UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Uma Abordagem para a Integração de Diferentes Fontes de Dados Provenientes de Redes de Sensores no Contexto de Internet das Coisas

Vinícius Aires Barros

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Vinícius Aires Barros

Uma Abordagem para a Integração de Diferentes Fontes de Dados Provenientes de Redes de Sensores no Contexto de Internet das Coisas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Júlio Cezar Estrella

USP – São Carlos
Outubro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

A101a   Aires Barros, Vinícius
              Uma abordagem para a Integração de Diferentes
Fontes de Dados Provenientes de Redes de Sensores
no Contexto de Internet das Coisas / Vinícius Aires
Barros; orientador Júlio Cezar Estrella. -- São
Carlos, 2019.
              104 p.

              Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2019.

              1. Internet das Coisas. 2. Interoperabilidade.
3. Avaliação de Desempenho. I. Cezar Estrella,
Júlio, orient. II. Título.

Vinícius Aires Barros

**An Approach for the Integration of Different Data Sources
from Sensor Networks in the Internet of Things Context**

Dissertation submitted to the Institute of Mathematics
and Computer Sciences – ICMC-USP – in
accordance with the requirements of the Computer
and Mathematical Sciences Graduate Program, for
the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Prof. Dr. Júlio Cezar Estrella

USP – São Carlos
October 2019

*Dedico este trabalho a Deus, por ser essencial em minha vida,
autor de meu destino, meu guia, socorro presente na hora da angústia.*

AGRADECIMENTOS

Primeiramente agradeço à Deus por tudo que tem feito em minha vida, sem a sua imensa misericórdia e seus ensinamentos através da sua palavra eu não estaria onde estou. Agradeço aos meus familiares, em especial à minha mãe Sebastiana A. Santana, pelo apoio em todos os momentos e aos meus colegas e professores do Laboratório de Sistemas Distribuídos e Programação Concorrente (LaSDPC) pelos instantes de descontração e amizade. Agradeço também o meu orientador Júlio C. Estrella, por me permitir conduzir este projeto e pelo suporte oferecido durante o mestrado. Agradeço imensamente ao meu colega Luiz H. Nunes, pela sua atenção e auxílio durante as etapas iniciais desta pesquisa. Agradeço também a professora Sarita M. Bruschi pela sua parceria nos artigos publicados, contribuindo imensamente na forma de como conduzir a análise dos experimentos. Faço agradecimento em especial a minha amiga Adrienne A. da Silva por me ter me ajudado com a revisão gramatical desta dissertação, pela sua amizade e paciência. Faço um agradecimento em especial à Angela C. P. Giampetro por ter me auxiliado com a revisão dos artigos em inglês. Agradeço também os alunos Herik M. Serra, Leonardo B. Prates e Sergio A. B. Junior, os quais auxiliei durante seus TCCs, pela parceria que resultaram em excelentes trabalhos. Agradeço em especial os meus colegas de faculdade/trabalho pela amizade e momentos de descontração que tornaram momentos de dificuldades mais alegres. Agradeço à CAPES (PROEX-10095318/M) e FAPESP (11/09524-7) por financiar esta pesquisa e o Instituto de Pesquisas Eldorado por custear a publicação na revista *Mobile Information System* e no Simpósio Brasileiro de Sistemas Multimídia e Web (*WebMedia*). Por fim, dedico este trabalho aos educadores que passaram pela minha vida, pois seus ensinamentos e votos de confiança me fizeram chegar onde eu jamais imaginaria.

*“Existem homens que lutam um dia e são bons; existem outros que lutam um ano e são melhores;
existem aqueles que lutam muitos anos e são muito bons.
Porém, existem os que lutam toda a vida. Estes são os imprescindíveis.”*
(Bertolt Brecht)

RESUMO

BARROS, V. A. **Uma Abordagem para a Integração de Diferentes Fontes de Dados Provenientes de Redes de Sensores no Contexto de Internet das Coisas**. 2019. 101 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Internet das Coisas (*IoT*) tem como principal característica a conexão em rede entre dispositivos como sensores, *smartphones* e *wearables*, com a finalidade de coletar informações dos ambientes físicos em que se encontram. Um desafio relacionado é a falta de padronização da comunicação entre dispositivos e de um mecanismo que realiza o processamento, armazenamento e recuperação de dados de forma simplificada, sendo, portanto, um grande desafio a interoperabilidade. Esta dissertação introduz o *Internet of Things Data as a Service Module (IoTDSM)*, uma abordagem que baseia-se no modelo de *Data as a Service (DaaS)*, que tem como objetivo auxiliar a gestão de fontes de dados de sensores heterogêneos, que independem da sua origem, formato ou sistema de banco de dados utilizado. Além disso, é apresentado o *Internet of Things Multi-Protocol Message Broker (IoTM²B)*, uma extensão do *IoTDSM* que permite a integração com diferentes protocolos de comunicação utilizados em redes de sensores. Neste sentido, para avaliação desta pesquisa foi utilizado a metodologia de avaliação de desempenho, a qual contribuiu para a identificação das limitações dos mecanismos propostos. Além disso, neste trabalho diferentes cenários de avaliação foram conduzidas: (i) avaliação de desempenho do *Middleware Global Sensor Network (GSN)*, que auxiliou na definição dos demais cenários de avaliação abordados; (ii) avaliação de desempenho *IoTDSM* utilizando diferentes sistemas de bancos de dados (*PostgreSQL* ou *MongoDB*) e formatos de dados (*JSON* ou *XML*), para o processamento de dados climáticos; (iii) avaliação de desempenho uma estratégia *IoTM²B* a qual permitiu a integração do *IoTDSM* com os protocolos *HTTP*, *MQTT* e *CoAP* em um ambiente de comunicação *Machine-to-Machine (M2M)* e Computação em Nuvem; e, (iv) avaliação de uma arquitetura que realiza a classificação de emoções de usuários em um ambiente de casa inteligente. Por fim, é feita uma discussão sobre os resultados obtidos, além de demonstrar a possibilidade da integração do *IoTDSM* e *IoTM²B* com diferentes formatos de dados, estratégias de armazenamento e protocolos de comunicação.

Palavras-chave: Internet das Coisas; Interoperabilidade; Avaliação de Desempenho.

ABSTRACT

BARROS, V. A. **An Approach for the Integration of Different Data Sources from Sensor Networks in the Internet of Things Context**. 2019. 101 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Internet of Things (IoT) has a primary characteristic of the network connection among devices e.g., sensors, smartphones, and wearables, in order to collect information about the physical environments. A related challenge is the non-standardization of communication among devices and a mechanism that performs data processing, storage, and retrieval in a simplified way, hence interoperability a big problem. This thesis introduces the Internet of Things Data as a Service Module (IoTDSM), an approach that is based on the Data as a Service (DaaS) model, which aims to assist the management of heterogeneous sensor data sources, regardless the data source, format or database system utilized. In addition, the Internet of Things Multi-Protocol Message Broker (IoTM²B) is presented, an extension of IOTDSM that allows integration with different sensor networks communication protocols. The performance evaluation methodology was utilized to evaluate this research, which contributed to the identification of the limitations of the proposed mechanisms. Furthermore, different evaluation scenarios were employed: (i) performance evaluation of the Global Sensor Network (GSN), which assisted the execution of the others evaluation scenarios; (ii) performance evaluation of IOTDSM employing different database systems (PostgreSQL or MongoDB) and input/output data formats (JSON or XML) for the processing world wide climatic data; (iii) performance evaluation of IoTM²B strategy which allowed the integration of IOTDSM with the HTTP, MQTT and CoAP protocols in a Machine-to-Machine (M2M) and Cloud Computing environment; and (iv) evaluation of an architecture for emotion recognition on smart home environment. Finally, a discussion is made about the results obtained, besides demonstrating the possibility of the integration of IOTDSM and IoTM²B with different data formats, storage strategies, and communication protocols.

Keywords: Internet of Things; Interoperability; Performance Evaluation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de Funcionamento de Rede de Sensores (Adaptado de (AKYILDIZ <i>et al.</i> , 2002)).	31
Figura 2 – Estatísticas de Crescimento da <i>IoT</i> (Adaptado de (WITCHALLS; CHAMBERS, 2013)).	32
Figura 3 – Estatísticas sobre o Impacto do Crescimento da <i>IoT</i> (Adaptado de (WITCHALLS; CHAMBERS, 2013)).	32
Figura 4 – Total de Pesquisas Realizadas no <i>Google</i> em Relação à Ciência (Fonte: <i>Google</i>).	33
Figura 5 – Estrutura Arquitetural da <i>IoT</i> (Adaptado de (NUNES, 2018)).	34
Figura 6 – Arquitetura <i>ViSIoT</i> (Adaptado de (NUNES <i>et al.</i> , 2016)).	56
Figura 7 – Diagrama de Sequência da Arquitetura <i>ViSIoT</i> (Adaptado de (NUNES <i>et al.</i> , 2016)).	57
Figura 8 – Etapas de Execução da Dissertação.	58
Figura 9 – Diagrama de Caso de Uso do Módulo Repositório.	59
Figura 10 – Diagrama de Arquitetura do <i>IoTDSM</i>	60
Figura 11 – Diagrama de Sequência do <i>IoTDSM</i>	61
Figura 12 – Modelo Entidade Relacionamento.	63
Figura 13 – Modelo Orientado a Documentos.	64
Figura 14 – Diagrama de Arquitetura do <i>IoT^{M2B}</i>	65
Figura 15 – Diagrama de Sequência do <i>IoT^{M2B}</i>	66
Figura 16 – Setup Experimental.	69
Figura 17 – Carga de Trabalho com 10 Requisições.	70
Figura 18 – Carga de Trabalho 100 Requisições.	71
Figura 19 – Carga de Trabalho com 1.000 Requisições.	71
Figura 20 – Setup Experimental.	74
Figura 21 – Análise de Influência de Fatores Principais.	75
Figura 22 – Tempo de Resposta <i>JSON</i> x <i>XML</i>	76
Figura 23 – Fluxo de Comunicação (Experimento A).	78
Figura 24 – Setup Experimental (Experimento A).	78
Figura 25 – Fluxo de Comunicação (Experimento B).	79
Figura 26 – Setup Experimental (Experimento B).	79
Figura 27 – Fluxo de Comunicação (Experimento C).	79
Figura 28 – Tempo de Resposta (Experimento A).	80
Figura 29 – Tempo de Resposta (Experimento B).	81

Figura 30 – Tempo de Resposta (Experimento C1).	82
Figura 31 – Tempo de Resposta (Experimento C2).	83
Figura 32 – Arquitetura ENLACE.	84
Figura 33 – Casa Inteligente com Arquitetura ENLACE.	84
Figura 34 – Tempo de resposta dos serviços <i>HRA</i> + <i>FA</i> em um dispositivo local.	86
Figura 35 – Tempo de resposta do serviço <i>HRA</i> em um dispositivo local.	86
Figura 36 – Tempo de resposta do serviço <i>FA</i> em um dispositivo local.	86
Figura 37 – Tempo de resposta dos serviços <i>HRA</i> + <i>FA</i> em um servidor local.	87
Figura 38 – Tempo de resposta do serviço <i>HRA</i> em um servidor local.	87
Figura 39 – Tempo de resposta do serviço <i>FA</i> em um servidor local.	87
Figura 40 – Tempo de resposta dos serviços <i>HRA</i> + <i>FA</i> em um ambiente de nuvem.	88
Figura 41 – Tempo de resposta do serviço <i>HRA</i> em um ambiente de nuvem.	88
Figura 42 – Tempo de resposta do serviço <i>FA</i> em um ambiente de nuvem.	88

LISTA DE TABELAS

Tabela 1 – Principais características dos protocolos HTTP, MQTT e CoAP.	37
Tabela 2 – Tabela Comparativa entre <i>Middlewares</i> (Adaptado de (RAZZAQUE <i>et al.</i> , 2016)).	39
Tabela 3 – Formulário de Extração de Dados (* campos obrigatórios).	46
Tabela 4 – Relação das Principais Características dos Trabalhos Relacionados.	53
Tabela 5 – <i>Endpoints</i> disponibilizados pela <i>API</i>	64
Tabela 6 – Ferramentas utilizadas para a construção do <i>IoTDSM</i>	65
Tabela 7 – Ferramentas utilizadas para a construção do <i>IoT²B</i>	66
Tabela 8 – Lista de Parâmetros do Experimento.	68
Tabela 9 – Configurações de <i>Hardware</i> (Geradores de Carga).	69
Tabela 10 – Comparação de <i>Hardware</i> (<i>Raspberry Pi</i>).	69
Tabela 11 – Experimental Design.	72
Tabela 12 – Parâmetros Fixos de Experimento.	73
Tabela 13 – Especificações de <i>Hardware</i>	73
Tabela 14 – Desenho de Experimento.	77
Tabela 15 – Configurações de <i>Hardware</i>	80
Tabela 16 – Experimento A (Tempo μs).	80
Tabela 17 – Experimento B (Tempo μs).	81
Tabela 18 – Experimento C1 (Tempo μs).	81
Tabela 19 – Experimento C2 (Tempo μs).	82
Tabela 20 – Configuração do Ambiente Experimental.	85

LISTA DE ABREVIATURAS E SIGLAS

OWM	<i>Open Weather Map</i>
API	Application Programming Interface
AWS	Amazon Web Services
CA4IoT	Context Awareness for Internet of Things
CASCOM	Context Aware Sensor Configuration Model
CoAP	Constrained Application Protocol
CoT	Cloud Of Things
CPU	Central Processing Unit
DaaS	Data as a Service
DQS-Cloud	Data Quality-Aware Sensor Cloud
EB	Exabyte
ER	Entidade e Relacionamento
FIT IoT-LAB	Future Internet of the Things IoT-LAB
GCP	Google Cloud Platform
GPS	Global Positioning System
GSMC	Green School Motorcycle
GSN	Global Sensor Network
GUI	Graphical User Interface
HTTP	<i>Hypertext Transfer Protocol</i>
IaaS	Infrastructure as a Service
Id	Identity Document
IDC	International Data Corporation
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoTDSM	Internet of Things Data as a Service Module
IoTM ² B	Internet Multi-Protocol Message Broker
IP	Internet Protocol
JavaEE	Java Enterprise Application Server
JSON-LD	JSON for Linking Data
KDS	Knowledge Discovery in Services

LSM-Light	Linked Stream Middleware Light
M2M	<i>Machine to Machine</i>
M3 Framework	Machine-to-Machine Measurement Framework
MQTT	<i>Message Queuing Telemetry Transport</i>
NIC	National Intelligence Council
OASIS	Organization for the Advancement of Structured Information Standards
OMNet++	Objective Modular Network Testbed in C++
OpenIoT	Open Internet of Things
OWL	Web Ontology Language
OWL-S	Semantic Markup for Web Services
PaaS	Platform as a Service
QoS	<i>Quality of Service</i>
REST	Representational State Transfer
RFID	Radio-Frequency IDentification
S2aaS	Sensing as a Service
SaaS	Software as a Service – SaaS
SD & UM	Service Delivery & Utility Manager
SE	Stream Elements
SenaaS	Sensor as a Service
SenML	Sensor Markup Language
SensorML	Sensor Model Language
SGBD	Sistema de Gerenciamento de Banco de Dados
SGS	Semantic Gateway as Service
SIGHTED	Semantic InteGration of HeTerogeneous SEnsor Data
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOCRADES	Service Oriented Architecture in der Automatisierungstechnik
SPARQL	SPARQL Query Language for RDF
SQL	Structured Query Language
SSN	Semantic Sensor Network
SSNO	Semantic Sensor Network Ontology
TB	Terabyte
TCP	Transmission Control Protocol
URI	Universal Resource Identifier
WoT	Web of Things
WoT-ESN	Web of Things - Emerging Sensor Network
WSDL	Web Services Description Language

WSN	Wireless Sensor Networks
X-GSN	eXtensible Global Sensor Network
XMPP	eXtensible Messaging and Presence Protocol
ZB	Zettabyte

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Contextualização	26
1.2	Motivação e Objetivo	26
1.3	Estrutura da Dissertação	27
2	REFERENCIAL TEÓRICO	29
2.1	Considerações Iniciais	29
2.2	<i>Internet</i> das Coisas	29
2.3	Computação em Nuvem e <i>Internet</i> das Coisas	41
2.4	Considerações Finais	43
3	TRABALHOS RELACIONADOS	45
3.1	Considerações Iniciais	45
3.2	Revisão da Literatura	45
3.3	Considerações Finais	53
4	DESENVOLVIMENTO DO TRABALHO	55
4.1	Considerações Iniciais	55
4.2	Projeto de Pesquisa	55
4.3	Metodologia Adotada para o Desenvolvimento da Pesquisa	57
4.4	Considerações Finais	66
5	AVALIAÇÕES EXPERIMENTAIS	67
5.1	Considerações Iniciais	67
5.2	Avaliação de Desempenho	67
5.3	Avaliação de Desempenho do <i>Middleware GSN</i>	67
5.4	Avaliação de Desempenho do <i>IoTDSM</i>	70
5.5	Avaliação de Desempenho do <i>IoT^{M2B}</i>	77
5.6	Avaliação de Desempenho da Arquitetura <i>ENLACE</i>	82
5.7	Considerações Finais	87
6	PUBLICAÇÕES E SUBMISSÕES	89
7	CONCLUSÕES	91
7.1	Contribuições e Limitações do Trabalho	91

7.2   Trabalhos Futuros 92

REFERÊNCIAS 95

INTRODUÇÃO

A tecnologia encontra-se em constante evolução, o que permite uma diminuição progressiva no preço de dispositivos móveis, ao mesmo tempo em que ocorre o aumento do poder de processamento e armazenamento. Diferentes aplicações são desenvolvidas com base no monitoramento das informações coletadas por estes dispositivos, como por exemplo, aplicações nas áreas médicas, controle do trânsito e agricultura. Neste sentido, surge o termo *Internet das Coisas* (*Internet of Things (IoT)*) que refere-se à conexão de aparelhos como sensores, relógios e dispositivos móveis à *Internet*. Essa nova percepção trouxe a possibilidade de uma troca efetiva de informações entre um conjunto de dispositivos conectados.

