# Gerenciamento Adaptável de Dispositivos IoT Heterogêneos em Smart Homes

Ivo A. Pimenta
*Centro de Ciências e Tecnologia*
*Universidade Estadual do Ceará*
Fortaleza-CE, Brasil
aguiar.pimenta@aluno.uece.br
ORCID 0009-0005-4571-6242

Antonio M. Braga Neto
*Centro de Ciências e Tecnologia*
*Universidade Estadual do Ceará*
Fortaleza-CE, Brasil
mozar.braga@aluno.uece.br
ORCID 0009-0000-1836-0985

Evellin S. Moura
*Centro de Ciências e Tecnologia*
*Universidade Estadual do Ceará*
Fortaleza-CE, Brasil
evellin.moura@aluno.uece.br
ORCID 0009-0008-7786-9990

Kaynan S. Freitas
*Centro de Ciências e Tecnologia*
*Universidade Estadual do Ceará*
Fortaleza-CE, Brasil
kaynan.freitas@aluno.uece.br
ORCID 0009-0003-6903-5295

Marcello H. Lee
*Centro de Ciências e Tecnologia*
*Universidade Estadual do Ceará*
Fortaleza-CE, Brasil
marcello.lee@aluno.uece.br
ORCID 0009-0001-2668-3116

Rafael L. Gomes
*Centro de Ciências e Tecnologia*
*Universidade Estadual do Ceará*
Fortaleza-CE, Brasil
rafa.lopes@uece.br
ORCID 0000-0001-7922-0695

***Resumo*—Nos últimos anos, as Casas Inteligentes (SHs) tornaram-se uma realidade devido à popularização da Internet das Coisas (IoT). Este tipo de ambiente é composto por vários dispositivos heterogêneos sem fio, formando uma rede complexa para ser gerenciada. Portanto, uma SH necessita de uma abordagem capaz de abstrair a comunicação com esses dispositivos de IoT. Dentro desse contexto, este artigo propõe um middleware para abstração da comunicação dos dispositivos IoT em casas inteligentes. O middleware proposto permite a comunicação com os dispositivos através de uma interface de alto nível, possibilitando o gerenciamento com os dispositivos através de uma interface genérica. Os experimentos realizados em um cenário real sugerem que o middleware proposto é escalável e permite o controle dos dispositivos de IoT de forma genérica.**

***Index Terms*—Casas Inteligentes, Comunicação Sem Fio, Internet das Coisas, Middleware.**

## I. Introdução

Nos dias atuais, onde as pessoas estão cercadas de dispositivos de comunicação que trocam informações, a chamada era da Internet das Coisas (*Internet of Things* - IoT) [1]. Como consequência disso, aumentou a heterogeneidade das redes sem fio, tanto em relação ao tipo de dispositivos conectados (como, por exemplo, PCs, tablets, e smartphones), quanto na sua função, protcolos e serviços atribuídos (câmeras de segurança, eletrodomésticos inteligentes, etc) [2]. Dentro desta nova realidade, tem-se o surgimento de um novo contexto: as Casas Inteligentes (*Smart Homes* - SHs): residências compostas de diversos dispositivos inteligentes que comunicam-se entre si e com os dispositivos dos usuários [3].

Casas Inteligentes caracterizam-se por um ambiente com diversos dispositivos de uso pessoal (aparelhos de uso pessoal, que na maioria das vezes geram tráfego de/para a Internet) e inteligentes (dispositivos que visam automação de funções na residência, que comunicam-se entre si e/ou com os usuários) interconectados por uma rede sem fio residencial (*Wireless Home Networks* - WHNs). Casas Inteligentes tornaram-se um ambiente complexo com diversos serviços executados a fim de dar suporte ou aprimorar a experiência de vida dos usuários, tais como identificação de doenças [4], ocorrência de ataques cibernéticos [5], monitoramento dos ambientes e desempenho dos equipamentos [6], dentre outros. Muitos desses serviços são baseados em modelos de Inteligência Artificial (IA) [7], [8], os quais são implantados e alimentados através da comunicação com os dispositivos heterogêneos conectados através da WHN.

Entretanto, devido à natureza heterogênea desses dispositivos (diferença entre fabricantes e propósitos), eles possuem características distintas, utilizando diversas tecnologias de comunicação diferentes [9]. Ainda não existe uma forma padronizada de os dispositivos presentes nas Casas Inteligentes se comunicarem entre si, bem como com a Internet, comprometendo assim um gerenciamento eficaz dos dispositivos e serviços nas Casas Inteligentes [10].

