar um mundo conectado, onde bilhões de dispositivos podem receber, analisar e trocar informações na hora. Luigi Atzori, Antonio Iera e Giacomo Morabito (2010), dizem que a IoT é um conceito em que objetos físicos têm identidade digital e conseguem se comunicar. Observa-se que a automação avança rápido, mas ainda é difícil imaginar que o escopo da automação e da inteligência dos sistemas se amplie ainda mais sob a perspectiva. A criação de dados cresce muito rápido, e essa velocidade coloca uma carga pesada na infraestrutura poderosa e eficiente que precisa armazenar e processar os dados. Esse crescimento exponencial e pressão sobre os sistemas ficam claros ao analisarmos a evolução dos dados de 2003 a 2020. Em 2003, com a Terra abrigando 6,3 bilhões de pessoas, apenas 500 milhões de aparelhos estavam ligados à rede, o que resultava em uma média de 0,08 aparelho por indivíduo. A situação se transformou significativamente: em 2010, as conexões dispararam para 12,5 bilhões (1,8 aparelho por pessoa), e em 2015, esse número duplicou para 25 bilhões, aumentando a média para 3,47. Por fim, as previsões para 2020 demonstraram o impressionante total de 50 bilhões de equipamentos conectados para uma população de 7,6 bilhões, chegando a 6,58 dispositivos por indivíduo (CISCO, 2011).

Conforme as projeções levantadas pela Cisco (2011), os 50 bilhões de aparelhos ligados em 2020 confirmam um quadro onde o enorme volume de dados sobrecarrega as formas usuais de infraestrutura. Levar toda essa massa para ser tratada apenas em centros de servidores na nuvem (Cloud Computing) causa problemas e lentidão na comunicação. É por causa desse excesso que aparece a exigência de um novo jeito de organizar. Para permitir o funcionamento de sistemas de IoT que precisam de retornos imediatos – ou seja, que não toleram atrasos e precisam do ambiente local.

A Computação em Névoa (Fog Computing) tem usado a escalabilidade, a elasticidade e o armazenamento distribuído para atender a essa demanda. Mas Chengliang Yi, Zhi Li e Qun Li (2015), mostram que o modelo tem limites quando a gente precisa de baixa latência, Isso ocorre justamente porque, quando o sistema depende da infraestrutura da nuvem tradicional, o envio de dados para servidores centralizados e distantes gera atrasos incompatíveis com aplicações em tempo real.

O tempo de resposta é fundamental para soluções críticas, como veículos autônomos, monitoramento de pacientes e automação industrial. Pouco atraso ou nenhum atraso não só diminui o desempenho do sistema, mas também coloca em risco a segurança dos usuários. A latência aqui não é apenas uma medida de desempenho; a latência é necessária por si mesma para que as aplicações funcionem bem. Diante dessas limitações, a computação em névoa pode ser a solução. A computação em névoa descentraliza o processamento de dados e traz o processamento de dados mais perto dos dispositivos finais. Flavio Bonomi et al. (2012), explicam que a computação em névoa funciona como uma camada entre a nuvem e os dispositivos IoT. A computação em névoa faz o processamento ficar mais rápido, diminui a latência e usa a rede de forma mais eficiente.

1.1 Objetivos

O objetivo deste trabalho é responder à pergunta de pesquisa: como a computação em névoa pode reduzir a latência nas aplicações IoT? Defino o problema de pesquisa como a necessidade de reduzir a latência nos sistemas de Internet das Coisas (IoT), considerando as limitações das arquiteturas centralizadas. Notamos que a computação em névoa pode ser decisiva para reduzir a latência, pois coloca os cálculos mais perto das portas de entrada e saída de dados.

O objetivo geral deste trabalho é investigar as vantagens da computação em névoa nas aplicações de Internet das Coisas (IoT) que são sensíveis à latência. Os objetivos estabelecidos para alcançar esse propósito foram divididos em quatro etapas interligadas: primeiro, explicar de forma simples a computação em névoa; segundo, descrever as aplicações da Internet das Coisas; terceiro, descrever e justificar os principais benefícios da computação em névoa; quarto e, finalmente, definir o papel do tempo de resposta rápido em sistemas distribuídos.

1.2 Contribuições da Pesquisa

- A pesquisa se concentra em explorar e identificar as vantagens da computação em névoa (*Fog Computing*) para cenários de Internet das Coisas (IoT) onde o tempo de resposta é crucial.
- O trabalho expõe como a computação em névoa distribui o processamento e o aproxima da origem dos dados, servindo como um complemento à computação em nuvem convencional.

- A análise mostra que a aplicação dessa abordagem soluciona desafios de lentidão em sistemas vitais, como os empregados em saúde, manufatura, cidades inteligentes e na indústria automotiva autônoma.
- O estudo demostra que a implantação da computação em névoa reduz a necessidade de banda larga, aprimora a performance geral, assegura maior capacidade de expansão e eleva a segurança e a robustez das operações.

2 METODOLOGIA

Neste trabalho, apresento uma pesquisa de descrição com abordagem qualitativa e base bibliográfica. A pesquisa de descrição tem como objetivo analisar, interpretar e descrever os fenômenos. Neste caso, a pesquisa de descrição analisa os conceitos, as características e os benefícios da computação em névoa no contexto da Internet das Coisas (IoT). Essa abordagem serve bem a estudos que buscam entender os fenômenos tecnológicos emergentes e as aplicações da computação em névoa em diferentes áreas.

A metodologia que adotada se baseia em um levantamento bibliográfico. Para o levantamento bibliográfico usando fontes secundárias de caráter científico e acadêmico, como artigos publicados em revistas, livros especializados e documentos técnicos de organizações conhecidas na área de tecnologia. Entre os principais referenciais teóricos que utilizado, destacam-se os estudos de Flavio Bonomi et al. No ano (2012), os autores introduzem o conceito de computação em névoa como uma continuação da computação em nuvem. Mahadev Satyanarayanan (2017), discute a importância do processamento distribuído próximo à borda da rede para aplicações sensíveis à latência. Além disso, o OpenFog Consortium (2017), publica documentos técnicos que estabelecem diretrizes e modelos de referência para sistemas baseados em computação em névoa.

A coleta de dados aconteceu quando selecionei e analisei as publicações importantes sobre computação em névoa, computação em nuvem, Internet das Coisas e latência em sistemas distribuídos. Dando prioridade a materiais que explicam tanto os conceitos quanto as aplicações práticas dessas tecnologias, assim entende-se o tema de forma completa.

A análise dos dados usou uma abordagem qualitativa. Interpretando as informações que foram obtidas nas fontes consultadas. O que exigiu identificar os conceitos principais, as vantagens, os desafios e as aplicações da computação em névoa.

A análise dos dados também mostrou como a computação em névoa se relaciona com as demandas dos sistemas IoT. Flavio Bonomi et al. (2012), apontaram que a descentralização do processamento é um ponto importante para a eficiência desses sistemas. considerando esse ponto ao avaliar os benefícios apresentados ao longo do trabalho.

Além do mais as pesquisas foram ligando as diferentes teorias, para ter uma visão completa do assunto. Com essa análise, conclui que a computação em névoa funciona como uma solução que complementa a computação em nuvem, como também apontou Mahadev Satyanarayanan (2017). Essa combinação ajuda muito em situações que precisam de baixa latência e alta disponibilidade.

Por fim, a abordagem qualitativa permitiu descrever os benefícios da computação em névoa. Ajudando a entender a relevância da computação em névoa no cenário atual das tecnologias emergentes. Também mostrou como a computação em névoa pode ser útil hoje. A metodologia usada mostrou que era possível atingir os objetivos que foram propostos. E ainda trouxe uma base teórica consistente para a análise feita neste trabalho.

3 FUNDAMENTOS DA COMPUTAÇÃO EM NÉVOA

A computação em névoa (*Fog Computing*) é um jeito de levar os serviços da computação em nuvem mais perto dos dispositivos finais. Reduzindo tempo de resposta e melhorando a velocidade das aplicações distribuídas. Segundo Bonomi, a computação em névoa surge como uma extensão natural da computação em nuvem, posicionando recursos computacionais na borda da rede para atender às demandas de aplicações sensíveis ao tempo, fazendo o pré-processamento, armazenamento e análise de dados mais próximos da sua origem. Assim, reduz a latência e melhora o desempenho geral dos sistemas distribuídos. (BONOMI et al., 2012).

Isso quer dizer que a computação em névoa, não está sendo usada como substituta da computação em nuvem; ela é simplesmente uma parte da nuvem. Assim, a computação em névoa traz recursos de computação para a borda da rede, e ao reduzir o tempo entre a geração e o processamento de dados — o que a torna muito importante em aplicações que são sensíveis ao tempo. No dia a dia, isso pode fazer com que tarefas que antes eram 100% remotas passem a ser feitas em parte aqui, no local, ou de forma intermediária. Essa mudança pode reduzir bastante a latência e deixar os sistemas mais rápidos e eficientes.