A *IoT* é considerada o futuro da *Internet*, pois a modifica, tal que ela deixa de ser composta predominantemente por usuários e passa ser constituída também por bilhões de dispositivos interconectados. A comunicação entre objetos na *IoT* é realizada por meio de protocolos de rede como o Protocolo de Controle de Transmissão (*Transmission Control Protocol (TCP)*) e o Protocolo de Interconexão (*Internet Protocol (IP)*), por meio de conexões de rede com ou sem fio. Estas características facilitam a integração da *IoT* com outras aplicações, como, por exemplo, *softwares* com base em Computação em Nuvem e Arquiteturas Orientadas a Serviços (*SOA*) (ATZORI; IERA; MORABITO, 2010).

A capacidade de coletar, armazenar e processar dados de diferentes fontes e formatos é um tópico de estudo presente na *IoT* (ATZORI; IERA; MORABITO, 2010). Neste sentido, diferentes *middlewares* são utilizados para intermediar a coleta de dados, como por exemplo, o *Global Sensor Network (GSN)*¹, *Open Internet of Things (OpenIoT)*² e o *Xively*³. A integração e padronização da comunicação entre diferentes dispositivos e fontes de dados é um desafio presente em *IoT* (CHEN; MAO; LIU, 2014). Uma possível solução consiste no desenvolvimento de uma abordagem que padronize a comunicação entre dados heterogêneos, além de armazenar

---

¹ <https://github.com/LSIR/gsn/>

² <http://www.openiot.eu/>

³ <https://www.xively.com/>

e recuperar dados de sensores de forma simplificada.

1.1 Contextualização

Atualmente, a *IoT* é amplamente aplicada para o monitoramento de ambientes físicos de forma autônoma (BARI; MANI; BERKOVICH, 2013). A combinação de diferentes tipos de sensores e sistemas embarcados conectados em rede possibilitou uma melhoria no monitoramento de ambientes físicos por meio da tecnologia (FAN; ZHOU, 2011) (PARWEKAR, 2011). Deve-se salientar que a aplicação dos conceitos da *IoT* é bastante vasta, um exemplo consiste no desenvolvimento de uma rede de sensores para o gerenciamento do consumo de energia elétrica (XU; HE; LI, 2014). Além disso, ela pode ser utilizada para a coleta de dados e reconhecimento de padrões que, após classificados, podem auxiliar na previsão de catástrofes naturais e o congestionamento do trânsito (AAZAM; HUH, 2014). Portanto, para facilitar a comunicação entre rede de sensores com os demais serviços conectados à *Internet* são utilizados *middlewares*, pois eles fornecem uma forma simplificada de integração com diferentes aplicações (PERERA *et al.*, 2014).

Um dos desafios enfrentados em relação a *IoT* diz respeito ao gerenciamento eficiente dos dados coletados pela rede de sensores que, de acordo com Zaslavsky, Perera e Georgakopoulos (2013), ultrapassam o volume de dados na casa do *Exabyte* (EB). Entretanto, as soluções em termos dessa tecnologia ainda estão em desenvolvimento e apresentam desafios ainda não superados, como a aquisição redundante de dados, a heterogeneidade do ambiente, a topologia dinâmica da rede e os meios de comunicação que apresentam problemas quanto à confiabilidade, segurança, qualidade e interoperabilidade (PERERA *et al.*, 2014) (BHUYAN *et al.*, 2010). Dado o contexto apresentado, esta dissertação contribui com os desafios relacionados à interoperabilidade em *IoT*, através da definição de uma abordagem que fornece mecanismos distintos para integração com diferentes fontes de dados de sensores.

1.2 Motivação e Objetivo

Esta dissertação tem como propósito apresentar uma abordagem para a obtenção de dados heterogêneos provenientes de sensores, a fim de contribuir com a integração com diferentes fontes de dados e auxiliar no armazenamento e recuperação destes. Para isso, é proposta a abordagem *Internet of Things Data as a Service Module* (*IoTDSM*), que tem como objetivo auxiliar a interoperabilidade de aplicações *IoT*, possibilitando a integração com diferentes fontes ou formatos de dados e sistemas de bancos de dados. Além disso, é apresentada uma extensão do *IoTDSM* chamada *Internet of Things Multi-Protocol Message Broker* (*IoTM²B*) que permitiu a integração com diferentes protocolos de comunicação utilizados em *IoT* (*HTTP*, *MQTT* e *CoAP*). Esta dissertação aborda os seguintes objetivos:

1. Criar uma interface de comunicação (*API RESTful*) para a integração com diferentes fontes de dados de sensores heterogêneos;
2. Elaborar um repositório de dados que possibilite a recuperação de informações independente do sistema de banco de dados utilizado (*SQL* ou *NoSQL*);
3. Definir um modelo de armazenamento para organização do repositório de dados, que possibilite a inserção, recuperação e conversão de dados heterogêneos;
4. Apresentar uma proposta de integração multi-protocolo que permite a integração com protocolos de comunicação distintos; e
5. Avaliar o desempenho dos mecanismos propostos.

1.3 Estrutura da Dissertação

Esta dissertação está estruturada da seguinte forma: O Capítulo 2 apresenta o referencial teórico desta dissertação, destacando os conceitos básicos relacionados a *Internet das Coisas*, Computação em Nuvem e Nuvem de Coisas. Em seguida, o Capítulo 3 apresenta os trabalhos relacionados, expondo a metodologia adotada para a definição da *string* de busca e seleção dos estudos primários. Acrescenta-se também uma análise crítica sobre os trabalhos relacionados e uma tabela comparativa. Por sua vez, o Capítulo 4 apresenta o desenvolvimento do trabalho que consiste na apresentação do *IoTDSM* e *IoT²M²B*; Na sequência, o Capítulo 5 apresenta os cenários de avaliação da abordagem proposta, além de discutir os resultados obtidos através de uma avaliação de desempenho e métodos estatísticos. O Capítulo 6 lista os artigos aceitos para publicação e os que estão em processo de avaliação; Por fim, no Capítulo 7 são discutidas as conclusões obtidas e as perspectivas de trabalhos futuros.

REFERENCIAL TEÓRICO

2.1 Considerações Iniciais

Este capítulo apresenta os referenciais teóricos relacionados a esta dissertação. Serão expostos ao longo das seções, os conceitos relacionados à Internet das Coisas, Redes de Sensores e Computação em Nuvem. Além disso, discute-se os desafios relacionados e expectativas da indústria de TI com relação à estas tecnologias.

2.2 *Internet* das Coisas

Internet das Coisas refere-se a uma novo conceito relacionado a interconexão de tecnologias e objetos. Essa abordagem ao longo do tempo proporcionou a definição do conceito de rede global de dispositivos (KORESHOFF; ROBERTSON; LEONG, 2013). É preciso salientar que o propósito deste novo paradigma é promover um mundo melhor para os seres humanos, isso porque os objetos presentes no dia a dia das pessoas serão capazes de capturar informações sem que haja instruções explícitas sobre o seu funcionamento (DOHR *et al.*, 2010). Assim, em uma definição simplificada, a *IoT* visa permitir que pessoas e objetos estejam conectados por meio de diferentes caminhos, redes ou serviços (VERMESAN *et al.*, 2011) e (NUNES, 2018).

Essas tecnologias são comumente associada ao conceito de rede global de dispositivos, que consiste em bilhões de objetos conectados em rede com endereçamento único e que se baseiam nos protocolos de comunicação tradicionais da *Internet* (BASSI; HORN, 2008). Esta nova concepção tem permitido a conexão entre objetos do cotidiano com outros dispositivos e serviços, com a finalidade de produzir resultados desejáveis. Além disso, os objetos presentes na *IoT* são os mais diversos possíveis, tais como sensores, atuadores, *smartphones*, etc (WHITMORE; AGARWAL; XU, 2015) (ATZORI; IERA; MORABITO, 2010). Tais características têm tornado essa tecnologia foco de diversas pesquisas e de um número considerável de discussões

sobre os impactos que o crescimento dessa nova área causará na sociedade.

Devido à alta capacidade de comunicação, os objetos inteligentes presentes na *IoT* compartilham informações com todos os dispositivos da rede. Esta característica torna possível a tomada de decisão inteligente em relação ao contexto em que os dispositivos estão inseridos. Assim, a *IoT* proporcionará a criação de ambientes inteligentes, em que o real, o digital e o virtual convergem entre si, com o propósito de melhorar e agilizar atividades do dia a dia das pessoas em diversas áreas (VERMESAN *et al.*, 2013).

Redes de Sensores

Segundo Rawat *et al.* (2014), as redes de sensores são constituídas por objetos autônomos que atuam de forma distribuída em áreas geograficamente distintas e utilizam diferentes estratégias de comunicação em rede. Geralmente, ela é formada por dispositivos pequenos de baixo custo e gerenciados de forma remota (KULKARNI; FORSTER; VENAYAGAMOORTHY, 2011). Além do mais, estes dispositivos normalmente têm baixo poder de processamento e necessitam trabalhar de forma cooperativa para realizar atividades (AKYILDIZ *et al.*, 2002). Os dispositivos são dispostos em regiões de interesse, cujo objetivo é realizar a detecção e monitoramento de diferentes fenômenos físicos e ambientais. Observa-se então que diferentes tipos de aplicações podem ser desenvolvidas com base em redes de sensores, dentre elas destacam-se: o monitoramento de áreas remotas e inóspitas; análise de fenômenos climáticos de alto risco, como tornados, furacões e enchentes; detecção de incêndios em regiões florestais; o estudo das condições de trânsito em metrópoles; etc (AKYILDIZ *et al.*, 2002) e (NUNES, 2018).

Tratando-se de redes de sensores, os objetos geralmente são posicionados aleatoriamente e não são dispostos de forma fixa, sem que haja uma definição prévia das possíveis rotas de comunicação (AKYILDIZ *et al.*, 2002). Além disso, os dispositivos são organizados em *clusters* e contam com a presença de um nó principal (nó mestre) que recebe os dados de todos os dispositivos e os redireciona para um nó receptor (*sink*) ou para uma estação base (KULKARNI; FORSTER; VENAYAGAMOORTHY, 2011). Por sua vez, os autores Rawat *et al.* (2014) desconsideram a presença de um nó mestre e utilizam a comunicação *multi-host*. Esta estratégia se caracteriza pela troca de dados entre um conjunto de objetos, anteriormente à sua chegada ao destino final (AKYILDIZ *et al.*, 2002). A Figura 1 descreve este comportamento.

Mercado e Expectativas

Em relatório, as Nações Unidas proferem que estamos entrando em uma nova era da ubiquidade, pois é previsto que os seres humanos serão minoria como geradores e receptores de tráfego na *Internet* (BOTTERMAN, 2009). Neste mesmo sentido, o Conselho Nacional de Inteligência Americano (*National Intelligence Council (NIC)*) prevê que objetos de uso cotidiano, como documentos em papel, pacotes de comida, utensílios domésticos, móveis, dentre muitos

Figura 1 – Exemplo de Funcionamento de Rede de Sensores (Adaptado de (AKYILDIZ *et al.*, 2002)).

outros estarão conectados à rede mundial de computadores até o ano de 2025 (COUNCIL, 2008). Estes números demonstram a importância da *IoT* como facilitador na vida das pessoas. Segundo o relatório publicado por Sundmaeker *et al.* (2010), os dispositivos de uso doméstico com acesso à *Internet* representavam 75% do total de aparelhos e terão um crescimento acentuado até a próxima década. Contudo, em 2020 estes aparelhos terão uma parcela bem menor do mercado, em torno de 25%. A Figura 2 mostra a expectativa de crescimento da *IoT* até o ano de 2020. Observa-se que, são apresentadas as estimativas de crescimento da população mundial e a quantidade de dispositivos conectados por pessoa. Portanto, com base nestas expectativas, pode ser verificado que em 2020 o número total previsto de objetos será de 50 bilhões e apresentará uma relação de 6.58 dispositivos por pessoa. Entretanto, Says (2014) destaca que o número total de dispositivos conectados à *Internet* será aproximadamente 25 bilhões em 2020. Além disso, o estudo também afirma que o impacto da *IoT* será em todas as áreas da indústria e sociedade, com receitas previstas em 263 bilhões de dólares.

O relatório econômico Witchalls e Chambers (2013) aponta que cerca de 30% dos líderes empresariais acreditam que a *IoT* permitirá novas oportunidades de negócios. Além do mais, o estudo mostra que 29% dos líderes acreditam que a *IoT* inspirou a criação de novas práticas ou modelos de processos e que 23% deles creem que será provocada uma mudança significativa no modelo estratégico de negócio de suas empresas. A Figura 3 ilustra o restante das estatísticas mencionadas.

Segundo pesquisa realizada no *Google Trends*¹ foi constatado um crescimento acentuado de pesquisas realizadas no *Google*, relacionadas aos termos: *Internet of Things*, *Wireless Sensor Network* e *Cloud of Things* nos últimos 4 anos. Observa-se que, o termo *Internet of Things* foi o mais pesquisado, seguido por *Wireless Sensor Network* e *Cloud of Things* (Figura 4).

Os números apresentados representam o interesse de pesquisa relativo ao ponto mais

¹ <https://trends.google.com.br/trends/>

Figura 2 – Estatísticas de Crescimento da *IoT* (Adaptado de (WITCHALLS; CHAMBERS, 2013)).

Figura 3 – Estatísticas sobre o Impacto do Crescimento da *IoT* (Adaptado de (WITCHALLS; CHAMBERS, 2013)).

alto no gráfico de uma determinada região em um dado período. O valor 100 representa o pico de popularidade de um termo. Por outro lado, o valor 50 significa que o termo teve metade da popularidade. Portanto, uma pontuação de 0 significa que o termo pesquisado teve menos de 1% de popularidade durante determinado período.

Interesse de Pesquisa no Google nos últimos 4 anos

Figura 4 – Total de Pesquisas Realizadas no *Google* em Relação à Ciência (Fonte: *Google*).

Arquitetura em IoT

A *IoT* é considerada uma arquitetura de três camadas. A Figura 5(a) apresenta as camadas de percepção, de rede e aplicação, porém alguns pesquisadores acrescentam as camadas de *middlewares* e de negócio, como mostrado na Figura 5(b) (ZHANG; SUN; CHENG, 2012) (KHAN *et al.*, 2012) (AAZAM; HUH, 2014) (FERSI, 2015). Analisando a composição da arquitetura, a camada de mais baixo nível é a de percepção. Nesta se encontram os dispositivos utilizados para coleta de dados como sensores, leitores de códigos de barras, dispositivos de Identificação por Radiofrequência (*Radio-Frequency IDentification (RFID)*), Sistema de Posicionamento Global (*Global Positioning System (GPS)*), câmeras e outros (AAZAM; HUH, 2014). A camada de rede é responsável por realizar a troca de dados entre a camada de percepção e as demais. A comunicação é feita a partir de redes cabeadas ou sem fio, como por exemplo: o *3G/4G/5G*, *Wifi*, *WiMax*, *Bluetooth* e *Infrared*. Por sua vez, a camada de *middlewares* tem como responsabilidade receber os dados enviados pela camada de rede e oferecer um serviço de armazenamento dos dados coletados (KHAN *et al.*, 2012). A camada de aplicação possui como finalidade a apresentação dos dados recebidos pela camada de rede ou de *middlewares*. Ela necessita de uma maior atenção, pois é importante que os dados apresentados estejam de acordo com as especificações definidas pelo Acordo de Nível de Serviço (*Service Level Agreement (SLA)*) adotado (AAZAM; HUH, 2014). A camada de negócio é responsável pela gestão do sistema, o que inclui suas aplicações e serviços. Ela deve definir os modelos de negócio, os gráficos e o fluxo de execução com base nos dados recebidos pela camada de aplicação. O

sucesso da utilização da *IoT* depende de uma definição efetiva dos modelos de negócio e, por meio da análise dos resultados obtidos por esta camada é que se fundamentam as tomadas de decisões futuras (KHAN *et al.*, 2012) e (NUNES, 2018).

Figura 5 – Estrutura Arquitetural da *IoT* (Adaptado de (NUNES, 2018)).