Dentro deste contexto, este artigo apresenta uma proposta para abstrair a complexidade dos diferentes dispositivos conectados a Casas Inteligentes através de um middleware para intermediar a comunicação (acesso e troca de informações) com os dispositivos inteligentes, bem como atender à escalabilidade necessária para este tipo de cenário. O objetivo do middleware proposto é facilitar a configuração desses dispositivos inteligentes, além de auxiliar a utilização desses dispositivos pelas aplicações e serviços dos usuários da rede. Experimentos realizados em um *testbed* real, apontam a capacidade do middleware proposto de habilitar o gerenciamento de diversos dispositivos inteligentes heterogêneos, bem como a escalabilidade necessária para ser implantado em um cenário real.

Desta forma, as principais contribuições deste trabalho são: Desenvolvimento de um middleware modular que abstrai a heterogeneidade dos dispositivos IoT em casas inteligentes; Facilitação da interoperabilidade entre dispositivos de diferentes fabricantes e protocolos (Wi-Fi, ZigBee, Bluetooth, etc.); Suporte à escalabilidade, possibilitando o gerenciamento eficiente de ambientes com um número crescente de dispositivos e aplicações; e, Superação de limitações de soluções existentes, que são focadas em protocolos específicos ou não ofereciam abstração completa da camada de dispositivos.

O restante deste artigo está organizado da seguinte forma. A Seção II descreve uma revisão da literatura com alguns trabalhos já feitos na área. A Seção III irá descrever o estudo realizado nesse trabalho, enquanto a Seção IV discute os experimentos e os resultados, respectivamente. Por fim, a Seção V conclui o artigo e apresenta trabalhos futuros.

## II. Trabalhos Relacionados

A seguir, apresenta-se alguns trabalhos relacionados a estratégias de integrar dispositivos físicos heterogêneos e prover uma interface de alto nível para desenvolvimento de aplicações em SHs. A Tabela I resume os pontos de destaque de cada trabalho, tais como o contexto e a abordagem aplicada pelo mesmo.

Bouloukakis et al. [11] propõem um middleware baseado em redes definidas por software, focando no suporte a serviços de coleta de dados para missões críticas, habilitando aspectos de confiabilidade e temporização através da priorização de envio e controle de filas. Da mesma forma, Abbasi et al. [12] descrevem middleware multi-camada para rastrear a interação entre dispositivos IoT heterogêneos, visando identificar comunicações não seguras e interoperáveis que podem comprometer o sistema IoT como um todo. Todavia, estas propostas não tem por objetivo abstrair a interação dos serviços com os dispositivos ao implantar soluções.

Kelly et al. [13] implementam um modelo de automação residencial para monitoramento de condições ambientais e gerenciamento de energia. O esquema é composto de uma rede de sensores usando o protocolo ZigBee e um coordenador conectado a um roteador que age como *gateway* IoT ligando a rede ZigBee com a internet. Os dados são coletados dos sensores por uma aplicação no *gateway* e enviados para um servidor web onde podem ser visualizadas. No entanto, esse trabalho contempla apenas dispositivos ZigBee e nas residencias atuais existe dispositivos inteligentes usando outros protocolos, como WiFi e Bluetooth. Por outro lado, neste artigo propõem-se uma abordagem que aceite diversos protocolos de comunicação, pois o foco é exatamente abstrair detalhes de baixo nível para as camadas superiores.

Chilcanan et al. [14] desenvolvem um assistente virtual para automação de ambientes residenciais, gerenciando sensores IoT, bem como permitindo o controle e monitoramento destes sensores. Os autores aplicam um middleware orientado a mensagem (Message Oriented Middleware - MOM) para interconectar os dispositivos, focando no monitoramento do consumo de energia da residencia em tempo real. Portanto, esta proposta visa somente o consumo de energia, tendo assim sua aplicabilidade restrita em relação ao middleware proposto neste trabalho.

Benson et al. [15] propõem Resilient IoT Data Exchange (RIDE) um middleware para mediar a troca de informações entre as aplicações executando em nuvens e dispositivos IoT, aumento a capacidade de análise e armazenamento de dados. RIDE foca na comunicação resiliente através de infraestruturas de rede SDN em duas fases, uma de coleta de dados para análise e outra de disseminação de alertas inteligentes. Desta forma, o RIDE trabalha na intermediação entre nuvem e dispositivos, mas não atua no tratamento de dispositivos heterogêneos e a abstração para os mesmos.