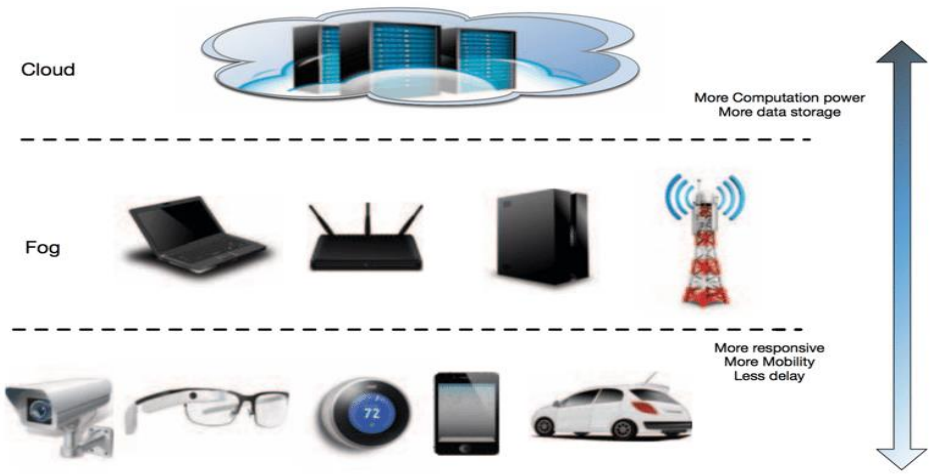
Além disso, o processamento, a análise e o armazenamento ficam mais perto da fonte de dados do que um nó que está muito longe. Por isso, reduz o consumo de rede e aumenta o funcionamento do sistema. Assim traz um grande avanço para a computação moderna. E também ajuda a criar aplicações que respondem mais rápido e que podem crescer conforme a necessidade. Essas aplicações atendem aos requisitos atuais.

Diferente das plataformas que rodam na nuvem, onde o processamento de muitos dados acontece em um único data center grande, a computação em névoa distribui o processamento pelas camadas da rede. Com a arquitetura descentralizada, pelo menos uma parte dos dados pode ser tratada localmente, e o tempo de resposta diminui. Mahadev Satyanarayanan (2017), afirma que é essencial trabalhar mais perto dos dispositivos finais quando o processamento é usado em novas tecnologias, como IoT (Internet das Coisas), veículos autônomos e aplicações de cidades inteligentes.

A arquitetura do sistema de computação em névoa tem uma estrutura em camadas. As camadas da arquitetura do sistema trabalham juntas para melhorar o processamento, a transmissão e o armazenamento dos dados de IoT que são criados no ambiente descentralizado. Percebe que a organização permite que a carga de computação seja distribuída de forma inteligente. Assim, a arquitetura do sistema de computação em névoa melhora o desempenho geral do sistema.

Na prática, a camada de dispositivos (IoT) está bem perto da fonte onde os dados são gerados, como mostra na figura 1.

Figura 1- Arquitetura em três camadas: dispositivos inteligentes e sensores, Névoa/cloudlets/MEC e a Computação em Nuvem



Fonte: Coutinho, Carneiro e Greve (2016), adaptado de Yi et al. (2015). Disponível em: https://www.researchgate.net. Acesso em: 10 abril. 2026.

A camada de dispositivos reúne os sensores, os atuadores e os dispositivos inteligentes. Os dispositivos coletam informações do ambiente físico, como temperatura, localização, batimentos cardíacos e movimentação. Os dispositivos têm, na maioria das vezes, recursos limitados de processamento e energia. Por isso, os dispositivos não conseguem fazer análises complexas no próprio local. Por consequente, a camada de dispositivos funciona como geradora e transmissora de dados brutos, enviando os dados para as camadas superiores da nuvem. Flavio Bonomi et al. (2012), explica que a variedade e a grande quantidade de dispositivos conectados exigem um modelo computacional distribuído. O modelo computacional distribuído ajuda a lidar com esse volume de informações de forma eficiente.

A camada de névoa (*fog layer*) fica no meio da arquitetura e tem um papel importante. A camada de névoa é feita de dispositivos que têm mais capacidade de cálculo que os dispositivos IoT, como gateways, roteadores, servidores locais e micro data centers (BONOMI et al., 2012). Essa camada de névoa faz o processamento dos dados mais próximo aos dispositivos IoT, filtrando, juntando, analisando rapidamente e tomando decisões na hora. O objetivo principal da camada de névoa é reduzir a latência e diminuir o volume de dados que vão para a nuvem (*Cloud)*.

Para que esse processamento de filtragem e redução de latência ocorram de forma eficiente, essa comada de névoa opera de forma organizada em camadas (*tiers*), da seguinte forma segundo Bonomi et al. (2012):

A primeira Camada de Névoa (nível mais Baixo e Rápido):

- **Velocidade e Ação**: Na linha de frente. Os coletores da névoa na borda recebem os dados gerados pelos sensores. Lidando com ciclos de proteção e controle que exigem repostas em tempo real, variando de milissegundos a frações de segundo.
- **Comunicação Máquina-a-Máquina (M2M)**: Esta camada é projetada especificamente para interações de máquinas comunicando-se umas com as outras (M2M) é a capacidade de aparelhos eletrônicos para "dialogarem" mutuamente, trocando dados e executando tarefas sem precisar de intervenção humana. O sistema processa os dados automaticamente e emite comandos de controle diretos para os atuadores (BONOMI et al., 2012, tradução nossa).

- **Filtragem de Dados:** Os nós aqui atuam como filtros: eles separam o que precisa ser consumido imediatamente de forma local e enviam apenas o restante dos dados para as camadas superiores.
- **Armazenamento:** Devido à necessidade de velocidade imediata, o armazenamento de dados nesta camada inferior e breve (temporário).

A Segunda e a Terceira Camadas da Névoa (Níveis Intermediários)

- **Mistura de Processos (M2M e HMI):** À medida que os dados sobem, as interações mudam. Estas camadas lidam não apenas com sistemas e processos M2M, mas também introduzem interações (HMI) interface homem-máquina funciona como um elo de ligação, que facilita o diálogo entre o usuário e máquina. Dessa forma, conseguimos compreender as operações em andamento e intervir quando for preciso, focadando em visualização e relatórios.
- **Escalas de Tempo Maiores:** As camadas intermediárias, ao invés de respostas imediatas na casa dos milissegundos, funcionam em períodos de tempo mais longos. Esses intervalos podem variar de segundos e minutos, utilizados em análises que ocorrem em tempo real, até períodos de dias, dedicados ao processamento de dados transacionais (BONOMI et al., 2012).
- **Escopo e Armazenamento**: Os autores estabelecem uma regra arquitetural clara: quanto mais alta a camada, maior é a sua cobertura geográfica e maior é a escala de tempo envolvida. Consequentemente, o armazenamento de dados aqui passa de breve para algo que dura muito, mas não definitivamente (BONOMI *et al.*, 2012).

Impacto em Redes de Sensores e Atuadores Sem Fio (WSANs). Em termos de rede e hardware, a arquitetura modifica as antigas Redes de Sensores Sem Fio (WSNs), antes de fluxo unidirecional, para Redes de Sensores e Atuadores (WSANs), agora com comunicação bidirecional. A inclusão de atuadores, capazes de executar ações físicas, converte o sistema em um "circuito fechado" (*closed-loop*). Isso, por sua vez, torna a diminuição da latência e o controle do (*jitter*), características da arquitetura de Névoa, essenciais para a estabilidade do sistema (BONOMI *et al.*, 2012).

A camada de nuvem (cloud) é a parte mais alta da arquitetura. Na realidade, a camada de nuvem cuida do armazenamento de longo prazo e do processamento pesado de dados. É muito útil. E também faz análises avançadas, como a mineração de dados e a execução dos algoritmos de aprendizado de máquina nos dados que entram e saem.

Oferecendo a capacidade computacional muito grande e a capacidade de crescer quase sem limites.

Por isso, é uma boa escolha para juntar as informações volumosas das camadas base. Devido à distância da nuvem, não é convencional usa-la para decisões que exijam reação urgente. Por isso, a nuvem funciona como parceira da névoa, recebendo apenas dados que já foram filtrados ou processados. O OpenFog Consortium (2017), explica como juntar a nuvem, a névoa e a borda em uma arquitetura modular que fica escalável, que consome menos energia, que é resiliente, e que organiza o processamento de acordo com os requisitos da aplicação. Percebe-se que essa solução dá ao usuário um jeito de ter um sistema bem melhor, sem forçar o usuário, os dispositivos, os sistemas ou as aplicações com pressões de rede que não são necessárias.

A arquitetura em camadas de computação em névoa organiza o fluxo dos dados e ainda tira o máximo proveito do poder computacional, usando os recursos computacionais que estão disponíveis. Cada nível do sistema faz o trabalho que cabe a cada nível, de acordo com a capacidade do nível e a distância da fonte dos dados. É notório que como consequência, só enviamos os dados relevantes para a nuvem e reduzimos o tráfego de rede. Essa redução ajuda o sistema a funcionar de maneira mais eficiente. Conforme destacado por (OpenFog Consortium, 2017). A computação em névoa melhora a escalabilidade e otimiza o uso de largura de banda ao distribuir inteligência ao longo da infraestrutura, permitindo que o processamento ocorra de forma descentralizada entre os dispositivos, a borda da rede e a nuvem, o que resulta em maior eficiência, redução de congestionamentos e melhor desempenho dos sistemas distribuídos em ambientes com grande volume de dispositivos conectados.” (OPENFOG CONSORTIUM, 2017).

Além disso, a computação em névoa oferece mais autonomia aos dispositivos, permitindo que os dispositivos continuem operando mesmo quando a conexão com a nuvem falha. Este ponto da computação em névoa é muito importante em ambientes críticos, onde a disponibilidade contínua dos sistemas é essencial.