Desafios Relacionados a utilização da *IoT*

Em *IoT* diversas pesquisas focam no desenvolvimento de tecnologias para superar as limitações de *hardware* de dispositivos móveis, tais como baterias de longa duração, melhorias na velocidade e qualidade de conexão de redes sem fio, segurança, privacidade, interoperabilidade, gerenciamento de dados e escalabilidade. No que diz respeito à esta dissertação, existem poucos trabalhos que abordam os problemas relacionados a interoperabilidade entre aplicações *IoT*. Desta maneira, as soluções presentes devem considerar aspectos da integração de diferentes fontes de dados e serviços *IoT*. A seguir, são descritos os principais desafios enfrentados (NUNES, 2018):

- **Interoperabilidade:** diversas fabricantes de *hardware* utilizam tecnologias, protocolos e padrões diferentes, o que torna um desafio a integração e composição de serviços entre diversos tipos de sensores e tecnologias. A criação de um padrão de comunicação para *IoT* é extremamente importante a fim de permitir a integração entre diferentes dispositivos conectados em rede (KHAN *et al.*, 2012);

- **Gerenciamento dos dados:** os objetos conectados à *Internet* estão em crescente expansão, o que proporciona um aumento significativo na quantidade e tamanho dos dados gerados. Dessa forma, o gerenciamento destas informações se tornou um grande desafio (BURANGE; MISALKAR, 2015);
- **Segurança e Privacidade:** a segurança e privacidade são desafios muito importantes presentes na *IoT*. Além disso, se dividem em dois grandes grupos de estudo: segurança operacional e segurança dos dados. O primeiro se relaciona à capacidade de manter o desempenho e garantir que os dados não serão perdidos, caso parte da rede fique *offline*. Já o segundo foca em garantir a confidencialidade das informações geradas por meio da autenticação, com a finalidade de impedir que usuários não autorizados tenham acesso às informações confidenciais (FERSI, 2015);
- **Escalabilidade:** relativo a capacidade de um sistema, rede, ou processo de suportar diferentes cargas de trabalho de forma eficiente. Este é também um dos grandes desafios da *IoT*, pois o número de dispositivos geradores de tráfego e de dados estão em ampla expansão (GOMES; RIGHI; COSTA, 2014) e;
- **Confiabilidade:** referente à criação de mecanismos capazes de estabelecer o quão confiável é um dispositivo, com base na sua reputação. Geralmente, o grau de confiabilidade apresenta uma alta variação, podendo ser aferida a qualidade das informações geradas por um dispositivo e a sua disponibilidade (DAUBERT; WIESMAIER; KIKIRAS, 2015).

Interoperabilidade

Segundo o Instituto de Engenheiros Eletricistas e Eletrônicos (do inglês, *Institute of Electrical and Electronics Engineers (IEEE)*), a interoperabilidade consiste na capacidade de um sistema de interagir com outros sistemas ou produtos de forma simplificada, sem nenhum esforço adicional (Tolk, 2013). Em *IoT*, ela abrange diferentes camadas, que vão desde a camada de percepção até a camada de aplicação. Portanto, pode ser realizada tanto em nível de sensores, quanto na composição de serviços. Um dos desafios enfrentados pela *IoT* diz respeito à interoperabilidade entre tecnologias e sistemas para a entrega de informações (VERMESAN; FRIESS, 2013). A aquisição de dados de sensores, nesse sentido, relaciona-se à competência de interoperar diferentes fontes de dados, o que torna os desafios de processamento e compressão de informações secundários. A seguir, encontra-se a relação das principais dificuldades encontradas (VERMESAN; FRIESS, 2013; NUNES, 2018):

- **Integração:** refere-se à troca de informações entre diversos tipos de sensores e atuadores disponíveis. As contribuições esperadas nesta área relacionam-se ao desenvolvimento de soluções escaláveis que realizem a integração de diferentes sensores e atuadores de forma a permitir a composição de serviços no nível de usuário;

- **Anotações:** diz respeito ao estabelecimento de ligações e preferências em consultas realizadas em rede de sensores. Esta abordagem visa proporcionar uma maior facilidade na integração e reuso de dados de sensores;
- **Gerenciamento:** relaciona-se à criação e gestão de serviços no contexto de *IoT*. O desenvolvimento deste tipo de aplicação permite a integração entre múltiplas fontes de dados de sensores distribuídos heterogêneos;
- **Busca e Seleção:** abrange a criação de mecanismos para a busca e seleção de sensores de acordo com um conjunto de restrições. No que lhe concerne, espera-se que o motor de buscas forneça interfaces de alto nível e mecanismos para integração de diferentes fluxos de dados;
- **Análise dos Dados:** apresenta-se como um grande desafio, o que torna necessária a criação de ferramentas que permitam a análise e extração dos dados de forma eficiente e eficaz, visto que a *IoT* é uma fonte geradora de dados em larga escala e;
- **Visualização:** a visualização de dados de sensores é uma etapa importante e crítica. Isso porque os dados providos pela *IoT* são heterogêneos e de grande volume.

Protocolos de Comunicação

Diferentes protocolos de comunicação são utilizados em *IoT*, por exemplo, *HTTP*, *CoAP*, *MQTT*, dentre outros. A principal motivação do desenvolvimento de tais protocolos diz respeito às restrições de *hardware* de dispositivos embarcados, o que tem dificultado o uso de protocolos de rede tradicionais (VERMESAN *et al.*, 2013). A seguir são descritos os principais protocolos de comunicação utilizados.

Hypertext Transfer Protocol - HTTP

O protocolo de transferência de hipertexto (do inglês *Hypertext Transfer Protocol* (HTTP)) é um protocolo em nível de aplicação, com a leveza e velocidade necessária para distribuição de sistemas de informação hipermídia, que vem sendo utilizado desde os anos 90 e que possui as seguintes características: permite a comunicação entre clientes e servidores e é flexível, ou seja, qualquer dado pode ser transferido. Além disso, ele é usado amplamente como um protocolo genérico de comunicação entre agentes, *proxies*, *gateways* e diversos outros sistemas *Web* que compõem a *Internet* (FIELDING *et al.*, 1999). O *HTTP* é o protocolo predominante na *Internet*, baseado no padrão requisição/resposta (NAIK, 2017).

Message Queuing Telemetry Transport - MQTT

Em 1999, Andy Stanford-Clark e Arlen Nipper introduziram um novo protocolo chamado *Message Queuing Telemetry Transport* (MQTT) e em 2013 tornou-se um padrão mantido pela

Organization for the Advancement of Structured Information Standards (OASIS). Ele se baseia no padrão *publish/subscribe*, e é considerado o protocolo mais utilizado para comunicação *Machine to Machine (M2M)*. Além disso, utiliza também o modelo de cliente/servidor, logo, todo dispositivo é conectado a um servidor conhecido como *broker*, o que o caracteriza como um protocolo orientado à mensagens. O endereço no qual a mensagem é encaminhada é chamado de tópico e os clientes podem se inscrever em um ou mais tópicos.

No que diz respeito à arquitetura, é possível dividi-la em cliente e *broker*, em que o primeiro pode ser qualquer dispositivo *IoT* que envia ou recebe dados, ou seja, que pode publicar ou assinar um tópico. Já o *broker* é um objeto central, que tem a finalidade de processar a comunicação entre os clientes com base nos tópicos de interesse (AL-FUQAHA *et al.*, 2015). Há três níveis de qualidade do serviço (do inglês *Quality of Service (QoS)*): O nível 0, em que a mensagem é entregue no máximo uma vez e não existe confirmação de que a mensagem foi recebida; O nível 1, cada mensagem é entregue pelo menos uma vez e uma confirmação de mensagem recebida é necessária e; O nível 2, que conta com um mecanismo de *handshake* de quatro vias que é utilizado pra que cada mensagem seja entregue uma vez (YASSEIN *et al.*, 2017).

Constrained Application Protocol - CoAP

O Constrained Application Protocol (CoAP) é um protocolo genérico apresentado pela *Internet Engineering Task Force (IETF)* em 2010, que foi desenvolvido com o objetivo de atender ambientes de redes de sistemas embarcados (NAIK, 2017). Este protocolo visa possibilitar a dispositivos de baixo consumo a utilização de recursos pertencentes ao padrão *REST*, que abstrai todos os objetos da rede em forma de recursos. Nesse modelo, cada objeto possui um Identificador Universal de Recurso (*Universal Resource Identifier (URI)*) e esses recursos podem ser manipulados com operações similares ao *HTTP: GET, PUT, POST e DELETE* (AL-FUQAHA *et al.*, 2015).

Comparativo entre os Protocolos

A Tabela 1 apresenta uma análise comparativa entre os protocolos descritos anteriormente.

Tabela 1 – Principais características dos protocolos HTTP, MQTT e CoAP.

	HTTP	MQTT	CoAP
Arquitetura	Client/Server	Client/Broker	Client/Server ou Client/Broker
Abstração	Request/Response	Publish/Subscribe	Request/Response ou Publish/Subscribe
Tamanho do Cabeçalho	Indefinido	2 Bytes	4 Bytes
Tamanho da Mensagem	Largo e indefinido	Pequeno e indefinido	Pequeno e indefinido
Semântica/métodos	GET, POST, HEAD, PUT, PATCH, OPTIONS, CONNECT e DELETE	Connect, Disconnect, Publish, Subscribe, Unsubscribe e Close	GET, POST, PUT e DELETE
QoS	Limitado pelo TCP	Level 0 - (no máximo uma vez) Level 1 - (pelo menos uma vez) Level 2 - (exatamente uma vez)	Mensagem Confirmada e Não Confirmada
Formato	TCP	TCP	UDP

Gerenciamento de Grandes Volumes de Dados

Ao longo dos últimos anos o volume de informações armazenadas em disco cresceu em larga escala em diversos setores no mundo (CHEN; MAO; LIU, 2014). Isso porque, com o aumento da capacidade de processamento de dispositivos móveis, tornou-se uma tarefa fácil produzir e compartilhar informações. Conforme pesquisa publicada pelo *International Data Corporation (IDC)*, no ano de 2011 o volume total de dados gerados no mundo foi de aproximadamente 1.8 *Zettabyte (ZB)*, um aumento de quase nove vezes em relação ao total de informações armazenadas há cinco anos atrás (GANTZ; REINSEL, 2011).

Atualmente, a aplicação de conceitos relacionados a *Big Data* trouxe o rápido crescimento de várias empresas de tecnologia. Exemplos disso são empresas como a *Google*, que processa centenas de *Petabyte (PB)* de dados diariamente, *Facebook*, que gera dados de registros de seus usuários na casa dos 10 *PB* por mês, a chinesa *Baidu*, que processa dados de dezenas de *PB* por dia e a *Taobao*, empresa subsidiária do grupo *Alibaba*, que gera dezenas de *Terabyte (TB)* diariamente no ramo de comércio eletrônico (CHEN; MAO; LIU, 2014).

Em uma definição amplamente aceita, *Big Data* consiste em um conjunto de dados estruturados e não estruturados que não puderam ser percebidos, adquiridos, gerenciados e processados pelos modelos tradicionais de *hardware* e *software* existentes (CHEN; MAO; LIU, 2014). Além disso, a *IoT* e *Big Data* estão diretamente relacionadas, devido ao crescente número de dispositivos e a heterogeneidade dos dados gerados. Para que uma base de dados seja classificada dessa maneira deve estar de acordo com as seguintes características, conhecidas como 5V's: Volume, Variedade, Velocidade, Veracidade e Valor.

Middlewares

Em rede de sensores, um grande número de eventos podem ser gerados de forma espontânea, o que torna os desafios existentes na computação ubíqua ainda mais complexos (GAVRAS *et al.*, 2007) (PARIDEL *et al.*, 2010). Diferentes *middlewares* são utilizados em *IoT*, em geral, com o objetivo de facilitar o processo de desenvolvimento através da integração de diferentes dispositivos de comunicação heterogênea, proporcionando maior interoperabilidade em aplicações que utilizam rede de sensores (RAZZAQUE *et al.*, 2016).

Atualmente, essa heterogeneidade se aplica ao gerenciamento de rede de sensores em *IoT* também. Alguns exemplos de *middlewares* populares utilizados são: o *GSN*², *OpenIoT*³ e o *Xively*⁴. Uma característica comum a eles é a disponibilidade de uma interface de comunicação simplificada para o gerenciamento e consumo de dados.

É preciso salientar que os *middlewares* utilizam formas distintas de representação de

² <<http://lsir.epfl.ch/research/current/gsn/>>

³ <<http://www.openiot.eu/>>

⁴ <<https://www.xively.com/>>

sensores, o que culmina no uso de diferentes ontologias e Web semânticas para descrevê-los. Nessa perspectiva, ontologias como o *Semantic Sensor Network (SSN)*, *Semantic Sensor Network Ontology (SSNO)* e *Web Ontology Language (OWL)* destinam-se tanto à descrição dos sensores quanto dos dados coletados (O'BYRNE; BRENNAN; O'SULLIVAN, 2010). Além disso, os descritores apresentam formas distintas de representação de dados, assim, a integração entre eles é um ponto chave para garantir uma maior interoperabilidade (CHAQFEH; MOHAMED, 2012).

Os autores Razzaque *et al.* (2016) publicaram uma pesquisa *survey* contendo um levantamento dos *middlewares* mais utilizados em *IoT*. Obteve-se como resultado uma tabela comparativa com os requisitos funcionais e não funcionais dos mesmos, que foram fornecidos de acordo com a literatura. A Tabela 2 expõe as características de alguns, dentre os mais populares.

Tabela 2 – Tabela Comparativa entre *Middleware*s (Adaptado de (RAZZAQUE *et al.*, 2016)).

	Requisitos não Funcionais					
	Escalabilidade	Segurança	Disponibilidade	Confiabilidade	Real-Time	Privacidade
GSN	AL	Integridade	Suportado	Não Informado	NRT	Não Suportado
OpenIoT	AL, NLIoTS	Confidencialidade	Suportado	Não Suportado	-	Não Suportado
Xively	AL, NLIoTS	Confidencialidade	Suportado	Não Suportado	-	Não Suportado
Hydra	AL, NLWSNS	Suportado	Não Informado	-	SRT	Não Suportado
SOCRADES	AL, NLIoTS	Confidencialidade	Não Informado	-	SRT	Não Suportado
IrisNet	NLWSNS	Confidencialidade	Suportado	Não Suportado	SRT	Suportado
TinyDB	NLWSNS	Não Suportado	Não Suportado	Não Suportado	NRT	Não Suportado
Impala	AL, NLWSNS	Integridade e Disponibilidade	Não Suportado	Dados	SRT	Suportado
Ubiware	AL, NLIoTS	Confidencialidade e Disponibilidade	Suportado	Não Informado	SRT	Não Suportado
Legendas	AL - Application Level NRT - Non Real-Time SRT - Soft Real-Time			NLIoTS/NLWSNS - Scalability: Network Level SA (Statically Adaptive) IoT/WSN scale		

Global Sensor Network (GSN)

O *Global Sensor Network (GSN)* é um *middleware* que realiza o processamento de fluxo de dados de sensores, no contexto de sistemas distribuídos. Dessa forma, ele provê uma interface simplificada para a implantação, gerenciamento e manutenção de redes de sensores em *IoT* (SALEHI, 2010). Além disso, oferece diversos recursos, dentre os quais destacam-se as seguintes características: gestão flexível dos recursos, aquisição e processamento de dados integrados, reconfiguração em tempo de execução, camada de aquisição de fluxo de dados flexível, serviços adicionais (Sistema de Autenticação e *API RESTful*), simplicidade, escalabilidade e implementação leve (SALEHI, 2010).

Para aquisição de dados a tecnologia tratada utiliza classes *Wrappers*, a fim de simplificar o gerenciamento de sensores físicos. Uma característica importante delas é que herdam a classe *AbstractWrapper*, primordial para a formatação e encapsulamento dos dados coletados no *GSN*. Além do mais, durante a coleta, as informações são convertidas em objetos *Stream Elements (SE)*, que são sequências temporais de tuplas *Structured Query Language (SQL)* (*Timestamped Tuples*). Como resultado, uma lista de objetos é enviada para um ou mais sensores virtuais, responsáveis pela filtragem, processamento e armazenamento dos dados (SALEHI, 2010).

Open Internet of Things (OpenIoT)

O *OpenIoT* é um *middleware* que visa facilitar a integração de diferentes ecossistemas de serviços em *IoT*. Ele consiste basicamente em três plataformas do lado servidor, sendo elas o *Planet Platform*, *Mashup Platform* e *Store Platform*, e uma do lado do dispositivo, chamado *Device Platform*. Uma característica importante delas é que todas apresentam *Application Programming Interface (API)* abertas baseadas em *RESTful* (KIM; LEE, 2014). Além disso, têm-se que a arquitetura utilizada no *OpenIoT* é dividida em sete módulos, os quais são listados a seguir (SOLDATOS *et al.*, 2015):

- **Módulo *eXtensible Global Sensor Network (X-GSN)***: *middleware* baseado no *GSN* utilizado para coleta, filtragem e combinação de *data streams* por meio de sensores virtuais;
- **Módulo de Armazenamento de Dados na Nuvem**: atua em conjunto com o *Linked Stream Middleware Light (LSM-Light)* como banco de dados em nuvem e permite o armazenamento de fluxos de dados provenientes do *X-GSN*;
- ***Scheduler***: módulo que realiza o processamento de pedidos e implantação sob demanda de serviços, além de garantir o acesso adequado aos recursos (ex. fluxos de dados). Ele efetua a descoberta de sensores e fluxo de dados para a composição de serviços e os gerencia. Além do mais, desenvolve a ativação dos recursos envolvidos;
- ***Service Delivery & Utility Manager (SD & UM)***: Determina a combinação de dados conforme indicado pelos fluxos de trabalhos do *OpenIoT*. Ele atua também como uma técnica para medição de serviço, acompanhada de métricas de utilidades;
- **Definição de Requisição**: componente que permite a especificação *on-the-fly* de solicitações de serviços para a plataforma *OpenIoT*. Compreende um conjunto de serviços para especificar e formular requisições, além de enviá-los ao *Scheduler*. Além disso, este componente apresenta uma *Graphical User Interface (GUI)*;
- **Representação de Requisição**: é responsável pela visualização das saídas obtidas pelos serviços. Este componente seleciona os *mashups* de uma biblioteca apropriada para facilitar a apresentação das informações e;
- **Monitoramento de Configuração**: permite o gerenciamento visual e a configuração de funcionalidades sobre sensores e serviços implantados na plataforma *OpenIoT*.