Cruz et al. [16] apresentam uma arquitetura de middleware para sistemas IoT que visa dar suporte aos protocolos de rede abertos mais usados pelas soluções existentes, tais como MQTT, CoAP e HTTP. A arquitetura, apesar de tentar se adequar as necessidades de um ambiente heterogêneo, ainda apresenta uma falta de flexibilidade e baixa carga de processamento, características necessárias para atender os requisitos dos serviços IoT em SHs, como por exemplo saúde e segurança. Similarmente, Anya et al. [17] apresentam um middleware cognitivo para gerenciar a interação entre os dispositivos inteligentes e os usuários presentes na SH. O mecanismo cognitivo do middleware aprende e adapta-se ao estilo de vida dos usuários e suas preferências. O middleware possui uma interface homogênea para as aplicações adaptarem-se ao comportamento do ambiente e seus requisitos. Contudo, este trabalho depende de dispositivos específicos presentes no ambiente capazes de interagir com o middleware, não contemplando assim a heterogeneidade das SHs.

A partir da revisão literária realizada, não foram encontrados trabalhos que focam diretamente no desenvolvimento de um middleware capaz de abstrair, de forma completa e eficiente, a complexidade inerente aos diferentes dispositivos conectados em casas inteligentes. Embora existam soluções que tratam aspectos específicos, como priorização de tráfego em redes IoT baseadas em SDN, frameworks de confiança ou arquiteturas restritas a determinados protocolos como ZigBee, estas abordagens não oferecem uma camada de abstração suficientemente genérica e adaptável para lidar com a heterogeneidade crescente dos dispositivos presentes em ambientes residenciais inteligentes. Além disso, algumas propostas estão limitadas ao monitoramento de parâmetros específicos, como consumo de energia, ou são dependentes de dispositivos específicos, o que restringe sua aplicabilidade em cenários mais amplos.

Diante deste cenário, a proposta deste trabalho avança o estado da arte ao oferecer uma arquitetura de middleware modular e agnóstica quanto aos protocolos e fabricantes, capaz de integrar sensores e atuadores diversos através de uma interface de alto nível. Diferentemente dos trabalhos analisados, que não endereçam de forma abrangente o problema da heterogeneidade, este middleware permite que aplicações se comuniquem com qualquer dispositivo suportado sem precisar lidar com os detalhes dos protocolos subjacentes ou das implementações específicas de cada fabricante. Além disso, o modelo proposto

Tabela I
TRABALHOS RELACIONADOS

| Referência | Contexto | Abordagem |
|---|---|---|
| [11] | Priorização de eventos em sistemas IoT baseados em SDN. | Middleware combina algoritmos de filas de prioridade e políticas de descarte. |
| [12] | DInteroperabilidade e confiança em sistemas IoT heterogêneos. | Middleware baseado em múltiplas camadas com um framework de confiança. |
| [13] | IoT para monitoramento ambiental em casas inteligentes. | Arquitetura de sensores conectados via ZigBee e internet. |
| [14] | Gerenciamento de processos IoT e automação de casas inteligentes. | Middleware orientado a mensagens (MOM) que integra dispositivos para monitorar o consumo de energia e gerenciar dispositivos conectados via um assistente virtual (chatbot). |
| [15] | Uso de SDN e computação em borda para continuidade em IoT. | Middleware Ride monitora e redireciona tráfego de dados IoT. |
| [16] | Comunicação eficiente e segura entre dispositivos IoT. | Middleware escalável com suporte a credenciais seguras e consultas otimizadas. |
| [17] | Interação consciente do contexto em casas inteligentes. | Middleware cognitivo que ajusta a casa inteligente com base em dados contextuais. |
| Nossa Proposta | Interoperabilidade e integração de dispositivos IoT heterogêneos em casas inteligentes. | Middleware modular que abstrai a comunicação entre dispositivos IoT de diferentes fabricantes e protocolos, permitindo escalabilidade, interoperabilidade e comunicação eficiente em tempo real, com suporte a múltiplos protocolos. |

não apenas garante a interoperabilidade, como também provê mecanismos eficientes de gerenciamento, cadastro, atuação e coleta de dados, o que amplia sua aplicabilidade para diferentes serviços e aplicações dentro do contexto de casas inteligentes.

## III. PROPOSTA

O middleware de abstração proposto nesse trabalho visa auxiliar as aplicações e serviços prestados na Casas Inteligentes ao habilitar o acesso e gerenciamento dos dispositivos inteligentes, ao prover uma interface de alto nível que abstrai a comunicação com esses dispositivos heterogêneos. Sendo assim, a Figura 1 apresenta a arquitetura do middleware proposto e como é feita a integração dos dispositivos e aplicações presentes nas Casas Inteligentes.

No middleware proposto, as aplicações fazem requisições através de uma interface superior. Essas requisições são feitas em alto nível, pois carregam apenas a informação necessária para serem executadas e não se preocupam com detalhes técnicos dos dispositivos envolvidos, já que as aplicações desconhecem as informações de baixo nível da rede. A interface encaminha as requisições a um analisador de requisições que irá examinar a mensagem, procurando identificar o emissor e o tipo da requisição.