3.1 Protocolos de Comunicação

A computação em névoa demonstra sua força na redução da latência e no aprimoramento do uso da largura de banda, não apenas por espalhar os pontos de processamento geograficamente e aproximar a computação da extremidade da rede, Para que o fluxo computacional distribuído funcione de forma integrada, é importante

estabelecer como os dados transitarão logicamente, ou seja, quais protocolos de comunicação e mensagens guiarão as interações entre os dispositivos (sensores e atuadores), os pontos intermediários da névoa e os servidores na nuvem. Em situações comuns da Internet das Coisas (IoT), os dispositivos na borda e os terminais frequentemente lidam com sérias limitações físicas de processamento, conhecidas como restrições *SWaP-C* (Tamanho, Peso, Consumo de Energia e Custo). Implementar protocolos de comunicação complexos e verbosos nesse contexto anularia os benefícios estruturais da névoa, pois o processamento excessivo de cabeçalhos elaborados traria de volta atrasos no processamento e longas esperas nas filas dos roteadores Assim, ao projetar sistemas que necessitam de respostas na ordem de milissegundos, é essencial empregar protocolos eficientes e leves na camada de aplicação, como MQTT, CoAP e as abordagens HTTP/REST, cada um desempenhando uma função vital em diferentes níveis da infraestrutura distribuída.

O protocolo MQTT (*Message Queuing Telemetry Transport*), surge como uma das opções mais eficazes para a comunicação na camada de acesso da computação em névoa, especialmente em cenários com banda limitada, necessidade de baixa latência e conexões sem fio instáveis. Baseado no protocolo TCP, o MQTT foge do modelo tradicional de comunicação direta entre cliente e servidor, adotando um sistema de mensagens assíncronas via Publicação/Assinatura. Com isso, os dispositivos IoT, como sensores industriais, medidores de tráfego em cidades inteligentes ou monitores de saúde, funcionam somente como emissores de informações. Eles enviam dados agrupados em tópicos específicos para um intermediário conhecido como *Broker*. Na arquitetura de névoa, esse *Broker* costuma ser implantado de forma distribuída em *gateways*, roteadores ou servidores locais na borda, impedindo que os dados brutos viajem longas distâncias até um data center remoto para serem encaminhados a atuadores e sistemas de acompanhamento. A principal vantagem do MQTT, quando se trata de gerenciar o tempo total de resposta, está na sua estrutura de encapsulamento extremamente leve e nos seus métodos de controle de entrega das mensagens. O cabeçalho fixo do MQTT tem um tamanho mínimo de apenas dois *bytes*, o que resulta em uma redução significativa no tempo de transmissão de pacotes e diminui a carga de processamento nos microcontroladores ao montar os pacotes (MARTÍNEZ et al., 2023). Além disso, para dar conta da criticidade variada das aplicações que são sensíveis à latência, o protocolo disponibiliza três níveis de Qualidade de Serviço (QoS — *Quality of Service*) (RAY,

2018). O nível QoS 0, conhecido como "uma vez", funciona de forma parecida com uma transmissão que não exige confirmação de recebimento, sendo perfeito para enviar dados de telemetria em tempo real de forma contínua e redundante, onde a perda ocasional de um pacote não compromete a segurança do sistema. O nível QoS 1, ou "pelo menos uma vez", assegura a entrega com um pacote de confirmação, sendo usado para avisos e alertas que necessitam ser registrados obrigatoriamente. Por fim, o nível QoS 2, chamado de "exatamente uma vez", utiliza um protocolo de validação em quatro passos mais seguro e confiável, sendo essencial para a automação industrial e sistemas *ciberfísicos* da Indústria 4.0, onde o envio de um comando para operar uma máquina deve ser executado sem o risco de duplicação ou falha na entrega (ATZORI; IERA; MORABITO, 2010).

Como uma alternativa para situações ainda mais limitada, o CoAP (*Constrained Application Protocol*) se apresenta como um protocolo de transporte lógico excelente para aparelhos com limitações críticas de energia e processamento. Ele é utilizado em redes de sensores e atuadores sem fio de baixo consumo com elevadas taxas de falha na transmissão de dados (WSANs e WSNs) (RAY, 2018). Ao contrário do MQTT, o CoAP foi concebido para funcionar sob o modelo de Solicitação e Resposta (*Request/Response*), o que lembra a forma como a web funciona, mas foi inteiramente recriado para otimizar o uso de energia e capacidade de processamento. Sua principal diferença estrutural é a utilização nativa do protocolo UDP (*User Datagram Protocol*) na camada de transporte (RAY, 2018). Ao eliminar a necessidade de estabelecer uma sessão e o *handshake* constante que o TCP exige, o CoAP diminui significativamente o tempo de espera para a primeira conexão. Isso permite que os dispositivos saiam rapidamente do modo de economia de energia (*sleep mode*), enviem suas informações para o nó de névoa e voltem imediatamente ao estado de repouso, estendendo a duração das baterias por muitos anos.

Mesmo operando com o protocolo UDP, que por si só não assegura a entrega de pacotes o CoAP resolve a questão ao incorporar um método próprio e simplificado de checagem em sua própria camada de aplicação (RAY, 2018). As mensagens transmitidas via CoAP podem ser estabelecidas como confiáveis (*Confirmable* — CON) ou não confiáveis (*Non-confirmable — NON*). Ao receber uma mensagem do tipo confirmável de um sensor, um nó da névoa automaticamente envia um pacote de confirmação (*Acknowledgement* — ACK) de tamanho reduzido; se o remetente não receber este pacote dentro de um período pré-determinado, o protocolo inicia reenvios automáticos com base em um timer de espera exponencial (*exponential backoff*). Além disso, por adotar uma

forma de identificar recursos semelhante à da internet convencional, o CoAP possibilita que os *gateways* da camada de névoa funcionem como intermediários de conversão direta, adaptando requisições simples dos sensores locais para os padrões HTTP de comunicação com a nuvem, sem introduzir latência ou sobrecarga de processamento nos dispositivos de ponta (MOURADIAN et al., 2017).

Na progressão de camadas apresentada pela arquitetura formal da névoa, a aplicação do padrão HTTP/REST torna-se estratégica à medida que as informações sobem na hierarquia do processamento (BONOMI et al., 2012). Na camada física e na primeira camada de névoa, os dados brutos e de altíssima frequência são recebidos por meio de MQTT ou CoAP, garantindo respostas de milissegundos para controles locais e de segurança. A partir do momento em que esses nós intermediários realizam a agregação espacial, a filtragem e a compressão temporal desses dados, eliminando redundâncias e reduzindo o volume do tráfego, as informações normalizadas e de maior valor semântico são repassadas para as camadas intermediárias da névoa e para a nuvem através de APIs RESTful (BONOMI et al., 2012). É nessa camada que atuam os microsserviços orquestrados em contêineres e hipervisores, a alimentação de bancos de dados históricos no formato DaaS (*Data as a Service*) e o fornecimento de informações visuais para as interfaces homem-máquina (HMI) em centros de controle e painéis de monitoramento de cidades inteligentes (OPENFOG CONSORTIUM, 2017). Por sua vez, as arquiteturas HTTP/REST (*Hypertext Transfer Protocol / Representational State Transfer*), ocupam um lugar essencial e insubstituível nas porções intermediárias e superiores da arquitetura de névoa, assim como na interface de conexão final com os datacenters em nuvem (OPENFOG CONSORTIUM, 2017). Apesar de os protocolos *RESTful* que usam HTTP serem vistos, em geral, como pesados para microcontroladores básicos, em função do tamanho considerável de seus cabeçalhos em texto e do overhead das conexões TCP e de segurança TLS (*Transport Layer Security*), eles proporcionam o mais alto nível de padronização, interoperabilidade e facilidade de acesso na construção de sistemas de informação. O conceito do *REST* se baseia em um modelo arquitetural que não mantém estado *(stateless*), onde cada operação é realizada de maneira totalmente autônoma e independente, empregando os verbos padrões e globais do HTTP, como *GET* para buscar informações, *POST* para inserir novos dados, *PUT* para modificar e *DELETE* para excluir — na gestão de elementos identificados de forma precisa na rede (RAY, 2018).

Em resumo, a forma como a informação se move num sistema de Internet das Coisas, que visa conciliar processamento distribuído com respostas rápidas, não se baseia numa única solução de comunicação. Em vez disso, conta com uma combinação inteligente e direcionada de vários métodos de intercâmbio de mensagens (YI; LI; LI, 2015). Os dispositivos na borda da rede e aqueles com exigências de tempo rigorosas utilizam a natureza assíncrona e eficiente do MQTT e do CoAP para permitir que máquinas se comuniquem quase instantaneamente, minimizando os tempos de espera e de envio. Ao mesmo tempo, os equipamentos na camada intermediária (névoa) gerenciam a complexidade de servir como tradutores e organizadores do fluxo de dados. Eles empacotam e enviam para a nuvem, usando as abordagens estabelecidas do HTTP/REST, somente os dados importantes e de longo prazo que são necessários para análises gerais (MOURADIAN et al., 2017). Essa organização lógica, dividida de forma precisa, da comunicação é o que realmente permite, no ambiente computacional, que os sistemas com computação em névoa ofereçam a escalabilidade, a robustez e a capacidade de resposta em tempo real que são prometidas (OPENFOG CONSORTIUM, 2017).

4 INTERNET DAS COISAS (IOT) E SUAS APLICAÇÕE S

Mas o que é, afinal, do ponto de vista técnico, a Internet das Coisas? Segundo o artigo (*Introduction to IoT)*, trata-se principalmente da expansão dos serviços convencionais da Internet para uma ampla rede de objetos físicos, como instrumentos, edifícios, veículos e utensílios do dia a dia. Esses objetos se tornam ativos ao serem equipados com eletrônica, circuitos, software e conectividade. Isso permite que sejam percebidos remotamente e controlados com precisão por meio da infraestrutura de rede (GOKHALE; BHAT; BHAT, 2018). Basicamente, a IoT converte elementos que antes eram inativos em participantes ativos na rede.