LogMeIn Xively

O *Xively* é uma solução de *middleware* em *IoT*, mantida pela empresa *LogMeIn*⁵, baseada no conceito de Computação em Nuvem. A empresa que o mantém possui diversas ferramentas

⁵ <<https://secure.logmein.com>>

e recursos para desenvolvedores, usados para o gerenciamento e coleta de dados de sensores (LOGMEIN, 2015). O objetivo dele é simplificar o processo de integração de sensores com a nuvem. Além disso, o *Xively* fornece um conjunto de *API* para facilitar a gestão de recursos de redes de sensores de uso doméstico e comercial. Entretanto, por se tratar de uma solução proprietária, informações mais detalhadas sobre a representação de sensores e as tecnologias internas utilizadas não estão disponíveis para o público geral.

Desafios Relacionados a utilização de Middlewares

Cada *middleware* utilizado em *IoT* apresenta uma especificação diferente para descrição de sensores e de seus recursos, o que caracteriza um desafio para a integração entre diferentes padrões de representação de sensores (CHAQFEH; MOHAMED, 2012). Além disso, ocorrem outros problemas, entre os quais destacam-se (CHAQFEH; MOHAMED, 2012):

- **Interoperabilidade:** representa um grande desafio pois os dispositivos heterogêneos devem colaborar para a comunicação e troca de informações. Isso aumenta o esforço de pesquisa para projetar um *middleware* que possa cobrir um grande número de diferentes tipos de dispositivos, considerando até mesmo aqueles que possam ser desenvolvidos posteriormente;
- **Escalabilidade:** configura-se como um desafio devido à necessidade de gerenciar os milhões de dispositivos suportados pela *IoT*. Neste contexto, um fato importante é que o *middleware* seja capaz de administrar ambientes de pequena e larga escala;
- **Abstração de Provisionamento:** um *middleware* ideal para um ambiente inteligente como o *IoT* deve fornecer abstrações em vários níveis, tais como dispositivos de *hardware* de entrada e saída heterogêneos, interfaces de *hardware* e *software*, fluxos de dados e o processo de desenvolvimento e;
- **Multiplicidade:** dois grandes desafios de multiplicidade devem ser levados em consideração no projeto de *middleware* em *IoT*. Em primeiro lugar, os dispositivos na *IoT* precisam, muitas vezes, se comunicar com outros dispositivos simultaneamente. Em segundo lugar, um dispositivo que está participando de um ambiente *IoT* precisa selecionar os serviços mais adequados a partir de um enorme conjunto, pois muitas vezes dependem de serviços que estão disponíveis em outros dispositivos próximos.

2.3 Computação em Nuvem e Internet das Coisas

A Computação em Nuvem é um modelo de computação em que recursos computacionais são compartilhados na *Internet* em forma de serviços (ARMBRUST *et al.*, 2010). Entre esses recursos estão os servidores, redes, armazenamento e processamento. Deve-se salientar que boa

parte destes são virtualizados, o que torna possível a automação da criação de ambientes e a escalabilidade da infraestrutura de forma simplificada. Segundo [Zhang, Yan e Chen \(2012\)](#), a virtualização permitiu a configuração de recursos de forma facilitada, permitindo que os usuários façam alterações de acordo com as suas necessidades. Ainda, conforme explícito por [Keshavarzi, Haghighat e Bohlouli \(2013\)](#), diversas empresas e universidades têm utilizado este modelo de computação com o objetivo de suportar as demandas de infraestrutura crescentes através de uma mesclagem entre o modelo de nuvens privadas e públicas.

Essa tecnologia apresenta quatro principais categorias: Nuvem Pública, Privada, Híbrida e Comunitária. Plataformas de Computação em Nuvem como *Amazon Web Services (AWS)*, *Google Cloud Platform (GCP)* e *Microsoft Azure* são exemplos de grandes fornecedores de nuvem pública. Tais provedores fornecem recursos de forma compartilhada e sua infraestrutura está disponível em diferentes países. Entretanto, no modelo de Nuvem Privada os recursos não são compartilhados e geralmente a infraestrutura é gerida pela própria instituição proprietária ([AHMED et al., 2012](#)) e ([KESHAVARZI; HAGHIGHAT; BOHLOULI, 2013](#)). Já a Nuvem Híbrida é um modelo que une as características da Nuvem Pública e Privada, pois possibilita que instituições possam alocar parte de seus recursos na Nuvem Pública (dados públicos) e que outra parte seja implantada em uma Nuvem Privada (dados confidenciais). Por fim, o modelo de Nuvem Comunitária tem como característica o compartilhamento de recursos entre usuários de uma mesma comunidade ([ATLAM et al., 2017](#)).

Em Computação em Nuvem são utilizadas três classificações de serviços, são elas o modelo de *Software como Serviço (Software as a Service – SaaS (SaaS))*, *Plataforma como Serviço (Platform as a Service (PaaS))* e *Infraestrutura como Serviço (Infrastructure as a Service (IaaS))*. O modelo *SaaS* tem como característica a entrega de *software* por meio serviços online, por exemplo, *Google Docs*, *Netflix*, *Dropbox*, dentre outros. O modelo de *PaaS*, por sua vez, oferece ambientes para que os desenvolvedores implantem *software* sem se preocupar com o processo de provisionamento dos recursos necessários (e.g., *Google App Engine* e *AWS Elastic Beanstalk*). Por fim, o modelo *IaaS* oferece o provisionamento de infraestrutura de forma simplificada, alguns exemplos famosos são o *Google Compute Engine* e *AWS EC2* ([PATIDAR; RANE; JAIN, 2012](#)) e ([KESHAVARZI; HAGHIGHAT; BOHLOULI, 2013](#)).

A *IoT* e a Computação em Nuvem tem se desenvolvido rapidamente nos últimos anos, sendo empregadas inclusive em conjunto, a fim de tornar possível a criação de aplicações em larga escala. A *IoT* tem como característica a presença de dispositivos amplamente distribuídos que apresentam recursos e desempenho limitados ([BOTTA et al., 2016](#)), enquanto a Computação em Nuvem apresenta uma alta disponibilidade de recursos e auxilia a integração com dados de fontes heterogêneas ([ZHOU et al., 2013](#)). A combinação entre os dois conceitos tem sido chamado de Nuvem de Coisas (do inglês, *Cloud Of Things (CoT)*) que tem como marco a possibilidade de desenvolvimento de novos serviços que auxiliam na criação de aplicações *IoT* escaláveis ([BABU; LAKSHMI; RAO, 2015](#)) e ([BOTTA et al., 2016](#)). Alguns exemplos de novos

serviços que surgiram após a união das duas áreas são: *Sensing as a Service (S2aaS)*: possibilita o acesso a uma rede de sensores; *Sensor as a Service (SenaaS)*: permite o gerenciamento de sensores remotamente; e o *Data as a Service (DaaS)*: possibilita o acesso a dados heterogêneos de forma simplificada.

2.4 Considerações Finais

Este capítulo apresentou o Referencial Teórico desta dissertação. Nas seções apresentadas foram descritos em detalhes os conceitos que caracterizam à Internet das Coisas, Computação em Nuvem e Rede de Sensores. Foram expostos os fundamentos básicos relacionados a esta dissertação, apontando os principais conceitos associados à grande área desta pesquisa, a *Internet das Coisas*. Por fim, foram discutidas também as diferentes técnicas e estratégias, assim como os desafios referentes a cada um dos temas abordados.

TRABALHOS RELACIONADOS

3.1 Considerações Iniciais

Este capítulo apresenta os Trabalhos Relacionados ao tema desta dissertação. Primeiramente, apresenta-se a metodologia de pesquisa adotada para a construção da *string* de busca, usada para a seleção dos estudos primários. Por fim, apresenta-se a análise crítica dos trabalhos e uma tabela comparativa com as principais características de cada um.

3.2 Revisão da Literatura

Para a elaboração desta pesquisa realizou-se uma revisão da literatura com base nos principais estudos primários. O levantamento bibliográfico se orientou através das seguintes bases de dados: *Scopus*¹, *IEEE Xplore*² e *ACM Digital Library*³. É preciso salientar que a *string* de busca foi estabelecida por meio de uma pesquisa exploratória, sob a qual definiu-se um conjunto de palavras chaves que relacionadas, trouxeram o seguinte resultado:

((“Interoperability”) AND (“Internet of Things” OR “Cloud of Things”) AND (“Data” AND (“Conversion” OR “Integration” OR “Management”))))

A partir da leitura dos trabalhos obtidos, selecionou-se os estudos primários que possuíam maior relevância para o tema desta dissertação. Para esse fim, foram definidos critérios de seleção de forma a se obter artigos diretamente relacionados ao foco de pesquisa deste projeto. Além disso, a extração de dados foi realizada com auxílio de um formulário do *Google Forms*⁴.

---

¹ <<https://www.scopus.com/>>

² <<http://ieeexplore.ieee.org/>>

³ <<http://dl.acm.org/>>

⁴ <<https://docs.google.com/forms>>

Formulário de Extração de Dados

A Tabela 3 representa o formulário de extração de dados utilizado. Observa-se que foram definidos itens que mapeiam aspectos diversos, dentre eles o local de publicação, tipo de publicação (conferência ou *journal*), tipo de aplicação (protótipo ou simulação), objetivos, contribuições, limitações, semelhanças com relação ao problema de pesquisa desta dissertação, limitações, dentre outras informações.

Tabela 3 – Formulário de Extração de Dados (* campos obrigatórios).

Pergunta	Opções de Respostas	Descrição
Título *	-	Título do trabalho analisado
Ano *	-	Ano de Publicação do Artigo
Conferencia/Journal *	Conferência Journal	Se o artigo foi publicado em Conferência ou <i>Journal</i>
Protótipo/Simulação *	Protótipo Simulação	Tipo de implementação utilizada
Tipo de Arquitetura *	-	Qual arquitetura computacional foi utilizada?
Camada de Atuação (IoT) *	Percepção, Rede, Middleware, Aplicação ou Negócio	Qual camada de atuação em <i>IoT</i> se encontra o trabalho analisado?
Foi aplicado alguma metodologia para interoperabilidade? *	Sim Não	Se utilizou alguma técnica para interoperar aplicações/serviços <i>IoT</i>
Qual o objetivo do artigo? *	-	Breve descrição do objetivo do artigo
Breve descrição da metodologia utilizada? *	-	-
Quais são as limitações do artigo? *	-	Breve descrição do objetivo do artigo
Quais são as semelhanças do artigo com o tema de pesquisa desta dissertação? *	-	Breve descrição das semelhanças encontradas
Outras informações relevantes *	-	Resultados, Conclusões, informações importantes

Análise Crítica dos Trabalhos Relacionados

Os autores [Khan, Silva e Han \(2017\)](#) apresentaram uma arquitetura chamada *Web of Things - Emerging Sensor Network (WoT-ESN)* cujo objetivo consiste em garantir uma melhor interoperabilidade entre diferentes tipos de sensores de uma *Smart Home*. Para tanto, baseou-se no padrão de comunicação *Representational State Transfer (REST)*. Nesse sentido, este trabalho apresenta uma descrição detalhada sobre a arquitetura, além dos resultados obtidos pela aplicação de uma avaliação de desempenho. De maneira simplificada, o procedimento consistiu em simular dois tipos diferentes de experimentos: a avaliação do consumo de energia e do tempo de resposta à requisição. Os resultados obtidos apontaram que o *WoT-ESN* apresenta baixa latência e baixo consumo energético.

Por sua vez, [Perera e Vasilakos \(2016\)](#) propuseram uma abordagem orientada a conhecimento, intitulado *Context Aware Sensor Configuration Model (CASCOW)*. A proposta da pesquisa consistiu em simplificar o processo de configuração das plataformas de *middlewares* em *IoT*, de forma a facilitar a recuperação de dados de sensores. Dessa maneira, foi apresentada uma

descrição semântica de recursos de sensores com base na ontologia *SSN*. É preciso evidenciar que a metodologia de avaliação consistiu na aplicação de métodos qualitativos e quantitativos. A partir disso, o *CASCOM* foi comparado com o modelo de configuração do *middleware GSN*, sendo que como medida quantitativa avaliou-se a capacidade de armazenamento, o tempo de carregamento e processamento de consultas e, de forma qualitativa, a usabilidade do sistema. Nos resultados obtidos o *CASCOM* apresentou boa usabilidade e desempenho no armazenamento de descrições semânticas e baixa latência em consultas.

Datta e Bonnet (2015), descreveram a *IoT* e *M2M* como áreas facilitadoras de iniciativas de *Smart Cities*, apresentando uma arquitetura que implementa mecanismos para a integração de variados tipos de sensores. Para isso, foi desenvolvida uma *API RESTful* sob o protocolo *HTTP* e utilizado o *Machine-to-Machine Measurement Framework (M3 Framework)* e o *Sensor Markup Language (SenML)* para a descrição semântica dos sensores. Uma característica importante é que a ferramenta apresentada atuou na camada de *middleware* e interoperava diferentes tipos de sensores e protocolos utilizados na camada de percepção. Os autores concluíram o trabalho apresentando detalhadamente o funcionamento da solução proposta, no contexto de cidades inteligentes.

Alam e Noll (2010), no que lhes concerne, apresentaram um *proxy framework* baseado em uma estrutura semântica de tecnologias relacionadas à Computação em Nuvem e *IoT*. O trabalho expõe objetos inteligentes da nuvem e da *IoT* com base na orientação e virtualização de serviços. Uma característica peculiar desse trabalho é o uso da abordagem *sensor-as-a-service*, de forma a prover não somente a virtualização no domínio de *IoT*, mas também a capacidade de expor sensores na forma de serviços. O objetivo consistiu em apresentar uma abordagem para a melhoria da interoperabilidade entre diferentes tipos de objetos inteligentes. Dessa maneira, foi proposto um modelo para descrição de serviços baseado na ontologia *OWL* para a associação de objetos com seus *endpoints*. Como medida avaliativa, por conseguinte, apresentou-se dois estudos de casos, o *Green School Motorcycle (GSMC)* e o monitoramento contínuo de trens e infraestrutura ferroviária.

Os autores Bovet e Hennebert (2014) revelaram uma ontologia para a descrição semântica de objetos no contexto de *IoT*. A contribuição do artigo foi a apresentação de uma metodologia para melhorar a interoperabilidade entre diferentes sensores. Em resposta a este desafio, foi sugerida uma abordagem para a representação e descrição de dispositivos, incluindo a invocação de dados e serviços. Um ponto importante a ser considerado é que esse estudo se baseia no paradigma *Web of Things (WoT)*, caracterizado pela representação de serviços de sensores como um recurso da *Web*. A fim de demonstrar a viabilidade da arquitetura proposta, foi desenvolvido um protótipo baseado na linguagem de programação *Java*, em execução em computadores *Raspberry Pi*. A implementação foi realizada com base no protocolo de comunicação *CoAP* e, como resultado, apresentou-se o funcionamento da ontologia *SSN* para a descrição semântica de sensores.

Datta, Bonnet e Nikaein (2014) apresentaram uma arquitetura para *IoT* que permite a integração, em tempo real, entre clientes móveis e objetos inteligentes (ex. sensores e atuadores) por meio de um *gateway*. Ele atua como parte central da arquitetura *IoT*, que é composta por três camadas: (i) detecção de camadas contendo os dispositivos *M2M* e os pontos finais, (ii) camada *API* do *gateway* e (iii) camada de aplicação. Para validação da mesma, foram apresentados protótipos submetidos a testes com base em metadados de sensores em tempo real, provenientes de dispositivos *M2M*. Além disso, o sistema apresentou escalabilidade para lidar com grande quantidade de tráfego, pois ele se baseia no padrão *RESTful* e o *SenML*. Dessa forma, a arquitetura pode ser usada em vários cenários, incluindo o monitoramento ambiental e de saúde.

Os autores Desai, Sheth e Anantharam (2015), por sua vez, propuseram um *gateway* e uma arquitetura semântica em *IoT* chamada *Semantic Gateway as Service (SGS)*. O objetivo deste estudo foi fornecer uma maior interoperabilidade entre sistemas que utilizam os padrões estabelecidos para comunicação e representação de dados de sensores. Desse modo, o *SGS* permitiu a tradução entre diferentes protocolos de mensagens como o *eXtensible Messaging and Presence Protocol (XMPP)*, *CoAP* e *MQTT* através de uma arquitetura de *proxy* multi-protocolo, além de que empregou-se, como descritor semântico, a ontologia *SSN*, com o objetivo de proporcionar a integração entre diferentes fontes de dados. Nesse sentido, foi apresentada uma solução para o problema referente à integração entre diferentes sensores no nível de aplicação, contornado os desafios associados à variabilidade dos protocolos de rede. Tem-se ainda que a interface de serviço do *gateway* proposto se conecta aos serviços em nuvem ou outras instâncias do *SGS* via *RESTful API* ou *publish subscribe*. Como resposta à pesquisa, obteve-se a descrição detalhada do funcionamento da arquitetura proposta.