Foram definidos três tipos de requisições: (I) cadastro, quando uma nova aplicação ingressa na rede; (II) Atuação, quando uma aplicação requisita alguma ação no ambiente, como ligar uma luz; e, (III) Coleta, quando uma aplicação deseja receber a informação coletada de algum dispositivo. Estes três tipos citados englobam a maior parte das funcionalidades necessárias para gerenciar sensores e atuadores em Casas Inteligentes [18].

Para cada tipo de requisição, o analisador de requisições dispara uma rotina de tratamento diferente. No caso da requisição ser do tipo cadastro, as informações da aplicação são guardadas na base de dados local. Em uma requisição do tipo atuação, é identificado o dispositivo, ou grupo de dispositivos, que a instrução deve ser entregue e que função

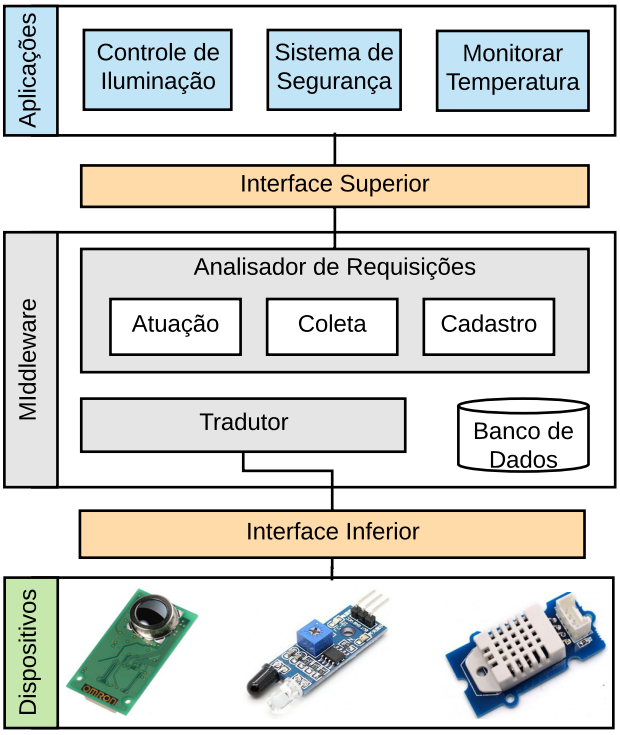


Figura 1. Arquitetura projetada para o Middleware.

deve ser exercida. Essas informações são repassadas para um tradutor com a intenção de gerar a instrução que será entregue aos dispositivos. Nas requisições de coleta, as informações requisitadas são buscadas na base de dados e devolvidas a aplicação que fez a requisição.

Neste cenário, o tradutor é encarregado de preparar a mensagem de atuação que será enviada ao dispositivo, mapeando a ação que será exercida para uma das funções do dispositivo, e receber as mensagens com informações coletadas pelos sensores para armazenar na base de dados. A interface inferior se encarrega de fazer a comunicação com os dispositivos. A seguir, cada um dos elementos do middleware será descrito.

### A. Aplicações

*Aplicações* referre-se as aplicações comuns presentes nos cenários de SH, como um aplicativo de controle residencial instalado em um smartphone ou smartTV, no qual o usuário pode monitorar e interagir com o ambiente através dos dispositivos conectados. Adicionalmente, *Aplicações* refere-se aos serviços prestados dentro do ambiente [19]. Rotinas inteligentes que controlam o ambiente sem a interação do usuário, como centrais de segurança que coletam informação de sensores de presença e câmeras e podem alertar a presença de intrusos, ou ainda, controles de luminosidade que gerenciam a intensidade de iluminação nos ambientes dependendo da hora do dia, temperatura ou época do ano, por exemplo.

O importante para a proposta é que todas essas aplicações e serviços comunicam-se com os dispositivos através do middleware, de uma forma comum e de alto nível, através de mensagens simples que possam identificar o emissor e sua requisição. Essas informações podem ser enviadas concatenadas, como ”app1/get/luz/sala”e encapsulada num pacote que deve respeitar o protocolo implementado na *Interface Superior*.