Ao integrar sensores, atuadores e processamento em uma ampla gama de dispositivos, desde aparelhos pessoais até sistemas industriais, é possível monitorar o entorno, coletar dados e permitir a comunicação automática entre máquinas (IBM, [s.d.]). O resultado é uma união sem precedentes entre o espaço físico e o digital, focada em otimizar recursos, diminuir custos e impulsionar um novo nível de automação na sociedade. Contudo, para um dispositivo ser reconhecido como é preciso de alguns

requisitos de engenharia, Segundo Gokhale, Bhat e Bhat (2018), os requisitos básicos para que um objeto físico seja considerado um dispositivo IoT incluem:

I. **Componentes Integrados:** A rede de objetos físicos possui eletrônicos, circuitos, softwares e sensores integrados

II. **Conexão com a Rede:** Para a Internet das Coisas funcionar, é preciso que os objetos na rede se comuniquem entre si, possibilitando a coleta e o compartilhamento de informações.

III. **Identidade Digital Unica:** Tanto os objetos físicos quanto os virtuais possuem características e identidades próprias.

IV. **Sensoriamento e Gerenciamento à Distância:** A IoT possibilita a detecção e o comando de objetos à distância, utilizando a infraestrutura de rede disponível. Na fase de detecção, os sistemas inteligentes devem conseguir perceber o ambiente de forma automática.

V. **Compartilhamento de Dados:** Na Internet das Coisas, todos os objetos precisam ter a capacidade de compartilhar informações.

VI. **Processamento de Dados:** Além de compartilhar dados, os aparelhos precisam processá-los conforme esquemas pré-determinados, quando necessário.

VII. **Autonomia:** A tecnologia da IoT pode operar sem a necessidade de intervenção humana.

VIII. **Interfaces Inteligentes e Compatibilidade:** Os dispositivos são capazes de utilizar interfaces inteligentes e ser integrados a uma rede de informações.

À medida que os dispositivos conectados ficam cada vez mais comuns,podemos ver desafios como a grande quantidade de dados que são gerados, a latência e a necessidade de um processamento rápido. Ou seja, a integração com a computação em névoa é essencial. Segundo Flavio Bonomi et al. (2012), a computação em névoa coloca os dispositivos IoT mais perto dos nós de processamento. Percebe-se que a proximidade entre os dispositivos IoT e os nós de processamento realmente faz diferença.

Internet das Coisas (IoT) já está presente em muitos setores. Ela ajuda a renovar processos antigos e a mudar a maneira como a gente lida com o ambiente. Isso permite que a gente interaja com novos mundos. Entre as aplicações mais importantes, estão:

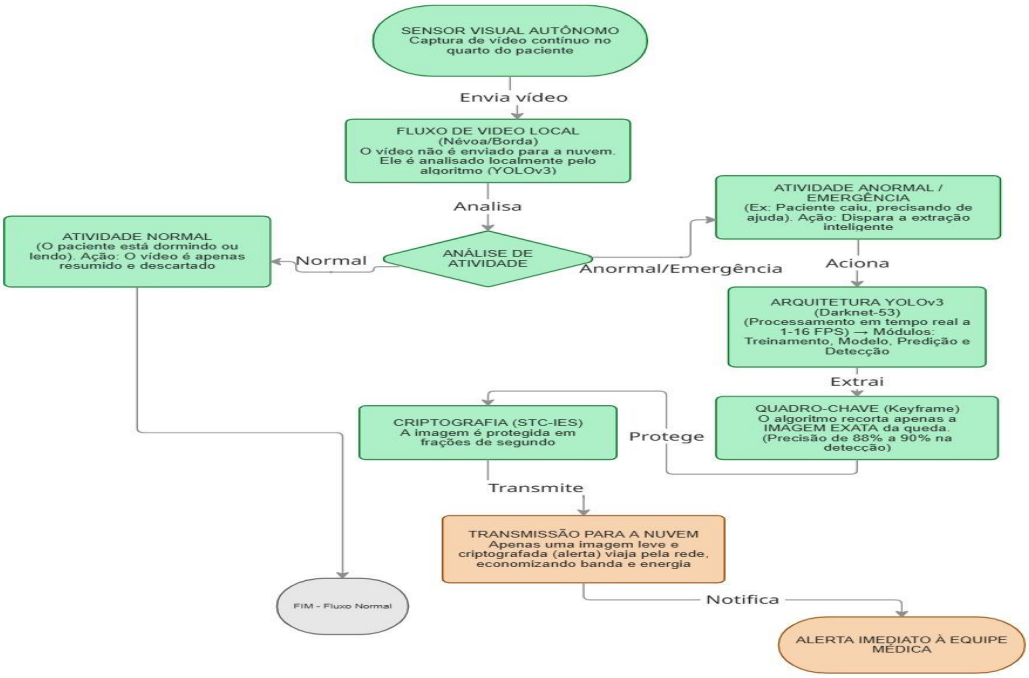
4.1 Saúde

Convenhamos que, por exemplo, na área da saúde, a IoT traz muitos avanços no monitoramento remoto. Os dispositivos vestíveis — smartwatches e sensores biométricos — registram os sinais vitais, como a frequência cardíaca, a pressão arterial e os níveis de oxigênio, em tempo real. Os dados são enviados para os sistemas de análise que monitoram a condição do paciente o tempo todo. A abordagem da IoT pode ser muito útil para os pacientes com doenças crônicas ou para os pacientes que estão se recuperando de cirurgias, porque menos hospitalização permite intervenções mais rápidas e mais convenientes em ambientes perigosos. O OpenFog Consortium (2017), observou que, quando IoT e computação em névoa são usados juntos, os dados cruciais são processados localmente. Dessa forma, IoT e computação em névoa dão respostas imediatas, mesmo se a conexão com a nuvem ficar instável.

O processamento local com baixa latência se mostra eficiente no desenvolvimento de ecossistemas recentes da Internet das Coisas Industrial (IoT) voltados à saúde. Uma análise arquitetural para uma aplicação IoT que demanda agilidade e atenção ao contexto revela que enviar arquivos de vídeo brutos para a nuvem seria excessivamente custoso em termos de banda, impossibilitando respostas rápidas. Por essa razão, além da coleta de dados biométricos, essas plataformas sofisticadas empregam sensores visuais independentes para monitorar o espaço físico do paciente diretamente na borda da rede. Em vez de transmitir vídeos de grande volume, a arquitetura utiliza algoritmos leves processados localmente, como podemos observar no digrama abaixo.

Figura 2- Fluxo de processamento visual descentralizado em ambiente IIoT, ilustrando a extração local de quadros-chave por meio do algoritmo YOLOv3

Fonte: Adaptado de Khan et al. (2022)



Khan et al. (2022), comprovam que essa inteligência descentralizada extrai apenas os quadros-chave associados a atividades incomuns ou emergências médicas, alcançando uma precisão de detecção entre 88% e 90%. Ademais, o sistema opera de maneira eficiente com uma taxa de 1 a 16 quadros por segundo (FPS), adaptando-se perfeitamente a sensores compactos com limitações significativas de processamento e energia. Ao invés de o sistema na nuvem precisar analisar horas de vídeo, o próprio sensor interpreta o contexto e envia um alerta específico e isolado, assegurando ações imediatas por parte da equipe médica para exemplificar.

4.2 Indústria

A Indústria 4.0, ou a Quarta Revolução Industrial, fundamenta-se na Internet das Coisas (IoT). A IoT se consolida pela convergência de três aspectos: os dispositivos físicos (sensores e objetos), a rede (internet) e a semântica (análise de dados). Isso possibilita que objetos comuns interajam autonomamente, visando objetivos compartilhados (ATZORI; IERA; MORABITO, 2010).

Na indústria, essa integração transforma os métodos de produção, acelerando a digitalização e automação em toda a cadeia. Máquinas e ferramentas operam como