Glombitza *et al.* (2010) abordou os desafios relacionados à utilização de Arquiteturas Orientadas a Serviços (do inglês, *Service Oriented Architecture (SOA)*) no contexto de rede de sensores, estendendo-a a sensores sem fio de forma compatível com o padrão, sem exceder os recursos restritos dos mesmos. Para fins de avaliação, dois exemplares de serviços da *Web* foram utilizados, sendo o primeiro deles responsável por oferecer uma operação para verificação das condições de armazenamento durante um intervalo de tempo. O segundo serviço, no que lhe concerne, é mais complexo e precisou de um *Identity Document (Id)* como entrada para retornar o tempo de carregamento, o *Id* do produtor, o número de unidades armazenadas e uma *URL* para uma página da *Web* com mais informações. No experimento, foram invocados dois serviços da *Web* a partir do qual foi medido o tempo de execução das etapas de conversões individuais. Esse procedimento repetiu-se 10.000 vezes obtendo-se a média e o desvio padrão para cada invocação. Como resposta, apontou-se ganhos significativos de desempenho.

Guinard *et al.* (2010), apresentaram uma proposta de processo e uma arquitetura *SOA* em *IoT* para a descoberta, consulta, seleção e provisionamento sob demanda de *Web Services*. Os objetivos desta abordagem foram as seguintes: (i) oferecer um serviço mínimo de despesas gerais, (ii) esforço de registro mínimo, (iii) melhor suporte para pesquisa dinâmica e contextual e (iv)

suporte para o provisionamento sob demanda. Para a realização desta pesquisa, desenvolveu-se um protótipo que foi integrado à arquitetura *Service Oriented Architecture in der Automatisierungstechnik (SOCRADES)*. Essa implementação foi desenvolvida em *Java* e implantada em *Java Enterprise Application Server (JavaEE)* em dois locais distintos. A avaliação do protótipo, por sua vez, foi dividida em três etapas segundo as subpartidas do processo de descoberta do serviço do mundo real (tipos de consultas, pesquisa de candidatos e provisionamento). Os resultados mostraram a flexibilidade que a abordagem do artigo apresentou para a integração entre *SOA* e *IoT*.

Os autores [Guinard, Trifa e Wilde \(2010\)](#) descreveram uma arquitetura e práticas recomendadas para *WoT* com base no padrão de comunicação *RESTful*. Para isso, foi apresentada uma discussão sobre protótipos projetados de acordo com os princípios de *WoT* que visam conectar nós de sensores ambientais em um sistema de monitoramento de energia para a *Web*. Isso porque alguns dispositivos não podem se conectar à Internet ou implementar completamente o padrão *REST* e, em resposta a essa provocação, o artigo propôs a utilização de *Smart Gateways*, que são servidores da *Web* que abstraem comunicações e serviços de dispositivos não habilitados para a *Web* por trás de uma *API RESTful*. Apresentou-se então, uma avaliação de desempenho da arquitetura proposta e, como resposta à pesquisa, o artigo sugere que a verbosidade do protocolo *HTTP* não impede a implantação de aplicativos altamente eficientes. Deve-se salientar que há diversos protocolos otimizados que minimizam a conexão de rede e latência para rede de sensores sem fio, no entanto, quando os dispositivos estão conectados a uma fonte de energia e quando a latência de segundos pode ser tolerada, as vantagens de *HTTP* compensam a perda de desempenho e latência.

[Gupta, Poursohi e Udupi \(2010\)](#) apontaram uma infraestrutura baseada na *Web*, chamada *Sensor.Network*. O objetivo dela é armazenar, compartilhar, pesquisar, visualizar e analisar dados heterogêneos de sensores, a fim de facilitar a integração entre dispositivos e usuários finais por meio de uma *RESTful API*. Ela consiste em uma plataforma de troca de dados com uma *API* aberta, o que elimina o baixo acoplamento e permite a composição de novos serviços que incorporam sensores heterogêneos e atuadores de diferentes fabricantes. O produto dessa pesquisa foi a demonstração do funcionamento do *Sensor.Network*, apresentando algumas telas do sistema e exemplos de consultas que poderiam ser realizadas.

O trabalho dos autores [Hsu, Lin e Chen \(2014\)](#) consistiu no estudo da seleção de sensores sensíveis ao contexto (*context-aware sensor selection*). O artigo propôs uma arquitetura para a escolha e detecção de um grande conjunto destes objetos de forma eficiente, tal que diferentes parâmetros de seleção poderiam ser utilizados, o que inclui as propriedades de contexto deles (precisão, faixa de detecção, energia residual, etc) e os requisitos de usuário. Além disso, a arquitetura em questão tentou reduzir o tráfego excessivo da rede causado pela mudança frequente dos estados dos sensores. Desta forma, para avaliação do trabalho, desenvolveu-se um protótipo de seleção de sensores semânticos que possibilita a integração com diferentes tipos de aplicações.

A linguagem de programação utilizada foi o *Java* e a arquitetura foi projetada na forma de um serviço da *Web* através do *Apache Axis2* e do *Apache Tomcat*. Além do mais, empregou-se a ontologia *SSN* para gerenciar as informações dos objetos inteligentes e o *Apache Jena* para acessá-las, visando a configuração automática e gerenciamento de contexto. A fim de realizar a avaliação de todo o processo, recorreu-se à análise de tráfego da rede, seguida por uma simulação do comportamento dos sensores com o auxílio do simulador *Objective Modular Network Testbed in C++ (OMNet++)*. É preciso ressaltar que abordagem utilizada apresentou uma eficiência considerável.

Foi evidenciado por Kothari *et al.* (2014) um projeto, chamado *Data Quality-Aware Sensor Cloud (DQS-Cloud)*, que consiste em uma infraestrutura para a provisão de acesso aos serviços de coleta de dados de sensores baseados na nuvem. Ele é caracterizado por três principais recursos: A qualidade dos dados, generalizada em toda a infraestrutura, desde a descoberta até o tratamento de falhas; Técnicas baseadas em computação autônoma, aplicadas para lidar com a dinâmica de qualidade dos dados e falhas decorrentes do gerenciamento de sensores e; Um mecanismo para o gerenciamento de fluxo de sensores que otimiza o desempenho do sistema colocando, de maneira dinâmica, operadores de gerenciamento de fluxo. O padrão implementado foi o *Publish Subscribe*, responsável pela comunicação e recuperação de dados. Além disso, analisou-se na pesquisa o consumo da *Central Processing Unit (CPU)*, largura de banda e otimização de *Data Stream* com o intuito de avaliar o desempenho da metodologia adotada. De maneira sucinta, alcançou-se uma resposta positiva para a entrega de *feeds* de dados de sensores aos clientes, provando-se o *DQS-Cloud* eficaz na minimização do consumo de recursos computacionais, como largura de banda e *CPU*.

Lin *et al.* (2016), por sua vez, propuseram soluções para descoberta de conhecimento automatizado e anotação semântica para serviços de rede e *Web*, assim como uma metodologia para transformar dados de sensores em conhecimento e inteligência acionável. Objetivou-se com a pesquisa o desenvolvimento de uma arquitetura *SOA* para o gerenciamento de redes de sensores. Para a descrição semântica dos mesmos utilizou-se a ontologia *Semantic Markup for Web Services (OWL-S)* (ontologia baseada na *OWL*), enquanto que para a descrição dos serviços disponibilizados pela arquitetura foram usados arquivos *Web Services Description Language (WSDL)*. Além disso, a metodologia *Knowledge Discovery in Services (KDS)*, tecnologia para adquirir conhecimento de serviços, foi aplicada para extração de conhecimento semântico, tal que após coletado este era traduzido para um formato de informação padrão. Como resposta a essa pesquisa conclui-se que a abordagem apresenta bons resultados para a anotação semântica de sensores no contexto de arquiteturas *SOA*.

Os autores Nagib e Hamza (2016) apresentaram um *framework* para integração de dados de sensores, chamado *Semantic InteGration of HeTerogeneous SEnsor Data (SIGHTED)*, cuja base é formada pelos princípios de dados da *Web Semântica* e *Linked Data*. Ele comporta uma estrutura em camadas para a integração heterogênea entre múltiplas fontes de dados de

sensores. Também foi implementado, como uma instância do *SIGHTED*, uma plataforma de demonstração, o *DotThings*, cujo objetivo é a publicação e consumo de dados de sensores semanticamente anotados. A pesquisa revela também uma avaliação de desempenho com base na definição de diferentes consultas *SPARQL Query Language for RDF (SPARQL)* realizadas na plataforma. É necessário salientar que as consultas foram executadas com tamanhos diferentes de armazenamento semântico *RDF* (50kb, 250kb e 1mb). Fez-se então uma avaliação do tempo médio de resposta para cada uma das consultas executadas e o *framework* demonstrou baixa latência nos testes realizados.

Orestis, Dimitrios e Ioannis (2012) discutiram a integração de *Wireless Sensor Networks (WSN)* e objetos inteligentes com a *Web*. A proposta foi um protótipo chamado *Uberdust* que realizava a integração entre diferentes fontes de dados de sensores. Essa tecnologia possibilitava o armazenamento, compartilhamento e descoberta de dados de objetos inteligentes, em tempo real, através da *Web*. Deve-se acrescentar que ele foi projetado como um sistema *M2M*, utilizando a *Web Semântica* para facilitar o desenvolvimento de aplicações em *IoT*. Uma característica importante é que ele não se limita à coleta de dados de sensores, mas possibilita a comunicação bidirecional entre aplicativos *IoT* e objetos inteligentes. Nessa perspectiva, a comunicação entre as diferentes camadas do protótipo realizava-se por meio de uma *API RESTfull* e *Websocket*, com o uso do padrão *REST (HTTP e CoAP)* para a comunicação. Como resultado do artigo, apresentou-se o funcionamento do protótipo.

Perera *et al.* (2012) expuseram uma arquitetura intitulada *Context Awareness for Internet of Things (CA4IoT)* que visava automatizar a seleção de sensores de acordo com os problemas e/ou tarefas desejadas. O foco desta é a automação do processo de configuração dos mecanismos de filtragem, fusão e raciocínio de dados. Dessa forma, o objetivo do trabalho consiste em permitir que usuários apresentem os seus problemas, para que o *CA4IoT* produza informações abrangentes e significativas dos dados gerados pelos objetos inteligentes. Além disso, foi utilizado o padrão *XML* e *Sensor Model Language (SensorML)* para a descrição semântica. Como medida avaliativa, apresentou-se um estudo de caso sobre o uso da arquitetura proposta na agricultura e, com base em um conjunto de restrições definidas, criou-se um processo de execução, em que ela foi responsável pela gerência e coleta de dados dos sensores utilizados. Nessa acepção, o artigo apresentou vários aspectos de *IoT* além de uma visão geral sobre a *CA4IoT*. É preciso observar que a arquitetura apresentada aprimorava a capacidade de conhecimento de contexto de soluções de *middleware* em *IoT* e permitia a construção de uma plataforma para detecção de serviço. Além disso, ela foi projetada com links para projetos, como o *OpenIoT*, *Phenonet* e *SenseMA*. Por fim, demonstrou-se o funcionamento e integração do *CA4IoT* com diferentes *middlewares*.

Perera *et al.* (2014), por sua vez, propuseram uma ferramenta chamada *SmartLink*, desenvolvida para realizar a descoberta e configuração de sensores. A ferramenta proposta estabelece uma comunicação direta entre objetos inteligentes e um *middleware* baseado em Computação em Nuvem. Tratando-se disso, o desafio da heterogeneidade foi abordado por meio

de uma arquitetura extensível com a utilização de *plugins*. Para a avaliação, desenvolveu-se um protótipo para plataforma móvel *Android* e obteve-se, por fim, uma análise a respeito da abordagem adotada para descoberta e configuração de 52 tipos diferentes de sensores.

Os autores [Petrolo et al. \(2014\)](#) trouxeram uma metodologia de integração entre o *Future Internet of the Things IoT-LAB (FIT IoT-LAB)* e o *middleware OpenIoT*. A lógica proposta representa uma maneira de redução da fragmentação da *IoT* e, além disso, permite que os usuários desenvolvam aplicativos para *Smart Cities* para interação de sensores em diferentes camadas. A fim de alcançar essa proposta, apresentou-se uma definição de *virtual sensor* que torna o *FIT IoT-LAB* compatível também com o *middleware GSN*. Para a descrição dos sensores, utilizou-se a *SSN*, sendo esta, a mesma utilizada pelo *OpenIoT*. Como resposta, demonstrou-se o funcionamento do *FIT IoT-LAB* em conjunto com os *middlewares OpenIoT* e *GSN*.

O trabalho apresentado pelos autores [Sandu, Costache e Balan \(2015\)](#) apontou mecanismos para facilitar a interoperabilidade entre diferentes protocolos de comunicação e formatos de representação de dados em redes de sensores. A sua abordagem objetiva a melhoria dessa integração de sub-redes em um laboratório de instrumentação universitária. A solução proposta utiliza o *JSON for Linking Data (JSON-LD)* para representação semântica dos sensores e a comunicação se dá através de uma interface *MQTT* e *RESTful* (por meio do protocolo *CoAP*). Como instrumento de trabalho, utilizou-se a linguagem *JavaScript* com base no *framework NodeJS* e o banco de dados *MongoDB* para a persistência dos dados semânticos. Foi possível concluir, com a análise dos mecanismos, que a abordagem adotada auxiliou na integração entre redes heterogêneas de sensores, cumprindo-se então o objetivo da pesquisa.

A Tabela 4 traz uma comparação entre os principais trabalhos relacionados a este projeto. Observa-se que, boa parte deles desenvolvem-se através de uma estratégia de comunicação baseada no padrão *RESTful API* e *SOA*. Deve-se destacar que alguns trabalhos apresentam uma arquitetura voltada para a Computação em Nuvem enquanto os demais não especificam a abordagem utilizada, além do que, a maioria atua na camada de aplicação realizando a integração entre diferentes fontes de dados de redes de sensores. É constatado a utilização de diversos descritores semânticos, entre os quais se destaca o *SSN*, seguido pelo *XML*, *OWL*, *SSNO*, *SensorML*, *SenML*, *CSV*, *JSON* e *JSON-LD*. Além disso, têm-se que boa parte dos trabalhos relacionados objetivam a padronização da comunicação entre diferentes tipos de objetos inteligentes, mas em contraponto, poucos entre eles aplicaram uma metodologia de avaliação de desempenho em suas abordagens. Outra lacuna encontrada nos trabalhos é não utilização, em sua maioria, de técnicas de processamento paralelo/distribuído para o processamento dos dados coletados pelos sensores, pois devido as expectativas de crescimento da quantidade de sensores no mundo a aplicação destas técnicas de processamento otimizadas se tornam indispensáveis para escalar este tipo de aplicação. Portanto, em contraste com as pesquisas apresentadas, esta dissertação contempla as características citadas na Tabela 4, como avaliação de desempenho dos resultados obtidos, a definição de um mecanismo para integração de fontes de dados heterogêneas

e a aplicação de diferentes técnicas para o processamento e recuperação de dados de sensores.

Tabela 4 – Relação das Principais Características dos Trabalhos Relacionados.

Artigo	Arquitetura/Comunicação	Camada de Atuação (IoT)	Interoperabilidade	Avaliação de Desempenho
(KHAN; SILVA; HAN, 2017)	RESTful API	Aplicação	Sim	Sim
(DATTA; BONNET; NIKAEIN, 2014)	RESTful API	Aplicação	Não	Não
(DESAI; SHETH; ANANTHARAM, 2015)	RESTful API	Middleware	Sim	Não
(GUINARD <i>et al.</i> , 2010)	RESTful API	Aplicação	Sim	Sim
(GUPTA; POURSOHI; UDUPI, 2010)	RESTful API	Aplicação	Sim	Não
(NAGIB; HAMZA, 2016)	RESTful API	Aplicação	Sim	Sim
(ORESTIS; DIMITRIOS; IOANNIS, 2012)	RESTful API (HTTP e CoAP)	Middleware	Sim	Não
(PERERA <i>et al.</i> , 2014)	Cloud Computing	Aplicação	Sim	Sim
(PETROLO <i>et al.</i> , 2014)	Cloud Computing	Aplicação	Sim	Não
(SANDU; COSTACHE; BALAN, 2015)	MQTT e RESTful API (CoAP)	Aplicação	Sim	Não

3.3 Considerações Finais

Este capítulo apresentou os Trabalhos Relacionados, descrevendo uma análise crítica sobre os estudos primários relacionados a este projeto de mestrado. A aplicação de uma metodologia para a definição da *string* de busca e dos critérios de seleção de trabalhos, foram fundamentais para o levantamento e análise crítica dos trabalhos relacionados a esta pesquisa. Após a seleção dos trabalhos, foi extraído as principais características encontradas nos estudos selecionados. Além disso, para auxiliar o levantamento destas características, foi utilizado um formulário para extração de dados. Por fim, foi apresentado uma análise crítica e uma tabela comparativa com as principais características dos trabalhos relacionados.