### B. Interface Superior

A *Interface Superior* é uma interface de comunicação para onde as aplicações enviam suas requisições. Essa interface deve ser capaz de se comunicar com as aplicações e serviços prestados na rede, podendo eles estarem conectados na rede local ou através da nuvem. A interface Superior é responsável por coletar as requisições das diversas aplicações e encaminhalas ao analisador de requisições. Devido a abstração, o analisador também não se preocupa em como a mensagem foi enviada, apenas com seu conteúdo. O middleware proposto é independente da estrutura definida pela *Interface Superior*, onde faz-se necessário somente a existência de uma padronização para esta funcionalidade. De maneira geral, a *Interface Superior* pode ser desde uma API (com tipos complexos) até um buffer que armazena requisições com estrutura pre-definida.

### C. Analisador de requisições

O *Analisador de Requisições* tem o trabalho de examinar a mensagem vinda da *Interface Superior* e identificar a aplicação ou serviço que fez a requisição e que tipo de requisição é essa. Com base no tipo de requisição o analisador irá iniciar uma rotina de tratamento diferente, que podem ser uma das seguintes: Cadastro, Atuação e Coleta. Foram escolhidas essas três rotinas de tratamento por serem de proposito geral e satisfazerem os requisitos necessários para o gerenciamento de Casas Inteligentes [18]:

- Cadastro: Quando uma aplicação nova tenta se conectar aos dispositivos, essa rotina trata de guardar na base de dados informações que sirvam para identificá-la. A *Aplicação* inicia a rotina realizando uma requisição de cadastro a *Interface Superior* (responsável por abstrair a comunicação entre as aplicações e os serviços prestados). A *Interface Superior*, por sua vez, encaminha a requisição para o *Analisador de Requisições*. Por fim, o *Analisador de Requisição* deve identificar o tipo de requisição (no caso, como sendo de cadastro) e deve realizar o cadastro da nova aplicação na *Base de Dados*. Assim, no futuro, é possível fazer uso das funcionalidades dessa nova aplicação.
- Atuação: Nessa rotina, o analisador de requisições verifica a função que será executada e que dispositivos serão acionados para executá-la. O analisador acessa a base de dados procurando identificar os dispositivos que tratam daquela função, por exemplo, em uma requisição do tipo ”app5/ligar/luz/sala”os dispositivos atuadores que tem função relacionada a luz e estão presentes no ambiente sala devem receber uma instrução do tipo ligar.
- Coleta: Quando o analisador detecta que a requisição é do tipo coleta, ele busca na base de dados pelas informações relacionadas a requisição e devolve essa informação para a Interface Superior, para que seja entregue a aplicação que fez a requisição. Caso a informação desejada não esteja presente na base de dados, os dados do sensor que deve ser acionado são enviados para o tradutor para que uma instrução de coleta seja enviada ao sensor, semelhante as instruções de atuação. Quando os dados retornam pela Interface Inferior, o tradutor os armazena na base de dados e repassa ao analisador, que verifica qual aplicação requisitou aqueles dados e informa a Interface Superior, os dados e o destinatário, para que a informação seja entregue.

### D. Tradutor

Quando o analisador envia informações sobre um dispositivo e a instrução que será encaminhada a ele, o tradutor é encarregado de verificar as funções implementadas daquele dispositivo e escolher qual função executa a instrução. A estrutura que informa, além das funções, outras características que definem o dispositivo é informada através de um arquivo XML. Esse arquivo XML é fornecido pelo administrador da rede sempre que um dispositivo é adicionado na rede. O seu formato pode variar, dependendo do formato de um arquivo XSD que valida as informações de todos os XMLs dos dispositivos e garante a presença de informações básicas necessárias para que as aplicações façam o bom uso dos dispositivos. Enquanto o XML contem a informação, o XSD define qual informação deve estar presente e seu tipo.

No XSD do middleware proposto foram definidos 11 tipos diferentes de tipos complexos a fim de englobar os diversos tipos de mensagens trocadas com os dispositivos inteligentes presentes em uma SH: *Device*, *base-sensor*, *binary-sensor*, *float-sensor*, *integer-sensor*, *string-sensor*, *base-actuator*, *binary-actuator*, *float-actuator*, *integer-actuator* e *string-actuator*. Na Figura 2 são mostrados parte dessas estruturas definidas no XSD descrever um sensor binário (as restrições de ocorrência foram omitidas por questão de espaço em texto).