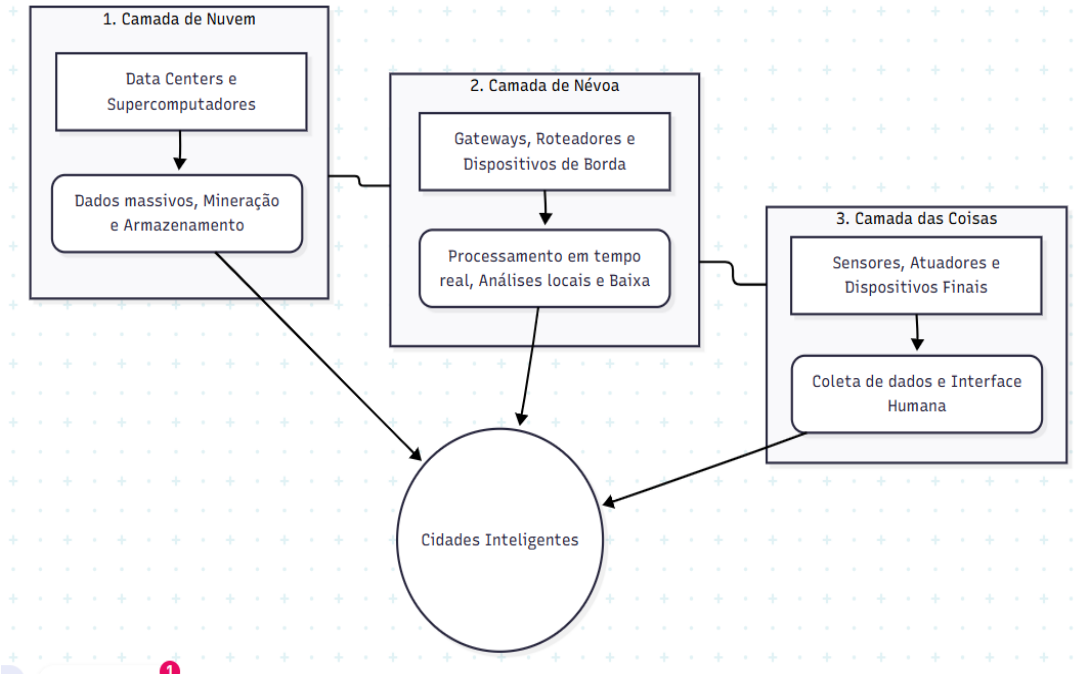
Sistemas *Ciberfísicos*, monitorando em tempo real e conectando o mundo real e o digital para otimizar o fluxo. Uma aplicação chave desse grande volume de dados é a manutenção preditiva. Com as informações constantes da IoT, o sistema identifica padrões e anomalias que sinalizam futuras falhas. Na realidade, prever e prevenir quebras reduz custos, aumenta a flexibilidade, garante entregas rápidas e melhora a produtividade. No entanto, para que essa avaliação em tempo real e distribuída aconteça sem problemas, conforme sugerido pela computação em névoa (BONOMI et al., 2012), as empresas precisam superar barreiras que vão além da infraestrutura de comunicação.

A mudança para esse modelo superconectado encontra obstáculos cruciais na sua implantação. Como apontam Brito, Guazeli Filho e Pepece Junior, a aceitação generalizada da Indústria 4.0 encontra dificuldades consideráveis, com a alta complexidade técnica, os custos elevados para capacitar trabalhadores e, sobretudo, as preocupações com a segurança e a confidencialidade das informações em destaque. Assim, a consolidação desse sistema não se baseia unicamente na tecnologia, mas requer uma colaboração entre o governo, as instituições de ensino e o setor privado para criar normas e incentivar o investimento constante em pesquisa e desenvolvimento, assegurando que o progresso tecnológico avance de maneira segura e duradoura.

4.3 Cidades Inteligentes

Cidades inteligentes empregam a tecnologia da Internet das Coisas (IoT) em conjunto com sistemas *ciber-físicos* (CPS) e redes de sensores sem fio (WSNs) para criar soluções avançadas de gestão urbana e aprimorar consideravelmente a qualidade de vida da população. De acordo com Lata e Kumar (2022), milhares de objetos inteligentes, sensores, atuadores e dispositivos móveis colaboram para coletar dados em tempo real sobre tráfego, consumo de energia, condições climáticas e segurança. A coleta e a análise técnica desses dados servem de base para a tomada de decisões voltadas ao bem-estar da população, resultando em inovações como semáforos inteligentes, iluminação pública adaptável e monitoramento ambiental. Para lidar com o grande volume de dados gerado, a infraestrutura precisa de uma arquitetura de rede dividida em três níveis: borda (edge), névoa (fog) e nuvem (cloud), como é mostrado na Figura.

Figura 3- Ecossistema de Dados em Nuvem e Névoa



Fonte: Adaptado de Lata e Kumar (2022, p. 122).

Conforme explicado por Lata e Kumar (2022), a camada de borda gerencia as operações por meio de software em dispositivos integrados, ligando sensores e executando algoritmos na rede local. Logo acima, a camada de *fog computing* (computação em névoa) cuida do processamento de dados em *gateways* de IoT, responsáveis por guardar, transmitir, pré-processar, agrupar informações e garantir a comunicação entre os componentes. Para facilitar a administração, as arquiteturas utilizam o modelo *DaaS* (*Data as a Service*), fundamentado em Arquitetura Orientada a Serviços (*SOA*) e microsserviços, permitindo o envio ágil de dados quando necessário. Para que esses sistemas críticos operem eficazmente, soluções rápidas são indispensáveis (SATYANARAYANAN, 2017).

O envio de volumes massivos de dados diretamente para a nuvem convencional gera congestionamentos e atrasos na rede. É por isso que o processamento na borda e na névoa forma uma ótima solução para evitar latência, pois funções como classificação, previsão e mineração de dados ocorrem próximas à fonte de geração (LATA; KUMAR, 2022). Adicionalmente, para suportar essa comunicação de alto desempenho, a infraestrutura de fog atua como facilitadora para as redes 5G, integrando os conceitos de Virtualização de Funções de Rede (NFV) e Gerenciamento e Orquestração Autônomos (MANO). Essa integração proporciona taxas de dados multi-Gbps, largura de banda

elevada e permite que os nós da rede tomem decisões autônomas frente a falhas de rede ou ameaças de segurança. Dessa forma, o processamento descentralizado na borda não apenas ameniza atrasos, mas também protege a segurança e as operações em tempo real dos serviços urbanos.

4.4 Veículos Autônomos

Concordamos que os veículos autônomos são um dos usos mais avançados da tecnologia IoT. Os veículos autônomos têm sensores, câmeras, radares e sistemas de inteligência artificial que ajudam a decidir o que fazer em tempo real. Eles precisam analisar muitos dados imediatamente, para garantir que a frenagem, o desvio de obstáculos e a navegação sejam seguros e precisos. A latência dos veículos também é fundamental, porque um atraso pequeno pode causar um erro grave. Mahadev Satyanarayanan (2017), apontou: a computação distribuída próxima à fonte de dados é uma ferramenta importante para ajudar os programas.

Notamos que a computação distribuída próxima à fonte de dados na prática. Pode reduzir o tempo que o sistema leva para responder e, assim, aumenta a confiabilidade do sistema. Portanto, a computação distribuída próxima à fonte de dados traz mais segurança para quem usa o sistema.

5 A LATÊNCIA É UM FATOR CRÍTICO

A latência é o tempo que leva para enviar dados de um ponto a outro numa rede. Isto inclui o tempo de processamento, de transmissão e de resposta. A latência tem grande importância em sistemas de IoT, porque várias aplicações precisam responder rápido e com precisão para funcionar bem. Mahadev Satyanarayanan (2017), afirma que reduzir a latência é um dos maiores desafios das arquiteturas computacionais atuais, especialmente com o crescimento de aplicações que dependem do tempo.

Essa questão ficou ainda mais importante em ambientes que têm muita conectividade, como a Internet das Coisas, como indica o texto. As respostas em tempo real são necessárias nesses casos; por isso, o modelo centralizado acaba sendo menos eficiente e menos confiável, e procuramos soluções descentralizadas como a computação em névoa. A latência não é só uma queda de desempenho, mas tem consequências sérias em aplicações críticas. Em carros autônomos, por exemplo, é preciso tomar decisões em

poucos segundos depois que os sensores e as câmeras coletam os dados. Um atraso de alguns milissegundos na análise dos dados pode impedir que os carros autônomos freiem ou evitem obstáculos, e o risco de acidentes aumenta. Mahadev Satyanarayanan, em 2017, já apontou que é necessário processar os dados perto da fonte para obter respostas quase instantâneas.

A latência é muito importante na área da saúde. Notamos que os sistemas de monitoramento remoto de pacientes, tanto nas unidades de terapia intensiva quanto na gestão de doenças crônicas, precisam de troca contínua de dados importantes. O processamento demorado dos dados pode atrasar o reconhecimento de eventos que ameaçam a vida. Eventos que ameaçam a vida como arritmias cardíacas ou quedas rápidas na pressão arterial podem ser detectados tarde. A latência alta reduz a eficiência das intervenções. Segundo o OpenFog Consortium (2017), o processamento localizado da computação em névoa ajuda a ter respostas rápidas e confiáveis. Portanto, é um ponto chave para garantir rapidez e confiabilidade.

Além desses casos, outras aplicações — como automação industrial, cidades inteligentes e sistemas de segurança — também são muito sensíveis à latência. Atrasos podem prejudicar as linhas de produção nas fábricas, podem atrapalhar a coordenação do tráfego nas cidades e podem atrasar as respostas de emergência nos sistemas de segurança.

Dessa forma, convenhamos que a latência não é só uma métrica de desempenho; a latência também é um fator crítico de segurança e de confiabilidade. Quando precisamos reduzir atrasos, isso nos leva a adotar arquiteturas distribuídas. Uma dessas arquiteturas é a computação em névoa, que coloca os recursos computacionais mais perto dos dispositivos finais. A latência não é só uma medida de desempenho técnico. Também é um ponto chave para a segurança e para a confiabilidade dos sistemas. Quando reduzimos a latência, a latência acaba impulsionando arquiteturas distribuídas, como a computação em névoa, a escalar. Quando melhoramos o desempenho dos dispositivos finais, o sistema consegue dar respostas mais rápidas. Também a operação funciona de forma mais suave, porque esses métodos são mais adequados para ambientes que precisam de alta disponibilidade e de processamento em tempo real.

A latência é um dos maiores problemas na construção de novas aplicações. A tecnologia de computação em nuvem e o processamento distribuído mudam o jeito que o sistema funciona. Essa tecnologia oferece uma solução que atende aos requisitos de desempenho, segurança do software conectado.