DESENVOLVIMENTO DO TRABALHO

4.1 Considerações Iniciais

Este capítulo descreve os mecanismos desenvolvidos nesta dissertação, dessa forma, possui tópicos relativos ao desenvolvimento do *Internet of Things Data as a Service Module (IoTDSM)* e o *Internet Multi-Protocol Message Broker (IoTM²B)*. Nas próximas seções, serão expostas as metodologias adotadas e as contribuições obtidas como resultado desta pesquisa.

4.2 Projeto de Pesquisa

Esta dissertação é parte integrante do projeto de pesquisa chamado “*ViSIoT – Visual Search for Internet of Things*”, o qual objetivou a construção de uma arquitetura para a provisão de rede de sensores como serviço. O Módulo Repositório, proposto pela *ViSIoT*, foi o objeto de implementação desta dissertação, que tem como propósito facilitar a integração entre diferentes fontes de dados heterogêneas provenientes de *IoT*. Assim, a principal contribuição deste trabalho é facilitar a interoperabilidade com múltiplas fontes de dados de sensores.

A arquitetura *ViSIoT* possibilita o acesso a um conjunto de sensores disponíveis em forma de serviço e a comunicação com sensores conectados em rede. A finalidade desta é simplificar o processo de criação e configuração de ambientes baseados em *IoT*, através da definição de mecanismos para a implantação, operação e recuperação de informações oriundas de rede de sensores. A *ViSIoT* foi dividida em sete módulos (Figura 6) (NUNES, 2018):

- **Interface:** responsável por fornecer uma interface gráfica para a busca e seleção de sensores, além de configurar o ambiente que será utilizado para processar tais informações;
- **Repositório:** tem como objetivo fornecer informações sobre o objetos pertencentes a um ou mais ambientes inteligentes. Este módulo é responsável por armazenar as descrições

dos sensores pertencentes aos diferentes ambientes disponíveis em *IoT*;

- **Seletor:** gerencia os algoritmos utilizados para realizar a busca e a seleção dos sensores. Estes devem ser capazes de tomar decisão com base em múltiplos objetivos estabelecidos;
- **Monitor:** recupera o estado e as informações dos objetos que foram selecionados em tempo real, com o intuito de impedir as violações do contrato de nível de serviço;
- **Ambiente Alvo:** contém os dispositivos que processam as informações coletadas dos sensores. Uma vez que o ambiente alvo é configurado pela arquitetura, os dispositivos que o compõem requisitarão as informações diretamente aos sensores escolhidos;
- **Conversor:** converte a forma de acesso aos sensores registrados no repositório para um formato específico que deverá ser interpretado pelos dispositivos existentes; e
- **Core:** orquestra todos os módulos da arquitetura. Este módulo gerencia todos os arquivos necessários para configurar o ambiente desejado.

Figura 6 – Arquitetura *ViSIoT* (Adaptado de (NUNES *et al.*, 2016)).

A Figura 7 apresenta um diagrama de sequência sobre o funcionamento da arquitetura em questão. Percebe-se que o Repositório Global (Módulo Repositório) é responsável por retornar ao Módulo Interface os sensores disponíveis para uso. Mas além disso, ele desempenha um papel

fundamental para o funcionamento da *ViSIoT* por possibilitar a obtenção de dados de sensores através de múltiplas fontes (NUNES, 2018).

Figura 7 – Diagrama de Sequência da Arquitetura *ViSIoT* (Adaptado de (NUNES *et al.*, 2016)).

4.3 Metodologia Adotada para o Desenvolvimento da Pesquisa

A metodologia adotada para o desenvolvimento desta dissertação é apresentada a seguir. Na Figura 8 encontram-se as etapas executadas e o detalhamento das principais atividades realizadas:

Revisão da Literatura

A leitura de trabalhos relacionados à integração de diferentes fontes de dados de sensores em *IoT* foi o primeiro passo realizado. Com base na leitura dos trabalhos e análise das estratégias

Figura 8 – Etapas de Execução da Dissertação.

adotadas, para solução do problemas relacionados a interoperabilidade de aplicações *IoT*, foram identificadas as lacunas existentes e definida a abordagem a ser adotada. Inicialmente, foram definidas as perguntas que serviram como guia desta pesquisa, como por exemplo: a) Quais são os cenários utilizados para implantação de aplicações *IoT*?; b) Quais ferramenta são utilizadas para o gerenciamento de recursos em *IoT*?; c) Quais infraestruturas computacionais são utilizadas para o provisionamento de recursos em *IoT*?

Estudo das Ferramentas

Uma das etapas mais importantes para o desenvolvimento do Módulo Repositório foi o estudo das ferramentas e tecnologias associadas a este projeto. Inicialmente, foi realizado um estudo sobre os principais *middlewares* disponíveis em *IoT*. Após o levantamento foi analisado o funcionamento destas ferramentas, identificando as possíveis estratégias para integração com o Módulo Repositório. A priori foram estudados alguns *middlewares* de código fonte aberto presentes na literatura, são eles o *GSN*, *OpenIoT* e o *Xively*. Inicialmente, foram realizados experimentos preliminares com a plataforma *GSN*. Além disso, outras ferramentas foram estudadas, sendo elas o *OpenIoT* e o *Kaa*. Por fim, foram anotadas as principais características e limitações das ferramentas mencionadas de forma a compor os requisitos funcionais do Módulo Repositório.

Desenvolvimento do Módulo Repositório

O Módulo Repositório exerce diferentes funcionalidades que visam auxiliar o gerenciamento de sensores. A Figura 9 exemplifica através de um diagrama de caso de uso as ações que os clientes do módulo podem realizar. Observa-se que, é possível realizar o gerenciamento das informações geridas pelo módulo, além de efetivar a requisição de dados e informações sobre as fontes de dados disponíveis. Além de permitir que diferentes clientes possam requisitar e gerenciar os dados coletados pelos sensores. A seguir são descritas as soluções desenvolvidas que tem como objetivo simplificar a integração de fontes de dados heterogêneos em *IoT*.

Figura 9 – Diagrama de Caso de Uso do Módulo Repositório.

IoTDSM: Internet of Things Data as Service Module

O *IoTDSM* baseou-se na proposta do Módulo Repositório, apresentado inicialmente pela arquitetura *ViSIoT* mas que se diferencia pela utilização dos conceitos de *SOA* e *DaaS*. A Figura 10 mostra uma visão geral do *IoTDSM*:

- ***Data Source Management***: refere-se à conexão com diferentes dados de sensores, realiza o tratamento, conversão e inserção dos dados em bancos de dados e oferece um mecanismo flexível, chamado *Wrapper*, para a manipulação de dados heterogêneos;
- ***Database Management***: possibilita a conexão e administração de diferentes tipos de bancos de dados (*SQL* e *NoSQL*);
- ***File System***: gerencia dados não estruturados (e.g., imagem, áudio e vídeo). Nesse sentido, outros tipos de sistemas de arquivos podem ser utilizados (e.g., Sistemas de Arquivos Distribuídos);
- ***Conversion***: habilita a conversão de dados em diferentes formatos, sendo suportados o *JSON* e o *XML*. A motivação de sua escolha deve-se ao fato de serem especificações estabelecidas pela *World Wide Web Consortium (W3C)* e *Internet Engineering Task Force (IETF)* e;
- ***Communication***: permite o recebimento de informações enviadas por um cliente *HTTP*. Este módulo permite também o acoplamento de outros protocolos de comunicação, por exemplo, *MQTT*, *CoAP*, *AMQP*, *WebSockets*, entre outros.

Figura 10 – Diagrama de Arquitetura do *IoTDSM*.

O diagrama de sequência expresso na Figura 11 representa o fluxo de operações do *IoTDSM*, descrito a seguir:

- **Data Consumption**: requisita dados e comunica com diferentes tipos de clientes por meio de uma interface de comunicação RESTful API;
- **Processing Engine**: processa informações requisitadas pelos clientes e habilita a serialização e deserialização de dados em diferentes formatos;
- **Data Storage**: provê uma camada de persistência de dados que suporta diferentes tipos de bancos de dados (*SQL* ou *NoSQL*);
- **Wrapper**: abstração que permite a conexão com diferentes tipos de dados de sensores e;
- **Data Source**: armazena dados de sensores disponíveis em diferentes formatos.

Observamos através do diagrama que o fluxo ocorre da seguinte maneira: (i) o *Data Consumption* realiza uma requisição ao *Process Engine*; (ii) o *Process Engine* consulta o *Data Storage* para saber quais são os sensores disponíveis; e, (iii) os dados são retornados pelo *Process Engine*, que serializa e retorna os dados requisitados. Uma vez que os dados estão disponíveis, o processo de recuperação executa os seguintes passos: (i) o *Data Consumption* solicita ao *Process*

Figura 11 – Diagrama de Sequência do *IoTDSM*.

Engine os sensores disponíveis, e (ii) os dados referentes ao sensor requisitado são consultados e serializados no formato desejado, sendo por fim enviados ao *Data Consumption*. Observa-se que os processos *Wrapper* e *Data Source* são executados em *loop*, o que permite a integração contínua de diferentes tipos de dados de sensores. Além disso, é importante frisar que a definição do *Wrapper* é de responsabilidade do usuário que detém o domínio sobre os dados. Portanto, a presença de um especialista, que detém o conhecimento sobre os dados processados, é necessário para a construção de um novo *Wrapper*.

O Código Fonte 1 ilustra o funcionamento base de um *Wrapper* no *IoTDSM*. Importante destacar que a classe *Wrapper* baseia-se na implementação de uma interface chamada *IDeserializer* que obriga a implementação dos seguintes métodos: *readObject()*, *readArray()*, *loadContent(String... args)* e *close()*. O fluxo de execução do *Wrapper* se inicia no método *loadContent(String... args)* que recebe como parâmetro informações sobre os dados que serão processados pelo *Wrapper*. Em seguida, tendo as referências necessárias para leitura dos dados os métodos *readObject()* ou *readArray()* são utilizados para leitura de uma ou mais linhas do *stream* de dados. Importante frisar que nestes métodos o usuário detetor do domínio dos dados fica responsável por implementar todas as transformações necessárias para o seu armazenamento

no modelo de banco de dados utilizado no *IoTDSM*. Por fim, após finalizar o transformações e armazenamento dos dados, o método *close()* fica com a responsabilidade de finalizar todas as transações envolvidas durante o processamento de importação dos dados.

Código-fonte 1 – Exemplo Base de uma classe *Wrapper*.

```

1 package edu.usp.icmc.lasdpd.serialization;
2
3 import edu.usp.icmc.lasdpd.model.*;
4 import java.util.*;
5 import java.io.File;
6 import java.io.FileNotFoundException;
7
8 /**
9  * University of São Paulo
10 * Internet of Things Data as Service Module (IoTDSM)
11 * @author Vinicius Aires Barros <viniciusaires@usp.br>
12 */
13 public class WrapperExample implements IDeserializer {
14
15     //Classe utilizada para leitura do stream de Dados de Entrada
16     private Scanner fileStream;
17
18     //Classe que mapeia a tabela tb_sensor_source
19     private SensorSource sensorSource;
20
21     //Classe que mapeia a tabela tb_sensor_measure_type
22     private SensorMeasureType sensorMeasureType;
23
24     //Classe que mapeia a tabela tb_sensor
25     private Sensor sensor;
26
27     //Classe que mapeia a tabela tb_sensor_measure
28     List<SensorMeasure> sensorMeasures = new ArrayList<>();
29
30     /*
31     * Método responsável por ler e processar apenas uma linha de um stream de dados.
32     */
33     @Override
34     public Object readObject() {
35     }
36
37     /*
38     * Método responsável por ler e processar múltiplas linha de um stream de dados.
39     */
40     @Override
41     public List<Object> readArray() {
42     }
43
44     /*
45     * Método responsável por receber um arquivo como entrada e retornar o stream de dados
46     */
47     @Override
48     public boolean loadContent(String... args) {
49     }
50
51     /*
52     * Método responsável por finalizar o stream de dados
53     */

```



```

54     @Override
55     public void close() {
56     }
57 }

```

Ainda se tratando do *IoTDSM*, temos que o Sistema de Gerenciamento de Banco de Dados (SGBD) é escolhido durante a inicialização do sistema no momento em que o arquivo de configuração do sistema é carregado. Este informa os parâmetros de conexão com o banco de dados. A Figura 12 ilustra o diagrama Entidade e Relacionamento (ER) que permitiu o suporte do *IoTDSM* à bancos de dados relacionais (*SQL*). Como é possível observar, cinco tabelas foram definidas para indexar informações relacionadas à localização dos dados, tipos de medidas e medidas de sensores. A escolha dos bancos de dados *PostgreSQL* e *MongoDB* deve-se ao fato de estes estarem entre os principais bancos de dados de código aberto disponíveis no mercado¹. Além disso, o modelo ER foi também adaptado para suportar o banco de dados orientado a documentos *MongoDB* (Figura 13), uma vez que suas estruturas de armazenamento diferem do modelo relacional. O modelo adaptado é semelhante ao apresentado, exceto pela não definição das chaves estrangeiras.

Deve-se salientar que o sistema de armazenamento de dados desenvolvido é flexível e possibilita a persistência de dados estruturados e não estruturados. Ele também provê uma *API* que possibilita a manipulação de dados por meio de uma interface *Representational State Transfer (REST)*, que utiliza diretivas do protocolo *HTTP* para indicar as ações realizadas. Operações *GET* e *POST* foram implementadas para possibilitar a execução de operações ao sistema desenvolvido.

Figura 12 – Modelo Entidade Relacionamento.

¹ <https://db-engines.com/en/ranking>

Figura 13 – Modelo Orientado a Documentos.

A API implementada para a manipulação dos dados segue o padrão *REST*, no qual os verbos *HTTP* servem como indicativo de qual ação deve ser realizada pelo servidor. Os métodos implementados nela encontram-se na Tabela 5. Considerando os parâmetros de entrada, tem-se que o *offset* e *limit* servem para controlar a lista de objetos retornados pela *RESTFull API*. O primeiro significa o número de objetos ignorados desde o início da lista, enquanto o segundo é utilizado para limitar o número de objetos retornados. Além disso, o parâmetro *output_format* indica qual o formato da resposta do servidor (*XML*, *JSON* ou *CSV*).

Tabela 5 – Endpoints disponibilizados pela API.

URL da Chamada	Método (verbo HTTP)	Parâmetros de entrada	Descrição
/sensor	GET	<i>offset</i> (opcional), <i>limit</i> (opcional), <i>output_format</i> (opcional)	Retorna uma lista de todos os sensores cadastrados no sistema.
/sensor/{id_do_sensor}/measure	GET	<i>output_format</i>	Retorna uma lista de medidas possíveis do sensor
/sensor/{id_do_sensor}/measure/{id_da_medida}/{data_início}/{data_fim}	GET	<i>output_format</i>	Retorna uma lista de valores medidos de uma especificada medida de um sensor em um intervalo de tempo. Os parâmetros de data devem estar no formato <i>ISO 8601</i> . O parâmetro {data_fim} na URL é opcional.
/sensor/upload	POST	-	Recebe um arquivo <i>JSON</i> , <i>XML</i> ou <i>CSV</i> do <i>Open Weather Map</i> para armazenar os dados. Deve ser enviado no corpo da requisição com o nome de "file".

Por fim, a Tabela 6 descreve a linguagem de programação e *frameworks* utilizados para construção do *IoTDSM*.

Tabela 6 – Ferramentas utilizadas para a construção do *IoTDSM*.

Ferramenta	Versão	Descrição
Java	1.8	Linguagem de Programação
Spark Java	2.8.0	<i>Micro Web Framework</i>
JPA Hibernate	5.2.12	Framework de Persistência de Dados Objeto/Relacional
GSON	2.7	<i>Parser JSON</i>
Xstream	1.4.10	Parser XML
PostgreSQL JDBC	42.2.1	Conector para Banco de Dados
Mongo Java Driver	3.5.0	Conector para Banco de Dados

IoTM²B: Internet of Things Multi-Protocol Message Broker

Neste trabalho estendemos O *IoTDSM* para suportar diferentes protocolos de comunicação. Nesse sentido, foi implementada uma estratégia multi-protocolo chamada *Internet of Things Multi-Protocol Message Broker (IoTM²B)* que possibilitou a integração com os protocolos *HTTP*, *MQTT* e *CoAP* (Figura 14). O *IoTM²B* permite a comunicação entre protocolos de comunicação distintos através da extração do corpo da mensagem, para posteriormente ser enviado ao sistema destinatário. Entretanto, ela também permite a integração com diferentes serviços *IoT* e sistema embarcados, auxiliando na construção de sistemas interoperáveis.

Figura 14 – Diagrama de Arquitetura do *IoTM²B*.

A Figura 15 ilustra o fluxo de execução do *IoTM²B*. Observa-se que o objeto *IoT Device or Service* envia uma requisição *HTTP*, *MQTT* ou *CoAP* ao serviço e, em sequência, o *Protocol Translator* extrai o corpo da mensagem e realiza uma nova requisição *POST HTTP* enviando os dados para um sistema destinatário. Diversos são os desafios relacionados à essa dinâmica, devido à translação de múltiplos protocolos e o baixo desempenho de bibliotecas de protocolos mais recentes, o que dificulta a interoperabilidade de sistemas *IoT*. Por fim, a Tabela 7 descreve a linguagem de programação e *frameworks* utilizados para construção do *IoTM²B*.

Figura 15 – Diagrama de Sequência do *IoTM²B*.Tabela 7 – Ferramentas utilizadas para a construção do *IoTM²B*.