A base *Device* engloba as características base de qualquer dispositivo, onde, além de definir o tipo de dispositivo (sensor

```
<xsd:complexType name="Device">
 <xsd:sequence>
  <xsd:element name="ID" type="xsd:int" />
  <xsd:element name="place" type="xsd:string"/>
  <xsd:element name="sensor" type="base-sensor" />
  <xsd:element name="actuator" type="base-actuator" />
 </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="base-sensor">
 <xsd:sequence>
  <xsd:element name="binary" type="binary-sensor" />
  <xsd:element name="integer" type="integer-sensor" />
  <xsd:element name="float" type="float-sensor" />
  <xsd:element name="string" type="string-sensor" />
 </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="binary-sensor">
 <xsd:sequence>
   <xsd:element name="type" type="xsd:string"/>
   <xsd:element name="value" type="xsd:boolean"/>
   <xsd:element name="get-value" type="xsd:int"/>
  </xsd:sequence>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
</xsd:complexType>
```

Figura 2. XSD definido.

ou atuador), tem-se um identificador único (*ID*) e o local que ele se encontra na SH (*place*). Esta estruturação permite que uma aplicação comunique-se com mais de um dispositivo inteligente de um mesmo tipo mas em locais da casa distintos. As demais estruturas visam atender as características de um sensor binário (por exemplo um sensor de presença), onde apenas é feita uma coleta do atual estado em que se encontra o sensor.

### *E. Interface Inferior*

Essa entidade é responsável por se comunicar com os dispositivos presentes no ambiente SH. Semelhante a *Interface Superior*, nela estão contidas as interfaces de comunicação que abranjam todos os dispositivos presentes (Ex: ZigBee, Bluetooth LE). Similarmente a *Interface Superior*, o middleware proposto é independente da *Interface Inferior* aplicada.

### *F. Visão Geral do Middleware*

A partir da descrição do middleware proposto, esta seção apresenta uma visão geral do mesmo ao ser aplicado em um possível cenário real de SH, ilustrado na Figura 3, onde há três dispositivos inteligentes (uma câmera de segurança, uma lâmpada inteligente e um sensor de temperatura) com cada um deles comunica-se com uma aplicação específica. Neste cenário tem-se uma residência composta de uma WHN e interligada um ambiente de computação em névoa (*Fog Computing*), o qual executa as aplicações que provêm serviços na SH. Portanto, o middleware proposto habilita a comunicação entre esses dispositivos inteligentes e as aplicações presente no roteador da WHN.

É válido ressaltar que a aplicabilidade do middleware proposto é mais abrangente do que o cenário ilustrado na Figura 3. O middleware proposto pode estar presente em diversos ambientes, como em smartphones, na *Fog* ou no próprio

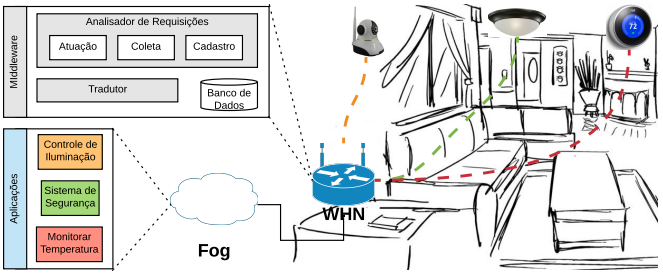


Figura 3. Visão Geral do Middleware em uma SH.

roteador da WHN, contanto que este seja o intermediário entre os dispositivos e o ambiente que execute as aplicações (que por sua vez podem executar nos mesmos ambientes que citados), atendendo assim as recomendações descritas na literatura para o contexto de SHs [20].

## IV. Experimentos

O cenário teste foi representado através de uma rede com sensores e atuadores, de tipos frequentemente empregados em ambientes de *smart home*, como sensor de coleta de temperatura e umidade, equipados em microcontroladores Arduino e módulos de comunicação NRF24L01. Uma Raspberry Pi 3 (com um processador Quad Core 1.2GHz e 1GB de mmeória RAM) foi utilizada para fazer a comunicação com os dispositivos.

Neste seção serão descritos os experimentos de medição de escalabilidade do middleware proposto, para isso os experimentos focaram em estressar o middleware com uma quantidade de requisições variada. Os experimentos tiveram um número variado de dispositivos presentes no cenário, que foram de 5, 15, 30, 50 e 100 dispositivos. Para cada quantidade de dispositivos foram feitos testes alterando o número de aplicações simuladas que fazem requisições a esses dispositivos, onde foram executadas 5, 10, 15, 30, 50 e 100 aplicações, fazendo cada uma 10 requisições para cada um dos dispositivos no teste. Portanto, em um cenário com 50 dispositivos e 10 aplicações, foram geradas 5 mil requisições a serem tradas pelo middleware. Cada cenário foi testado 30 vezes com um intervalo de confiança de 95%.

Durante os experimentos, foram utilizados dispositivos IoT heterogêneos, como sensores de temperatura, umidade e atuadores controlados por uma Raspberry Pi 3. Os critérios de escolha desses dispositivos basearam-se em sua representatividade no contexto de casas inteligentes e compatibilidade com o middleware proposto. A rede utilizada foi composta por uma WHN, com comunicação baseada em módulos NRF24L01 para os sensores e ZigBee para dispositivos maiores.