6 BENEFÍCIOS DA COMPUTAÇÃO EM NÉVOA

A computação em névoa já é uma das formas mais adequadas que a computação em nuvem pode usar para as aplicações. A computação em névoa se destaca principalmente quando a aplicação precisa de processamento em tempo real e quando a aplicação faz parte da Internet das Coisas (IoT). Na IoT há muitos dados e os dispositivos estão espalhados por toda a região. Na prática podemos identificar que a computação em névoa melhora várias áreas de processamento e traz resultados mais confiáveis, porque alguns recursos de computação ficam mais perto da borda da rede. Quando coloca a computação em névoa mais perto da borda da rede, faz com que esses impactos diminuam. Fazendo o fluxo de informação sejam mais rápido e ajuda a rodar aplicações em tempo real. Por isso, a computação em névoa pode ser a solução crítica para um cenário de Internet das Coisas muito conectado.

Assim também percebe que a computação em névoa trabalha junto com a nuvem em uma arquitetura híbrida. Essa arquitetura está pronta para atender aos requisitos dos casos de uso atuais. Mahadev Satyanarayanan (2017), argumenta que a forma como o processamento é distribuído na infraestrutura de rede é preciso para o desenvolvimento de aplicações futuras. As aplicações futuras precisam de resposta rápida, mobilidade e alta disponibilidade.

6.1 Redução da Latência

A redução da latência ajuda bastante para que os dispositivos IoT sejam inteligentes. Já que a aplicação da computação em névoa em sistemas distribuídos modernos, especialmente na IoT, a latência é o motivo pelo qual a computação em névoa está sendo adotada. Nas configurações tradicionais que usam só a computação em nuvem, o fluxo de dados dos dispositivos finais tem que ir para datacenters centrais. Nesses casos, o fluxo de dados dos dispositivos finais passa por pontos da rede que ficam mais longe do destino do que perto dele. Na prática, o caminho causa atraso na transmissão e faz passar por vários nós da rede pública (hops). Esse longo caminho introduz atrasos cumulativos de transmissão, enfileiramento e processamento em servidores remotos, degradando o desempenho de aplicações sensíveis.

A limitação da infraestrutura centralizado pode ser evidenciada pelo cálculo matemático do tempo de ida e volta (RTT – *Round-Trip Time)* da rede, expresso pela equação:

$$L_{total} = 2 \times \left(L_{prop} + L_{trans} + L_{proc} + L_{fila} \right)$$

Onde:

L_{total} – Tempo total de atraso na rede (Latência de ida e volta / RTT)

L_{prop} – Tempo de propagação física do sinal no meio de transmissão

L_{trans} – Tempo de transmissão do pacote para o canal (largura de banda)

L_{proc} – Tempo de processamento do pacote nos nós/dispositivos

L_{fila} – Tempo de enfileiramento nos roteadores

Ao Analisar a evolução dessas variáveis sob a ótica das redes modernas de transporte de dados, Martínez et al. (2023), demostra que os avanços em switches e roteadores de alto desempenho reduziram o tempo de processamento (L_{proc}) para escalas inferiores a 1 μs. Da mesma forma, o aumento contínuo da capacidade das bandas de transmissão (links acima de 10 Gb/s) minimizou o tempo de transmissão (L_{trans}) para menos de 1.2 μs em pacotes de tamanho máximo. O tempo de enfileiramento (L_{fila}) por sua vez, tende a se manter limitado a poucos microsegundos por salto IP, exceto em picos de tráfego de curta duração.

Consequentemente, em cenários sem congestionamento severo, a latência de uma rede passa a ser delimitada quase inteiramente pelo limite físico do atraso de propagação (L_{prop}). Em meios guiados convencionais, como a fibra óptica de sílica, a velocidade da luz introduz um atraso físico fixo de aproximadamente 5 μs por quilômetro percorrido. Isso significa que, na arquitetura de nuvem tradicional, a distância física continental impõe uma barreira intransponível para o (L_{prop}).

A computação em névoa soluciona essa restrição aproximando a capacidade de processamento da periferia da rede, reduzindo a distância até os dispositivos finais a

poucos metros ou quilômetros. Ao encurtar o caminho físico, o componente L_{prop} decresce drasticamente a valores quase nulos, permitindo respostas na escala de milissegundos.

Essa agilidade é importante tanto para aplicações críticas de segurança quanto para garantir a Qualidade de Experiência (QoE) em sistemas interativos cotidianos. Conforme destacado por Martínez et al. (2023), enquanto os sentidos humanos demandam tempos de resposta entre 1 ms e 100 ms, serviços como os jogos em nuvem exigem pings estritamente abaixo de 60 ms para um funcionamento satisfatório, sendo valores inferiores a 20 ms considerados o patamar ideal. Portanto, a redução da latência é muito importante não só nas aplicações críticas, mas também ajuda a melhorar a experiência do usuário em sistemas interativos, como aplicativos móveis, jogos na nuvem e serviços de streaming. Quando o sistema responde ao feedback, o usuário consegue interagir de forma mais rápida, com interações fluidas e precisas. Deixando assim a resposta do sistema mais eficiente e mais satisfatória do que antes. A computação em névoa ganha destaque quando a latência baixa é necessaria. Mostrando que a computação em névoa tem um papel fundamental nessa situação, quando a latência e reduzida até a borda da rede, mudando a forma como os dados são processados. Essa mudança traz mais agilidade, mais confiabilidade e faz as aplicações distribuídas de hoje se ajustarem melhor.

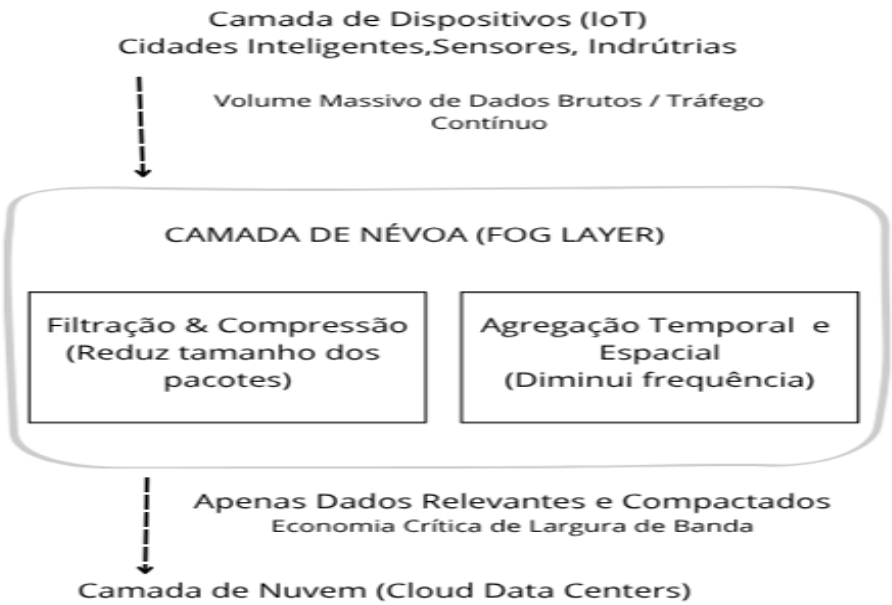
6.2 Economia de Banda

Um dos benefícios mais importantes da computação em névoa é a economia de largura de banda. Em aplicações de Internet das Coisas (IoT) há muitos dispositivos conectados. Sensores e dispositivos inteligentes nesses sistemas geram muitos dados o tempo todo, em vários formatos. Enviar tudo para a nuvem o tempo inteiro pode ser inviável. Ao mesmo tempo, os grandes volumes de dados sobrecarregam a infraestrutura de rede, aumentam os custos de transmissão e geram a congestão, afetam diretamente o desempenho do sistema.

A computação em névoa processa os dados perto do ponto de origem: filtra, remove dados repetidos, agrega os dados e faz uma análise inicial. Assim, só as informações realmente importantes ou já resumidas são enviadas para a nuvem. Além de reduzir o volume de dados que circula na rede e ajuda a usar melhor a largura de banda

disponível. Esse processo não só reduz os custos associados à transmissão de dados, como também ajuda a aumentar a eficiência geral da rede ao diminuir a ocorrência de congestionamento. Este fluxo de filtragem intermediária e o consequente alívio na infraestrutura de rede core estão ilustrados na Figura 4.

Figura 4- Fluxo de dados e otimização de banda na arquitetura de computação em névoa.



Fonte: Adaptado de Alsharif et al. (2025).

A consolidação desse modelo distribuído muda fundamentalmente como os recursos são administrados em ambientes de Internet das Coisas (IoT). Com o tratamento dos dados sendo feito de forma mais local, a estrutura diminui consideravelmente a necessidade de enviar informações por longas distâncias. Em virtude disso, essa melhoria não só mantém a eficiência dos caminhos de comunicação, como também baixa o gasto total de energia da rede e impede problemas de lentidão causados pelo excesso de tráfego na rede principal (Alsharif et al., 2025).

6.3 Segurança

A segurança da informação é outro assunto importante, e também um dos mais difíceis nos sistemas que usam IoT. A segurança da informação depende muito da forma como usamos a computação. A computação em névoa permite que parte do processamento de dados aconteça aqui mesmo, no dispositivo, ou nos nós que ficam na borda da rede. Ela reduz a necessidade de mandar sempre informações sensíveis para os servidores em nuvem centralizados. Essa característica diminui a exposição de dados na

conexão. No tráfego de rede, impede que alguém intercepte a comunicação. E também bloqueia os ataques "*man-in-the-middle (MitM)"* e impede vazamentos de informações.

Além disso, como os dados sensíveis ficam próximos ao local onde são gerados, a computação em névoa permite aplicar políticas de segurança que são mais adequadas a cada ambiente. O que também controla melhor o fluxo de informações. Os dados sensíveis podem comprometer seriamente a segurança do usuário ou do processo, e os dados sensíveis são fundamentais em áreas como saúde digital, cidades inteligentes e sistemas industriais.