Ferramenta	Versão	Descrição
Java	1.8	Linguagem de Programação
Eclipse Vert.x	3.5.0	Web Server HTTP e MQTT
Eclipse Californium	1.0.6	Web Server CoAP
Eclipse Paho	1.4	MQTT Client
CoAPthon	4.0	CoAP Client

4.4 Considerações Finais

Neste capítulo foi abordado o desenvolvimento desta dissertação de maneira mais detalhada. Para isso, foi descrita a arquitetura *ViSIoT*, utilizada como base para a construção do *IoTDSM*. Em seguida, apresentou-se a concepção e funcionamento do mecanismo desenvolvido, além de uma extensão de suas funcionalidades, que possibilitou a sua integração com diferentes tipos de fontes/formatos de dados e protocolos de comunicação.

AVALIAÇÕES EXPERIMENTAIS

5.1 Considerações Iniciais

Este capítulo apresenta as avaliações experimentais efetuadas para análise dos resultados desta dissertação. Elas foram divididas em três cenários: (i) avaliação de desempenho do *Middleware GSN*; (ii) avaliação de desempenho da abordagem *IoTDSM*; (iii) avaliação do mecanismo multi-protocolo *IoTM²B*; (iv) avaliação de uma arquitetura que realiza a classificação de emoções de usuários em um ambiente de casa inteligente e *Healthcare*. Importante destacar que parte dos resultados foram obtidos em conjunto com colaboração de alunos de graduação e pós-graduação cujo o escopo é parte integrante desta pesquisa.

5.2 Avaliação de Desempenho

Segundo Jain (1990) a avaliação de desempenho é uma metodologia fundamental para aquisição de novos sistemas computacionais. Além disso, ele também estabelece alguns termos para descrever um planejamento de experimento (JAIN, 1990): (i) Variável de Resposta: principal variável de saída de um experimento; (ii) Fator: item com maior influência no valor de saída de uma variável de resposta; (iii) Nível: valores alternativos que um fator pode assumir. Também descreve-se o conceito de experimento fatorial completo, que consiste na avaliação de todas as possibilidades de combinações entre fatores e níveis de um experimento. A seguir, são apresentadas as avaliações de desempenho conduzidas nesta dissertação.

5.3 Avaliação de Desempenho do *Middleware GSN*

A atividade inicial para a avaliação do desempenho do *middleware GSN* consistiu no levantamento das métricas de desempenho necessárias para condução das demais avaliações. Este cenário foi executado em um ambiente computacional de Computação em Névoa (*Fog*

Computing). Para isso, utilizou-se diferentes versões do *Raspberry Pi* na execução do referido *middleware*. Parte dos resultados deste estudo foi obtido com base em uma cooperação com um Trabalho de Conclusão de Curso (TCC) do aluno de graduação Herik Magalhães Serra de mesma orientação. Além disso, os resultados deste experimento foram publicados na 10ª Escola Regional de Alto Desempenho do estado de São Paulo (ERAD-SP 2019)¹. A seguir, será descrita a avaliação conduzida, os resultados e as conclusões obtidas.

Setup Experimental & Configuração de Ambiente

A metodologia adotada para a avaliação de desempenho consistiu na definição do ambiente e de parâmetros experimentais, representadas nas Tabelas 8, 9 e 10. Nesse contexto, a análise compreendeu um teste de carga aplicado na *API RESTful*, utilizada pelo *GSN* para o gerenciamento de dados de sensores. Para isso, efetuou-se um conjunto de requisições a fim de constatar o tempo de resposta do *middleware*.

Esta abordagem teve como objetivo identificar a viabilidade da utilização do *GSN* em execução no *Raspberry Pi*, além de apontar qual das suas versões obteve um melhor desempenho durante os testes. A Tabela 8 apresenta os parâmetros definidos para a execução do experimento. Observa-se o uso de uma variação de 10, 100 e 1.000 requisições ao *Raspberry Pi*.

Os experimentos executados foram definidos com base em duas distribuições estatísticas, a *uniforme* e *poisson*. Para isso, utilizou-se a ferramenta *JMeter*² que permitiu o uso dessas distribuições para a geração de carga de trabalho de forma simplificada. Importante destacar que as distribuições estatísticas foram utilizadas para determinar a taxa de chegadas das requisições ao *GSN*. Além disso, cada um dos experimentos foram executados 30 vezes a fim de calcular a média e o intervalo de confiança dos tempos de respostas aferidos.

Tabela 8 – Lista de Parâmetros do Experimento.

Parâmetros do Experimentos	
Número de Requisições	10, 100 e 1.000
Distribuição Estatística	Uniforme e Poisson
Número de Repetições	30

Para execução do teste foram utilizadas três máquinas, cujas configurações de *hardware* encontram-se na Tabela 9. É preciso ressaltar que todas elas apresentavam a mesma configuração de *software* e *hardware*. Além do mais, foi desenvolvido um *script* para facilitar a instalação do *GSN* em ambientes Debian Linux.

A respeito das configurações de *hardware* das versões do *Raspberry Pi* utilizadas, tem-se que a principal variação corresponde ao tamanho da RAM e a quantidade de núcleos do

¹ <<https://www2.sbc.org.br/cradsp/eradsp/2019/>>

² <http://jmeter.apache.org/>

Tabela 9 – Configurações de *Hardware* (Geradores de Carga).

Sistema Operacional	Ubuntu 16.04.2 LTS
CPU	Intel Core 2 Quad Processor (6 MB Cache, 2.66 GHz)
Memória	8 GB RAM DDR3
Rede	Gigabit
Número de Máquinas	3

processador. Por fim, a Figura 16 ilustra a configuração de experimento utilizada para execução do teste de carga. Observa-se que as máquinas clientes (geradores de carga de trabalho) e os *Raspberry Pi*'s estão em redes *Local Area Network (LAN)* distintas. Cada *Raspberry Pi* recebeu um *IP* público e uma porta e, a partir desses endereços, foram submetidas as requisições. A comunicação entre as duas redes foi realizado por meio de uma rede privada com a finalidade de avaliar o desempenho do *GSN* em um ambiente de rede isolado.

Tabela 10 – Comparação de *Hardware* (*Raspberry Pi*).

	<i>Raspberry Pi 1 B+</i>	<i>Raspberry Pi 2 B</i>	<i>Raspberry Pi 3 B</i>
CPU	ARM 11 @ 700 MHz	Quad Cortex A7 @ 900 MHz	Quad Cortex A53 @ 1.2 GHz
RAM	512 MB SDRAM	1 GB SDRAM	1 GB SDRAM
Ethernet	10/100	10/100	10/100

Figura 16 – Setup Experimental.

Análise dos Resultados

A partir do experimento realizado, obteve-se dados referentes ao tempo de resposta obtido pelo *Raspberry Pi* na execução do teste de carga aplicado ao *middleware GSN*, os quais serão apresentados a seguir. Um ponto importante a ser ressaltado é que o *Raspberry Pi 1* (*single-core*) foi desconsiderado nesta avaliação comparativa, devido a ocorrência de falhas no atendimento das requisições ($\pm 3\%$ das requisições), o que tornaria injusta a sua comparação

com o *Raspberry Pi 2* e *3* (*quad-core*). Dessa forma, concluiu-se que o *Raspberry Pi 1* apresenta configurações de *hardware* insuficientes para execução do *GSN*.

Na Figura 17 é possível observar o teste realizado com um conjunto de 10 requisições, no qual os tempos de resposta são muito próximos e, em alguns casos, ocorre a intersecção entre os valores obtidos. Porém, a distribuição uniforme apresenta uma variação do tempo de resposta maior quando comparado com a distribuição *Poisson*. Em todas situações, o desempenho do *Raspberry Pi 3* foi visivelmente superior. Portanto, constata-se que o tempo de resposta do *Raspberry Pi 3* foi ligeiramente menor, considerando o uso de 10 requisições, em ambos os cenários. Os resultados obtidos nas Figuras 18 e 19 ilustram um comportamento linear, com tendência de afastamento das retas a medida em que o número de requisições aumentam. Além disso, a distância entre as retas é maior na distribuição uniforme, o que pode ser observado em todos os resultados apresentados. Assim, nota-se que em todos os casos (100 e 1.000 requisições), o desempenho do *Raspberry Pi 3* foi superior, o que evidencia a possibilidade de uso do *GSN* neste tipo de dispositivo, como um nó de processamento (*fog node*) de Computação em Névoa. Entretanto, o tempo de resposta com 100 e 1.000 requisições foi significativamente alto o que torna necessária a utilização de múltiplos dispositivos em rede para a diminuição da latência.

Figura 17 – Carga de Trabalho com 10 Requisições.

5.4 Avaliação de Desempenho do *IoTDSM*

A avaliação de desempenho da abordagem *IoTDSM* objetivou identificar quais são os *overheads* relacionados à conversão, armazenamento e recuperação de dados de diferentes formatos. Para isso, foi desenvolvido um *wrapper* que possibilitou o emprego de uma base de dados climáticas de +200.000 cidades³. É interessante ressaltar que o ambiente experimental montado ofereceu suporte à verificação do desempenho de tipos diferentes de bancos de dados e formatos de dados. Parte dos resultados deste estudo foi obtido com base em uma cooperação com um Trabalho de Conclusão de Curso (TCC) do aluno de graduação Leonardo Beck Prates de

³ *Open Weather Map*: <<https://openweathermap.org/forecast16>>

Figura 18 – Carga de Trabalho 100 Requisições.

Figura 19 – Carga de Trabalho com 1.000 Requisições.

mesma orientação. Além disso, os resultados deste experimento foram publicados no *21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2018)*⁴. O Código-fonte 2 apresenta uma amostra da base de dados em formato *JSON* da cidade Shuzenji do Japão. A seguir serão descritos a avaliação de desempenho conduzida, os resultados e as conclusões.

Código-fonte 2 – Amostra de Dados climáticos de Shuzenji - Japão.

```

1 {
2   "cod ":"200",
3   "message ":0.0032,
4   "city ":{
5     "id ":1851632,
6     "name ":" Shuzenji ",
7     "coord ":{
8       "lon ":138.933334,
9       "lat ":34.966671
10    },
11    "country ":" JP ",
12    "timezone ":32400

```

⁴ <<https://doi.org/10.1145/3242102.3242141>>

```

13 },
14 "cnt":10,
15 "list":[{"
16     "dt":1406080800,
17     "temp":{"
18         "day":297.77,
19         "min":293.52,
20         "max":297.77,
21         "night":293.52,
22         "eve":297.77,
23         "morn":297.77
24     },
25     "pressure":925.04,
26     "humidity":76,
27     "weather":[{"
28         "id":803,
29         "main":"Clouds",
30         "description":"broken clouds",
31         "icon":"04d"
32     }],
33 }],
34 }

```

Setup Experimental

Com o propósito de avaliar o desempenho do *IoTDSM* foi desenvolvido um *Wrapper* que possibilitou a integração com o *Open Weather Map (OWM)*, uma *API* que contém dados climáticos de cidades ao redor do mundo⁵. Os dados utilizados continham 842.220 sensores com 13.821.204 registros e inicialmente estavam formatados em *JSON*. Eles continham uma amostra de 16 dias de informações sobre clima das cidades (e.g., temperatura, pressão e umidade) que posteriormente foi armazenada nos bancos de dados *PostgreSQL* e *MongoDB*. A avaliação deste cenário foi realizada por meio de um teste de carga que teve como objetivo retornar os sensores mapeados pelo *IoTDSM*. Importante destacar que as distribuições estatísticas Uniforme e Gaussiana foram utilizadas para determinar a taxa de chegada das requisições do teste de carga. A Tabela 11 apresenta os fatores e níveis de experimentos utilizados.

Tabela 11 – Experimental Design.

Fator	Nível
Formato de Dados	<i>JSON, XML</i>
Banco de Dados	<i>PostgreSQL, MongoDB</i>
Número de Requisições	2.000, 8.000
Distribuição Estatística	Uniforme, Gaussiana

⁵ <<https://openweathermap.org/about>>

Configuração de Ambiente

Foram definidos diferentes fatores e níveis para a configuração do ambiente, como por exemplo, o formato dos dados, banco de dados, número de requisições e distribuições estatísticas utilizadas para geração da carga de trabalho. Após definido o *setup* experimental, aplicou-se a metodologia de avaliação fatorial completo 2^k , em que k é o número de fatores e 2 o número de níveis, resultando em $2^4 = 16$ experimentos. Além disso, o número de replicações utilizado foi 30 e tinha como finalidade possibilitar a análise dos intervalos de confiança dos tempos de respostas analisados. A Tabela 12 apresenta os parâmetros fixos utilizados neste cenário de avaliação.

Tabela 12 – Parâmetros Fixos de Experimento.

Operação	Descrição	Valor
Número de Sensores	Sensores por Requisição	1.000
<i>Random Delay Maximum</i>	Distribuição Uniforme	100ms
<i>Constant Delay Offset</i>	Distribuição Gaussiana	100ms
<i>Deviation</i>	Distribuição Gaussiana	300ms
<i>Java Heap Space</i>	Alocação Máxima de Memória	32GB

A Tabela 13 apresenta as configurações de *hardware* e a Figura 20 o *setup* experimental utilizado nesta avaliação de desempenho. Uma máquina, representada por um conjunto de *CPU's* é responsável por gerar carga de trabalho ao *IoTDSM*, que por sua vez, utiliza diferentes diretivas do protocolo *HTTP* para enviar e receber dados, por exemplo, *GET/POST*, dentre outras.

Importante ressaltar que para o *IoTDSM* atenda uma requisição alguns passos são realizados anteriormente. Primeiramente, o servidor recebe uma solicitação e realiza uma consulta no banco de dados (*PostgreSQL* ou *MongoDB*), em seguida, os dados são serializados em *JSON* ou *XML* e depois transmitidos como resposta aos clientes da requisição. Além disso, importante notar que este processo é realizado com diferentes cargas de trabalho processando requisições de forma simultânea. Por fim, é importante salientar que o “tempo de resposta” foi a variável de resposta considerada neste experimento.

Tabela 13 – Especificações de *Hardware*.

CPU	Intel Core I7-4790 3.60GHZ (8 cores)
RAM	32GB DDR3
Armazenamento	1TB HDD
Rede	Gigabit Ethernet
Sistema Operacional	Ubuntu 16.04.3 LTS
Número de Máquinas	2

Análise dos Resultados

Esta seção apresenta a análise de influência de fatores e uma comparação entre cada fator/nível de avaliação. Adicionalmente, aplicou-se um teste estatístico sobre os dados analisados e foi constatado que eles apresentavam 95% de confiança.

Figura 20 – Setup Experimental.

Teste de Carga

A respeito do teste de carga aplicado, é possível analisar a influência de fatores na Figura 21, por meio da qual foram consideradas todas as possibilidades de combinações de experimento. Nesse sentido, o tempo médio do tempo de resposta foi comparado para cada combinação de nível, para identificação de qual o fator/nível com maior influência no tempo de resposta.

A Figura 21 (a) apresenta uma comparação entre os níveis *JSON* e *XML*, este último representando o fator com maior influência, por apresentar uma média do tempo de resposta superior. Conforme exibido, o tempo de resposta para o *JSON* foi 19.17s, enquanto o *XML* foi 61.54s. É possível ainda visualizar esse dado em função do tipo de banco de dados na Figura 21. Tendo como foco a figura anterior, tem-se em (b) a média do tempo de resposta em função do tipo de banco de dados, na qual os níveis avaliados se referem aos bancos de dados *PostgreSQL* e *MongoDB*. Nessa situação o *MongoDB* apresentou um tempo médio de resposta superior de 47.89s, enquanto o *PostgreSQL* obteve 32.83s. Em (c) aponta-se uma análise de influência de fatores para o número de requisições. Como esperado, a média do tempo de resposta para 8.000 requisições foi o nível de maior influência em comparação com 2.000 requisições (13.90s).

Por fim, a Figura 21 (d) apresenta a análise de influência de fatores para dois níveis de distribuições estatísticas, a uniforme e a Gaussiana, aplicadas para geração da carga de trabalho do sistema. Para isso foram utilizados os parâmetros de configuração detalhados na Tabela 12. Pode-se observar que a distribuição uniforme (42s) ostenta uma maior influência no tempo de resposta, superior à distribuição Gaussiana (38s).

A Figura 22 apresenta um conjunto de comparações do tempo de resposta entre todas as combinações de experimentos. Para esse fim, o formato de dados de saída, número de requisições, bancos de dados e distribuições estatísticas foram variados. Assim, a Figura 22(a) representa o tempo de resposta com relação à distribuição uniforme, tal que o banco de dados *PostgreSQL*, com 2.000 ou 8.000 requisições, obteve um tempo de resposta melhor. Com 2.000 requisições o *PostgreSQL* obteve 4.87s, enquanto o *MongoDB* 8.94s. Com 8.000 requisições, o *PostgreSQL* obteve um tempo de resposta de 20.12s e o *MongoDB* 35.35s.

Figura 21 – Análise de Influência de Fatores Principais.

Os resultados utilizando a distribuição Gaussiana podem ser vistos na Figura 22(b), a qual denota um tempo de resposta similar ao da Uniforme. Nesse sentido, com 2.000 requisições, o *PostgreSQL* obteve 4.61s e para o *MongoDB* 8.65s, e com 8.000 requisições 19.14s para o *PostgreSQL* e 35.04s para o *MongoDB*. Portanto, *PostgreSQL* obteve um tempo de resposta menor. É possível analisar ainda os resultados em relação ao tempo de resposta do formato de dados XML na Figura 22.