### *A. Resultado dos Experimentos*

Como forma de quantificar o desempenho obtido durante os testes, foi atribuída como métrica o tempo desde o recebimento da instrução pela aplicação até o envio da ação desejada para o dispositivo, ou seja, o tempo medido considerou apenas a

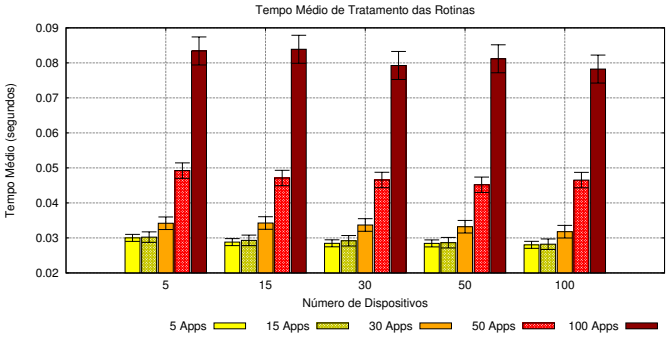


Figura 4. Tempo médio de tratamento de cada requisição.

parte que cabe ao middleware. Os resultados são apresentados na Figura 4.

Como pode ser visto na Figura 4, a quantidade de dispositivos na rede é o fator que mais interfere no tempo que o *middleware* levou para tratar cada requisição. O que já era esperado, pois, independente de quantas aplicações fazem requisições, o tempo de tratamento de cada requisição será maior se for necessário verificar um número maior de dispositivos. Adicionalmente, é válido ressaltar que o tempo mensurado é totalmente dependente do hardware utilizado (nos experimentos, uma Raspberry Pi 3). Desta forma, em casos com um número superior de dispositivos e aplicações, pode-se utilizar hardware com maior capacidade de processamento a fim de melhorar o tempo de tratamento das requisições, bem como integrar o middleware proposto com uma abordagem de computação em névoa (assim como descrito na Seção III-F).

Os dados mostrados nos experimentos sugerem a viabilidade de utilização do middleware proposto para habilitar uma comunicação abstrata entre aplicações e dispositivos inteligentes em SHs. A implantação do middleware proposto gera um pequeno impacto no tempo de comunicação, variando entre 30 ms (no melhor dos casos) e 80 ms (no pior dos casos), seguindo o padrão das aplicações mais comuns em SHs [19].

## V. Conclusão

IoT é um paradigma que cresceu bastante, a medida que os usuários utilizam um número cada vez maior de dispositivos diariamente, tanto no trabalho e muitas vezes em seus lares.

As residências hoje em dia estão repletas de dispositivos heterogêneos, o que dificulta a comunicação e a implantação de aplicações e serviços nas redes domésticas. Para auxiliar na utilização desses dispositivos, foi proposto um middleware para abstrair a comunicação entre eles e facilitar sua utilização pelas aplicações e serviços por meio de uma interface de alto nível. Através de um cenário de testes, foi concluído que o modelo proposto tem desempenho adequado, pois permitiu que os dispositivos fossem acionados por meio de instruções de alto nível e com um tempo de resposta aceitável (cerca de 30 milissegundos). Como trabalhos futuros, será importante estender a arquitetura com um broker, permitindo um melhor desempenho.

A adição de uma etapa extra de segurança para as requisições que acessam o *middleware*, embora representem um custo adicional de processamento, é importante para garantir que somente aplicações e serviços autorizados tenham acesso aos dispositivos presentes na *smart home* [21].

## Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) ($N^o$ 303877/2021-9 e $N^o$ 405976/2022-4) pelo apoio financeiro e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## Referências

[1] P. Pradeep, S. Krishnamoorthy, and A. V. Vasilakos, "A holistic approach to a context-aware iot ecosystem with adaptive ubiquitous middleware," *Pervasive and Mobile Computing*, vol. 72, p. 101342, 2021.

[2] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira, "A similarity model for virtual networks negotiation," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ser. SAC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 489–494. [Online]. Available: https://doi.org/10.1145/2554850.2554963

[3] A. L. Portela, R. A. Menezes, W. L. Costa, M. M. Silveira, L. F. Bittecnourt, and R. L. Gomes, "Detection of iot devices and network anomalies based on anonymized network traffic," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–6.