Um exemplo prático disso é apresentado por Praciano et al. (2019), que desenvolveram um sistema de segurança física com IoT e Machine Learning para reconhecimento facial. Com a computação em névoa, o reconhecimento facial é realizado perto da câmera (na borda), ao invés de ir até a nuvem. E também em vez de transmitir um fluxo contínuo de vídeos e dados biométricos brutos pela internet – o que implicaria um alto risco de interceptação –, o sistema processa a identidade localmente e envia apenas alertas ou notificações para o usuário final via aplicativo ou bot. Essa demonstração prática demostra como a distribuição do processamento minimiza o tráfego de informações críticas.

Juntamente com a rapidez nas respostas, essa forma de organização cria uma linha de defesa segura, que se destaca tanto pela sua solidez quanto pela otimização do uso de recursos. A pesquisa de Praciano et al. (2019), também demonstra que, ao concentrar o processamento intenso e as informações visuais sensíveis apenas nos servidores de névoa, impede-se o congestionamento da rede de internet. Essa vantagem garante que, mesmo diante de possíveis ataques de negação de serviço distribuído (DDoS) que visem a conexão principal, as atividades de vigilância permaneçem ativas. Por exemplo, Um ataque hacker DDoS e enviando uma enxurrada de tráfego que paralisa sua conexão de internet.

Na Névoa: Caso sua internet apresente falhas ou sofra um ataque, não há com o que se preocupar. O sistema central de reconhecimento facial está localizado dentro do próprio edifício. A câmera e o portão permanecem em comunicação através de conexões físicas locais. Quando um morador se aproxima, é identificado e consegue acessar as instalações, mesmo que todo o edifício esteja completamente desconectado da internet. Essa funcionalidade é o que o texto denomina "garantia de funcionamento contínuo diante de interrupções externas".

Assim, a computação em névoa funciona como uma proteção isolada: ela mantém os dados biométricos dentro da área protegida, garante a operação contínua contra interrupções externas e oferece uma segurança de dados que demanda muito menos estrutura de rede em comparação com uma solução inteiramente na nuvem.

Como foi dito anteriormente que a arquitetura distribuída traz mais resistência ao sistema. Ao contrário dos modelos centralizados, a computação em névoa apoia a operação mesmo quando a conexão com a nuvem falha ou quando alguns nós ficam indisponíveis. Quando um nó único falha, a computação impede que toda a operação pare, pois o processamento está distribuído. Isso deixa o sistema mais forte e mais capaz de aguentar falhas. A computação em névoa é fundamental quando colocamos o sistema em ambientes sensíveis a latência. A descentralização gera novos problemas de segurança. Quando mais nós processam, a superfície de ataque do sistema aumenta e pede mecanismos mais avançados de autenticação, de criptografia e de gerenciamento de identidade esse crescimento na superfície de ataque na computação em névoa, encontra suporte técnico e uma análise mais detalhada nos estudos de Carmine Cesarano, (2023).

Ao migrar o processamento, o armazenamento e o gerenciamento de dados das nuvens centralizadas para nós distribuídos na borda da rede, a arquitetura beneficia-se de menor tempo de resposta e maior robustez na operação, contudo, inevitavelmente, o número de terminais (*endpoints*) acessíveis se eleva. Essa distribuição estrutural faz com que a proteção não se limite mais a um único contorno o centro de dados na nuvem, passando a exigir a proteção de múltiplos dispositivos geodistribuídos, onde cada ponto de processamento pode se tornar um caminho possível para invasões e interceptações de rede.

A expansão dessa área de exposição a riscos é intensificada pela diversidade e pelas barreiras físicas dos pontos de processamento na borda. Ao contrário dos servidores na nuvem, os aparelhos localizados na borda da rede utilizam hardwares, microcontroladores e sistemas operacionais diversos, muitas vezes incorporando softwares intermediários de código aberto e processos de desenvolvimento acelerados que podem gerar pontos fracos. Adicionalmente, tais dispositivos enfrentam limitações rigorosas em dimensão, peso, consumo de energia e preço (aspectos conhecidos como *SWaP-C*), o que frequentemente impede a implementação de recursos de segurança convencionais e volumosos. Essa fraqueza operacional permite que atacantes tirem proveito de credenciais predefinidas ou falhas em equipamentos ligados para controlá-

los, constituindo redes de máquinas controladas remotamente (*botnets*) capazes de orquestrar ofensivas de interrupção de serviço distribuído (*DDoS*), como as observadas com softwares maliciosos como Mirai (2016) e IoT Reaper (2017).

Para harmonizar a distribuição de tarefas computacionais com a necessidade de defesas mais fortes, a computação em névoa usa um conceito de "defesa em camadas" voltado para a separação rigorosa de atividades em ambientes compartilhados por múltiplos usuários. Em pontos de processamento da névoa que dividem recursos físicos entre diversos clientes ou aplicações essenciais, a proteção inicial contra perigos se dá pela virtualização no nível do hardware, utilizando *hipervisores* integrados e minimalistas como ACRN, *Bao* ou *Jailhouse*. Tais ferramentas viabilizam a junção de várias unidades virtuais autônomas em um único dispositivo físico na borda da rede, assegurando que cada espaço de trabalho funcione de maneira isolada logicamente. Assim, o *hipervisor* funciona como um escudo estrutural, prevenindo interrupções diretas, visualização indevida de dados e partilha não autorizada de memória entre os sistemas operacionais ativos no mesmo hardware. Adicionando-se a essa separação inicial, a estrutura introduz um segundo nível de isolamento através de contêineres, geridos por ferramentas de coordenação de software intermediário tipo *Kubernetes*. Dentro de cada unidade virtual, os contêineres agrupam pequenos serviços e seus componentes relacionados, definindo estritamente o alcance de execução e os direitos de cada software. Essa organização dupla — unidades virtuais separadas por *hipervisores* integrados e softwares isolados por contêineres — garante que, se um atacante explorar uma falha e burlar a segurança de um serviço na borda da rede, o ataque ficará estritamente contido no limite daquela tarefa específica. Isso, por sua vez, impede que o invasor se mova lateralmente, evitando que ganhe acesso a níveis mais altos do sistema operacional ou comprometa dados e processos importantes de outros utilizadores da estrutura física.

Outro avanço fundamental para fechar brechas no *middleware* é a combinação de configurações seguras e técnicas de *debloating* (redução de software). Orquestradores complexos tendem a trazer centenas de funcionalidades nativas que, quando não utilizadas, apenas aumentam a superfície vulnerável. A pesquisa de Cesarano (2023), demostra que a aplicação de normas recomendadas, como *o CIS Benchmark*, juntamente com a análise de linguagem para interpretar registros de correções, permite estabelecer configurações que desativam funcionalidades ociosas e garantem criptografia robusta e barreiras de proteção integradas. De forma mais aprofundada, a remoção de software

guiada por configuração, uma metodologia de exame prévio, distingue e elimina de imediato quaisquer linhas de código ou elementos que não terão uso (como drivers de volumes de armazenamento não utilizados no *Kubernetes*), cortando efetivamente partes do programa onde falhas poderiam se esconder.

6.4 Escalabilidade

Com o aumento da complexidade da Internet das Coisas (IoT), os bilhões de dispositivos se juntam para gerar e enviar dados. A escalabilidade se torna uma métrica essencial nas muitas arquiteturas de computação. Por isso, é muito importante que os sistemas cresçam de forma eficaz e que os sistemas não atrapalhem o desempenho, a latência ou a confiabilidade das aplicações.

A computação em névoa traz uma arquitetura que cresce muito quando os recursos computacionais são distribuídos pela rede. Assim, permite que os novos dispositivos de borda, os sensores e os nós de processamento sejam integrados de forma gradual, sem precisar mudar a infraestrutura central de forma drástica. O sistema vai se expandir passo a passo, atendendo ao aumento da demanda de forma mais eficiente e equilibrada. A computação em névoa oferece escalabilidade horizontal ao sistema computacional.

Já a computação em nuvem tradicional depende de ampliar grandes centros de dados centralizados para suportar o crescimento das cargas de processamento. A computação em névoa permite introduzir novos nós na borda da rede e distribuir a carga de trabalho entre vários nós. Essa característica reduz gargalos, melhora o desempenho do sistema e aumenta a adaptabilidade a diferentes casos de usos. Na prática, esse tipo de crescimento distribuído é essencial em ambientes que são mais dinâmicos e diferentes, como cidades inteligentes, sistemas industriais complexos e infraestruturas críticas, incluindo a IoT (OpenFog Consortium 2017). Nesses casos, a flexibilidade e a capacidade de escalar e otimizar o desempenho exigem que o sistema opere de forma eficiente e seja sustentável.

Além do mais, a escalabilidade da computação em névoa ajuda a gerir os recursos, assim os pontos centrais da rede não ficam sobrecarregados e a computação em névoa faz um uso mais eficiente dos recursos que tem. Por isso, esse modelo é uma solução ideal para lidar com o crescimento rápido e constante dos ambientes conectados hoje.

6.5 Confiabilidade

A confiabilidade é uma característica importante dos sistemas distribuídos atuais. Nos ambientes de Internet das Coisas (IoT) a confiabilidade se torna ainda mais essencial, pois a operação sem interrupções garante a segurança e a eficiência das tarefas. A computação em névoa tem um papel importante para melhorar a confiabilidade dos sistemas distribuídos. Isso porque a arquitetura distribuída permite que o processamento não dependa totalmente dos servidores de nuvem centralizados. Assim, a confiabilidade dos sistemas distribuídos cresce.

Quando espalhamos a carga de computação por dispositivos de borda e por nós intermediários, reduzindo a dependência de uma infraestrutura central. Assim, os dispositivos locais e os nós de névoa continuam funcionando bem, mesmo se a nuvem se desconectar, se a rede ficar congestionada ou se os servidores remotos não estiverem disponíveis. Dessa forma, a computação em névoa garante que os serviços críticos continuem sendo executados sem interrupções.

Notamos também que a disponibilidade operacional faz o sistema ficar mais forte e faz a quantidade de interrupções totais de serviço ficar menor. A disponibilidade operacional é essencial para aplicações críticas como o monitoramento de pacientes em tempo real, o controle de infraestrutura urbana — semáforos, redes elétricas — e a automação industrial. Nesses casos, as interrupções podem causar um impacto grande na operação, na segurança dos usuários e no meio ambiente. Por isso, a confiabilidade deixa de ser apenas um requisito do sistema e passa a ser o que faz as aplicações críticas viáveis. A descentralização do processamento é um ponto importante para criar esses sistemas, como disse Mahadev Satyanarayanan (2017). A descentralização do processamento permite que o sistema continue funcionando mesmo quando algumas partes falham parcialmente. A resiliência que a descentralização do processamento oferece é uma das principais diferenças da computação em névoa em comparação com os modelos centralizados clássicos. Normalmente, as vantagens da computação em névoa incluem latência menor, uso reduzido da largura de banda, maior segurança, melhor escalabilidade e confiabilidade aprimorada. Essas vantagens mostram que a computação em névoa funciona como uma solução que complementa a computação em nuvem.

A computação em névoa, baseada nos princípios da computação descentralizada, permite que o processamento seja distribuído e que a computação fique mais próxima da fonte de dados do usuário. Essa proximidade é fundamental para atender aos requisitos

cada vez maiores das aplicações de IoT de hoje. Assim, a integração de nuvem e névoa é um processo de arquitetura estratégica. A integração de nuvem e névoa traz todas as capacidades que os sistemas modernos precisam para ser mais resilientes, mais eficientes, mais seguros e mais adequados, diante dos desafios da computação moderna.

7 CONCLUSÃO

O crescimento da Internet das Coisas (IoT) mudou a forma como os dados são criados, analisados e usados em vários setores, como a saúde, a indústria, a mobilidade e a gestão urbana. Eu percebo que, no contexto da Internet das Coisas (IoT), precisamos processar os dados na hora, ter tempo de resposta rápido e usar menos recursos. Essa necessidade da Internet das Coisas (IoT) deixa claros os limites da computação em nuvem tradicional, que ainda depende de servidores longe.

Podemos notar que a latência tem um papel muito importante nas aplicações modernas. A latência afeta as decisões em tempo real. As decisões em tempo real são essenciais para veículos autônomos. As decisões em tempo real também são essenciais para sistemas de monitoramento de saúde e para sistemas de monitoramento pessoais. E ainda são críticas na automação industrial. Mahadev Satyanarayanan (2017), explicou em seu artigo que as distâncias entre a fonte dos dados e os centros de processamento podem mudar muito o desempenho do sistema. As distâncias entre a fonte dos dados e os centros de processamento também podem mudar muito a confiabilidade do sistema.

Por isso, muitas vezes é preciso usar um tipo totalmente novo de métodos de computação. Para esse fim, a computação em névoa aparece como uma solução prática, trazendo recursos adicionais na nuvem que ficam mais perto do limite da rede. Flavio Bonomi et al. (2012), mostram que esse modelo permite que o processamento, o armazenamento e o controle sejam distribuídos por toda a infraestrutura, reduzindo a latência e melhorando o tráfego de dados. Explicamos a importância da descentralização para sistemas baseados em IoT. Outros benefícios aplicáveis.

Na computação em névoa, percebe-se que a segurança melhora, a escalabilidade cresce, a economia de largura de banda aumenta e a confiabilidade do sistema fica mais forte, além da redução de latência que já mencionei. O OpenFog Consortium (2017), aponta que a computação em névoa, ao processar dados localmente e de forma distribuída, fortalece o sistema e permite que ele se ajuste a aplicações mais dinâmicas.

Outra mensagem importante que destaquei na análise foi a relação entre a computação em névoa e a computação em nuvem. A computação em névoa não substitui a computação em nuvem. Em vez disso, a computação em névoa funciona como uma camada intermediária que melhora as capacidades e distribui os processos de maneira mais eficiente. Quando o trabalho exige armazenamento em massa ou processamento

intensivo, a computação em nuvem tradicional ainda é necessária. Mas a computação em névoa atende à necessidade de resposta rápida imediatamente. Que a integração da computação em nuvem com a computação em névoa funciona melhor hoje em um modelo híbrido, principalmente para a IoT. A computação em nuvem e a computação em névoa juntas criam um meio-termo entre o desempenho, a eficiência e a segurança. O desempenho, a eficiência e a segurança são fundamentais para que as aplicações vitais continuem operando de forma eficaz.

Os autores explicam que, com o aumento constante de dispositivos conectados e com a complexidade que aumenta nos sistemas distribuídos, a computação em névoa surgiu como um campo de estudo mais adequado. O estudo da computação em névoa e o desenvolvimento da computação em névoa são importantes para melhorar as infraestruturas tecnológicas. Mais também ajudam a apoiar novas aplicações. Assim, avançamos para uma nova era da informação.

REFERÊNCIAS

ALSHARIF, Mohammed H. et al. Survey of energy-efficient fog computing: Techniques and recent advances. **Energy Reports**, v. 13, p. 1739-1763, 2025 ATZORI, Luigi; IERA, ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The internet of things: A survey. **Computer networks**, v. 54, n. 15, p. 2787-2805, 2010.

BONOMI, Flavio et al. Fog computing and its role in the internet of things. In: **Proceedings of the first edition of the MCC workshop on Mobile cloud computing**. 2012. p. 13-16.

BRITO, Jean Carlos Valim de. Evolução da indústria 4.0: O papel fundamental da IOT. 2024.

COUTINHO, A. A. T. R.; CARNEIRO, Elisângela Oliveira; GREVE, Fabíola Gonçalves Pereira. Computaçao em névoa: Conceitos, aplicaç oes e desafios. **Minicursos do XXXIV SBRC**, p. 266-315, 2016.

CESARANO, Carmine. Security Assessment and Hardening of Fog Computing Systems. In: **2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)**. IEEE, 2023. p. 22-25.

DATTA, Soumya Kanti; BONNET, Christian; HAERRI, Jerome. Fog computing architecture to enable consumer centric internet of things services. In: **2015 International Symposium on Consumer Electronics (ISCE)**. IEEE, 2015. p. 1-2.

EVANS, D. The Internet of Things: How the Next Evolution of the Internet is Changing Everything. Cisco Internet Business Solutions Group (IBSG), White Paper, abr. 2011. Disponível em: https://www.cisco.com. Acesso em: 23 mai. 2026.

GOKHALE, Pradyumna; BHAT, Omkar; BHAT, Sagar. Introdução à IoT. Revista Internacional de Pesquisa Avançada em Ciência, Engenharia e Tecnologia , v. 5, n. 1, p. 41-44, 2018.

IBM. O que é IoT (internet das coisas)?. [s.d.]. Disponível em: https://www.ibm.com/br-pt/think/topics/internet-of-things. Acesso em: 12 jun. 2026.

KHAN, Jalaluddin et al. Secure smart healthcare monitoring in industrial internet of things (iiot) ecosystem with cosine function hybrid chaotic map encryption. **Scientific Programming**, v. 2022, n. 1, p. 8853448, 2022.

LATA, Manju; KUMAR, Vikas. Fog computing infrastructure for smart city applications. In: **Recent advancements in ICT infrastructure and applications**. Singapore: Springer Nature Singapore, 2022. p. 119-133.

MARTÍNEZ, Gonzalo et al. Round trip time (rtt) delay in the internet: Analysis and trends. **IEEE Network**, v. 38, n. 2, p. 280-285, 2023.

MOURADIAN, Carla et al. A comprehensive survey on fog computing: State-of-the-art and research challenges. **IEEE communications surveys & tutorials**, v. 20, n. 1, p. 416-464, 2017.

OPENFOG CONSORTIUM. OpenFog Reference Architecture for Fog Computing. [S. l.]: OpenFog Consortium, 2017. Disponível em:

https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf. Acesso em: 12 fev. 2026.

PRACIANO, Bruno JG et al. Segurança do Ambiente Usando Dispositivo IoT com Processamento Distribuído. In: **Atas das Conferências Ibero-Americanas WWW/Internet 2019 e Computação Aplicada 2019**. 2019. p. 163-170.

RAY, Partha Pratim. A survey on Internet of Things architectures. **Journal of King Saud University-Computer and Information Sciences**, v. 30, n. 3, p. 291-319, 2018 SATYANARAYANAN, Mahadev. The emergence of edge computing. **computer**, v. 50, n. 1, p. 30-39, 2017.

SPOHN, Marco Aurélio; DA SILVA BONETTI, Fernanda. Análise de desempenho de uma arquitetura de fog computing para internet of things via estudos de caso. Revista Brasileira de Computação Aplicada, v. 11, n. 3, p. 72-87, 2019.

YI, Shanhe; LI, Cheng; LI, Qun. A survey of fog computing: concepts, applications and issues. In: **Proceedings of the 2015 workshop on mobile big data**. 2015. p. 37-42.