[4] K. L. D. Almeida, L. Lessa, A. B. S. Peixoto, R. L. Gomes, and J. Celestino, "Kidney Failure Detection Using Machine Learning Techniques," in *Proc. of the 8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020)*, ser. Proc. of the 8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020), F. M. Mena, U. A. D. Yucatan, Mexico, E. Duarte, F. U. of Parana, and Brazil, Eds. Cancún, Mexico: Candy E. Sansores, Universidad del Caribe, Mexico, Nazim Agoulmine, IBISC Lab, University of Evry - Paris-Saclay University, Jan. 2020, pp. 1–8. [Online]. Available: https://hal.science/hal-02495264

[5] M. S. Souza, S. E. S. B. Ribeiro, V. C. Lima, F. J. Cardoso, and R. L. Gomes, "Combining regular expressions and machine learning for sql injection detection in urban computing," *Journal of Internet Services and Applications*, vol. 15, no. 1, p. 103–111, Jul. 2024. [Online]. Available: https://journals-sol.sbc.org.br/index.php/jisa/article/view/3799

[6] M. C. Ferreira, S. E. Ribeiro, F. V. Nobre, M. L. Linhares, T. P. Araújo, and R. L. Gomes, "Mitigating measurement failures in throughput performance forecasting," in *2024 20th International Conference on Network and Service Management (CNSM)*, 2024, pp. 1–7.

[7] I. Pimenta, D. Silva, E. Moura, M. Silveira, and R. L. Gomes, "Impact of data anonymization in machine learning models," in *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, ser. LADC '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 188–191. [Online]. Available: https://doi.org/10.1145/3697090.3699865

[8] M. Silveira, D. Santos, M. Souza, D. Silva, M. Mesquita, J. Neto, and R. L. Gome, "An anonymization service for privacy in data mining," in *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, ser. LADC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 214–219. [Online]. Available: https://doi.org/10.1145/3615366.3625074

[9] A. L. C. Portela, S. E. S. B. Ribeiro, R. A. Menezes, T. P. d. Araújo, and R. L. Gomes, "T-for: An adaptable forecasting model for throughput performance," *IEEE Transactions on Network and Service Management*, vol. 21, no. 3, pp. 2791–2801, 2024.

[10] M. Silva, S. Ribeiro, V. Carvalho, F. Cardoso, and R. L. Gomes, "Scalable detection of sql injection in cyber physical systems," in *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, ser. LADC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 220–225. [Online]. Available: https://doi.org/10.1145/3615366.3625075

[11] G. Bouloukakis, K. Benson, L. Scalzotto, P. Bellavista, C. Grant, V. Issarny, S. Mehrotra, I. Moscholios, and N. Venkatasubramanian, "Priodex: a data exchange middleware for efficient event prioritization in sdn-based iot systems," *ACM Transactions on Internet of Things*, vol. 2, no. 3, pp. 1–32, 2021.

[12] M. A. Abbasi, Z. A. Memon, N. M. Durrani, W. Haider, K. Laeeq, and G. A. Mallah, "A multi-layer trust-based middleware framework for handling interoperability issues in heterogeneous iots," *Cluster Computing*, vol. 24, no. 3, pp. 2133–2160, 2021.

[13] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of iot for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, Oct 2013.

[14] D. Chilcañán, P. Navas, and M. Escobar, "Virtual assistant for iot process management, using a middleware," in *Proceedings of the 2018 2Nd International Conference on Algorithms, Computing and Systems*, ser. ICACS '18. New York, NY, USA: ACM, 2018, pp. 209–213. [Online]. Available: http://doi.acm.org/10.1145/3242840.3242875

[15] K. E. Benson, G. Wang, N. Venkatasubramanian, and Y. Kim, "Ride: A resilient iot data exchange middleware leveraging sdn and edge cloud resources," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2018, pp. 72–83.

[16] M. A. da Cruz, J. J. Rodrigues, P. Lorenz, V. V. Korotaev, and V. H. C. de Albuquerque, "In. iot—a new middleware for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7902–7911, 2020.

[17] O. Anya and H. Tawfik, *Toward a Cognitive Middleware for Context-Aware Interaction in Smart Homes*. Cham: Springer International Publishing, 2018, pp. 41–54.

[18] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on smart homes system using internet-of-things," in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, April 2015, pp. 0330–0335.

[19] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "Homonit: Monitoring smart home apps from encrypted traffic," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1074–1088. [Online]. Available: http://doi.acm.org/10.1145/3243734.3243820

[20] T. K. Hui, R. S. Sherratt, and D. D. Sánchez, "Major requirements for building smart homes in smart cities based on internet of things technologies," *Future Generation Computer Systems*, vol. 76, pp. 358–369, 2017.

[21] M. M. Silveira, A. L. Portela, R. A. Menezes, M. S. Souza, D. S. Silva, M. C. Mesquita, and R. L. Gomes, "Data protection based on searchable encryption and anonymization techniques," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–5.