A Figura 22(c), por sua vez, mostra os seguintes resultados: 19.58s em média para o *PostgreSQL* e 22.94s para o *MongoDB* com 2.000 requisições; 91.99s para o *PostgreSQL* e 118.21s para o *MongoDB* com 8.000 requisições. De acordo com a Figura 22(d), *PostgreSQL* e *MongoDB*, obtiveram respectivamente 16.41s e 18.65s para 2.000 requisições e 71.48s e 108.03s para 8.000 requisições. Portanto, *PostgreSQL* obteve um tempo de resposta menor que

o *MongoDB*, em ambos os formatos de dados *JSON* ou *XML*. De modo geral, pode-se concluir que o formato de dados *JSON* e Banco de dados *PostgreSQL* obtiveram um melhor desempenho quando comparado ao *MongoDB* e *XML*.

Figura 22 – Tempo de Resposta *JSON* x *XML*.

5.5 Avaliação de Desempenho do *IoT²B*

A avaliação de desempenho do *IoT²B* visa verificar quais são as limitações de desempenho do mecanismo multi-protocolo. Para isso, foram conduzidos diferentes cenários de avaliação da abordagem: (i) execução de um ambiente *Machine-to-Machine (M2M)* utilizando a *Raspberry Pi* I e II e um microcontrolador ESP8266 e; (ii) avaliação de utilização da abordagem em um ambiente de Computação em Nuvem. O foco da avaliação foi verificar o desempenho da integração do mecanismo proposto com os protocolos de comunicação *HTTP*, *MQTT* e *CoAP*. Parte dos resultados deste estudo foi obtido com base em uma cooperação com um Trabalho de Conclusão de Curso (TCC) do aluno de graduação Sergio Baptista Jr de mesma orientação. Além disso, os resultados foram publicados no XXV Simpósio Brasileiro de Sistemas Multimídia e Web (*WebMedia*)⁶. A seguir, constam a avaliação conduzida, os resultados e as conclusões desse experimento.

Setup Experimental & Configuração de Ambiente

Para cumprir com o objetivo da avaliação foram definidos dois tipos de experimento: (i) avaliação do tempo de resposta para comunicação entre os protocolos de comunicação em um ambiente de dispositivos embarcados (*M2M*) para troca de mensagens: (a) ESP8266 e *Raspberry Pi* 1; (b) *Raspberry Pi* 1 e *Raspberry Pi* 3; (ii) avaliação da abordagem multi-protocolo em um ambiente de Computação em Nuvem, composta por três máquinas (gerador de carga, abordagem multi-protocolo e sistema destinatário). Ainda, para facilitar a relação de experimentos, eles foram divididos em três (A, B e C), tal que A e B referem-se à avaliação de desempenho da comunicação entre dispositivos embarcados (*M2M*) e o experimento C tem como foco a avaliação da abordagem multi-protocolo em um ambiente de computação em nuvem, em dois tipos de carga de trabalho (C1 e C2). Todos os experimentos foram replicados 30 vezes para o cálculo da média e intervalos de confiança. A Tabela 14 apresenta o desenho do experimento.

Tabela 14 – Desenho de Experimento.

Experimentos	Número de Requisições	Dispositivos	Comunicação
A	5, 25, 50	ESP8266 Raspberry Pi 1	802.11 N
B	5, 25, 50	Raspberry Pi 1 Raspberry Pi 3	Ethernet 10/100
C	C1: 25, 50, 75, 100, 125 C2: 150, 250, 350, 450, 550	Cluster	Ethernet Gigabit

Experimento A

O Experimento A mensurou o tempo de resposta de três protocolos de comunicação: (*MQTT*, *CoAP* e *HTTP*), por meio de sensores físicos (ESP8266 e *Raspberry Pi* 1). O ESP8266

⁶ <<https://doi.org/10.1145/3323503.3349546>>

adquire a temperatura e umidade do ambiente por meio de um sensor físico (AM2301). Em seguida, após o processamento dos dados capturados as informações são enviadas por uma rede *Wi-Fi* (802.11 N) para uma *Raspberry Pi* 1 (Figura 23).

Figura 23 – Fluxo de Comunicação (Experimento A).

Um roteador TP-LINK TL-WR841ND⁷ foi utilizado para conectar os dispositivos. O dispositivo ESP8266 conectou-se por meio de uma rede *Wi-Fi* (802.11 N) e o *Raspberry Pi* 1 através de um cabo *ethernet*. De maneira geral o ESP8266 coletava os dados de temperatura/umidade e, em seguida, os enviava para a *Raspberry Pi* 1. É importante salientar que a geração de carga de trabalho foi baseada em uma distribuição Uniforme com 1s de espaçamento entre as requisições. A Figura 24 ilustra o esquemático utilizado neste *setup* experimental.

Figura 24 – Setup Experimental (Experimento A).

Experimento B

O Experimento B é similar ao A, com foco também no tempo de resposta dos protocolos. Entretanto, os dados são coletados pelo *Raspberry Pi* 1, utilizando um sensor AM2301 para enviar dados de temperatura e umidade para uma *Raspberry Pi* 3 (Figura 25). O mesmo número de requisições e replicações do Experimento A foram aplicados e, ambos os dispositivos apresentam a mesma interface de rede (10/100 Mbps). Além disso, também foi utilizado o roteador TL-WR841ND para conexão dos dispositivos por meio de cabo *ethernet*.

⁷ <https://www.tp-link.com/en/products/details/cat-9_TL-WR841ND.html>

Figura 25 – Fluxo de Comunicação (Experimento B).

Ainda conforme o Experimento A, neste experimento avaliou-se os protocolos *HTTP*, *MQTT* e *CoAP*. Além do mais, a carga de trabalho gerada também foi baseada em uma distribuição Uniforme com um espaçamento de 1s entre requisições. A Figura 26 ilustra o *setup* experimental.

Figura 26 – Setup Experimental (Experimento B).

Experimento C

Três máquinas com alto poder computacional foram utilizadas para avaliar a abordagem multi-protocolo, tal que, para a comunicação entre as máquinas I e II utilizou-se os protocolos *HTTP*, *MQTT* e *CoAP* e, entre as máquinas I e III, o *HTTP* (Figura 27). Ademais, foram definidas duas classes de carga de trabalho: (i) C1 (25, 50, 75, 100 e 125) e (ii) C2 (150, 250, 350, 450 e 550). É importante destacar que as máquinas utilizadas apresentavam o mesmo sistema operacional e configurações de *hardware* (Tabela 15). De forma semelhante aos experimentos anteriores utilizou-se a distribuição Uniforme para a geração de carga de trabalho com 300ms de *delay*.

Figura 27 – Fluxo de Comunicação (Experimento C).

Tabela 15 – Configurações de *Hardware*.

Sistema Operacional	Ubuntu 16.04.3 LTS
CPU	Intel Core I7 4790 3.60GHZ
Memória	32 GB RAM DDR3
Número de Máquinas	3

Análise dos Resultados

Os resultados apresentados abaixo são referentes à análise do tempo de resposta de todas as combinações dos experimentos avaliados. Para o cálculo do intervalo de confiança cada experimento foi replicado em 30 vezes de acordo com a distribuição *t-student*.

Experimento A e B

A Figura 28 apresenta o tempo de resposta (em μs) dos protocolos *MQTT*, *CoAP* e *HTTP*. No cenário com 5 requisições, o protocolo *CoAP* apresentou o melhor desempenho, tal que *HTTP* e *MQTT* foram 203.76x e 1.22x mais lentos. De forma similar, com 25 requisições, os protocolos *HTTP* e *MQTT* foram, respectivamente, 168.41x e 1.23x mais lentos. Por fim, com 50 requisições, *HTTP* e *MQTT*, foram 179.64x e 1.29x mais lentos que o *CoAP*. A Tabela 16 apresenta os resultados obtidos em valores absolutos.

Tabela 16 – Experimento A (Tempo μs).

N. de Requisições	<i>HTTP</i>	<i>MQTT</i>	<i>CoAP</i>
5	105996.90	637.26	520.18
25	86325.93	634.63	512.56
50	87428.16	628.47	486.67

Figura 28 – Tempo de Resposta (Experimento A).

A Figura 29 apresenta o tempo de resposta do Experimento B, neste, o protocolo *MQTT* apresentou os melhores resultados. Com 5 requisições os protocolos *CoAP* e *MQTT* foram 6.80x e 18.19x mais lentos. De forma similar, com 25 requisições os protocolos *CoAP* e *MQTT* foram

7.09x e 17.9x mais lentos. Em um cenário com 50 requisições, o *MQTT* também obteve o melhor tempo de resposta (7.13x *CoAP* e *HTTP* 18.46x). Por fim, a Tabela 17 apresenta os resultados obtidos em valores absolutos.

Tabela 17 – Experimento B (Tempo μs).

N. de Requisições	<i>HTTP</i>	<i>MQTT</i>	<i>CoAP</i>
5	62180.52	3417.77	23251.45
25	61608.42	3441.68	24419.46
50	63272.90	3426.77	24456.78

Figura 29 – Tempo de Resposta (Experimento B).

Experimento C

A Figura 30 expõe o tempo de resposta da abordagem multi-protocolo com carga de trabalho C1, enquanto a Tabela 18 contém a média do tempo de resposta. Neste experimento, o *CoAP* apresentou o melhor tempo de resposta com 25 requisições, o *MQTT* e *HTTP* foram 1.13x e 1.36x lentos. Com 50 requisições o *CoAP* novamente apresentou o melhor desempenho, *MQTT* e *HTTP* foram 1.25x e 1.11x mais lentos. Novamente, com 75 requisições o *CoAP* foi o protocolo mais rápido (*MQTT* 1.20x e *HTTP* 1.10x). Por fim, com 100 requisições o *CoAP* apresentou um melhor desempenho, *MQTT* e *HTTP* respectivamente 1.17x e 1.15x mais lentos.

Tabela 18 – Experimento C1 (Tempo μs).

N. de Requisições	<i>HTTP</i>	<i>MQTT</i>	<i>CoAP</i>
25	3110.54	2590.37	2279.03
50	2214.62	1969.69	1771.29
75	1996.69	1834.26	1657.57
100	1785.46	1755.66	1519.48

A Figura 31 ilustra o tempo de resposta do experimento C2, e a Tabela 19 apresenta os valores absolutos as médias do tempo de reposta. Com 150 requisições o *CoAP* apresentou o melhor desempenho, sendo o *MQTT* 1.07x e *HTTP* 1.1x mais lentos. No cenário com 250

Figura 30 – Tempo de Resposta (Experimento C1).

requisições o *CoAP* obteve o menor tempo de resposta (*MQTT* 1.07x e *HTTP* 1.13x). Também com 350 requisições, o *CoAP* apresentou o melhor desempenho, nesse contexto, o *MQTT* e o *HTTP* foram 1.07x e 1.12x mais lentos, respectivamente. Por fim, com 450 requisições o *CoAP* manteve-se com o menor tempo de resposta, sendo *MQTT* 1.10x e *HTTP* 1.11x mais lentos.

Tabela 19 – Experimento C2 (Tempo μs).

N. de Requisições	<i>HTTP</i>	<i>MQTT</i>	<i>CoAP</i>
150	1756.35	1701.47	1584.22
250	1662.51	1568.83	1462.34
350	1566.52	1494.01	1392.63
450	1529.61	1511.37	1366.36

5.6 Avaliação de Desempenho da Arquitetura *ENLACE*

Além dos mecanismos avaliados por esta dissertação foi realizado uma avaliação de desempenho de uma arquitetura chamada *EmotionAl Level Architecture Communication hEalthcare (ENLACE)*. O *ENLACE* é uma arquitetura baseada em camadas que realiza o processamento de emoções em ambientes de *Healthcare*. Este trabalho foi obtido em parceria com o aluno de doutorado Leandro Y. Mano do ICMC USP orientado pelo professor Jó Ueyama. Este trabalho baseou-se no *IoTDSM* estendendo as suas funcionalidades para o reconhecimento de emoções de usuários de uma casa inteligente (*Smart Home*) e foi publicado no *Mobile Information System Journal*⁸.

⁸ <<https://doi.org/10.1155/2019/7329187>>

Figura 31 – Tempo de Resposta (Experimento C2).

A Figura 32 ilustra a arquitetura do *ENLACE*, a qual pode-se ser observado que o *IoTDSM* foi instanciado em diferentes camadas da arquitetura. Na arquitetura *ENLACE* foram utilizados diferentes algoritmos de aprendizagem de máquina (*kNN*, *SVM*, *Lógica Fuzzy*, *Árvore de Decisão* e *Redes Bayesianas*) para classificação de emoções por meio de fotos da face (*FA - Face Analysis*) e batimentos cardíacos (*HRA - Heart Rate Analysis*). Para tal funcionalidade foi desenvolvido um *Wrapper* que permitiu o armazenamento e processamentos dos dados utilizados. Além disso, a arquitetura implementa um mecanismo de tomada de decisão (*offloading*) que permitiu o seu uso para processamento de emoções em diferentes ambientes de computação em *IoT* (e.g., Dispositivo Local, Servidor Local ou Computação em Nuvem). Para isso, o algoritmo baseou-se na avaliação de métricas de consumo energético e de tempo de resposta dos ambientes de execução da arquitetura. Por fim, avaliou-se o desempenho da arquitetura em execução em uma *Raspberry Pi*, Servidor Local e uma máquina de alto poder computacional na nuvem.

Setup Experimental & Configuração do Ambiente

A Figura 33 apresenta um cenário de utilização da arquitetura *ENLACE* em uma Casa Inteligente. O ambiente em questão é protótipo de casa inteligente o qual diversos sensores (batimento cardíacos e câmeras) são utilizados para o reconhecimento de emoções dos residentes de uma casa. Além disso, na casa inteligente são configurados diversos dispositivos de processamento local (*Raspberry Pi*) e um servidor local que podem processar os dados capturados ou fazer o *offload* do processamento dos dados para a nuvem.

Por fim, a Tabela 20 apresenta a configuração do ambiente experimental utilizado neste cenário de avaliação. Observa-se que os experimentos foram conduzidos utilizando diferentes ambientes de execução do *ENLACE* (Dispositivo Local, Servidor Local e Nuvem). Importante

HRA). Além disso, variou-se a quantidade de requisições/min com a finalidade de verificar o desempenho da arquitetura com diferentes cargas de trabalho.

Tabela 20 – Configuração do Ambiente Experimental.

Especificações de <i>Hardware</i> .			
Ambiente	Processador	Memória	
Dispositivo Local	4 x ARM Cortex-A53, 1.2GHz	1GB LPDDR2	
Servidor Local	Intel Core 2 Quad 9400, 2.66GHz	8 GB DIMM	
Nuvem	AMD FX 8-Core, 4.0Ghz	32 GB RAM DDR3	

Desenho de Experimento.			
	Dispositivo Local	Servidor Local	Nuvem
Serviço	FA and HRA	FA and HRA	FA and HRA
#requisições	200	200	200
#requisições/min	16, 24, 32	32, 64, 96	64, 96, 128

Análise dos Resultados

As Figuras 34, 35 e 36 ilustram a variação dos tempos de resposta utilizando o *Raspberry Pi* como dispositivo local de execução do *ENLACE*. Observa-se que para os cenários de avaliação *HRA + FA* e *HRA* o tempo de resposta variou de forma estável para 16 req/min e 24 req/min, em contraponto o cenário com 32 req/min o tempo de resposta da aplicação se comportou de forma crescente e diminuindo ao se aproximar do fim do experimento. Entretanto, diferentemente o cenário *FA* obteve um tempo de resposta uniforme com poucos picos observados em todo o teste. Portanto, pode-se concluir que para um cenário de utilização do *ENLACE* em execução em uma *Raspberry Pi* apresenta um desempenho satisfatório para cargas de trabalho com baixo número de requisições por minuto.

Em seguida, as Figuras 37, 38 e 39 apresentam os resultados obtidos com a utilização do *ENLACE* em um servidor local. Observa-se que de forma similar ao primeiro experimento, os cenários *HRA + FA* e *HRA* o tempo de resposta variou de forma estável para 32 req/min e 64 req/min, entretanto, com 96 req/min o tempo de resposta se comportou de forma crescente apresentando uma diminuição ao se aproximar do fim do experimento. Além do mais, assim como no primeiro experimento, o cenário *FA* apresentou um tempo de resposta uniforme com a presença de alguns picos de variação. Portanto, neste caso conclui-se que para um cenário de execução do *ENLACE* em um servidor local o desempenho da arquitetura foi satisfatório para cargas de trabalho maiores mas não muito elevado.

Por fim, as Figuras 40, 41 e 42 ilustram os tempos de resposta do *ENLACE* em um ambiente de Computação em Nuvem. Observa-se que de forma similar ao primeiro e segundo experimento, os cenários *HRA + FA* e *HRA* o tempo de resposta se comportou de forma uniforme

para 64 req/min e 96 req/min, entretanto, com 128 req/min o tempo de resposta se comportou de forma crescente mas que ao se aproximar do fim do teste apresentou uma diminuição do tempo de resposta. Além disso, assim como no primeiro e segundo experimento o cenário *FA* apresentou um tempo de resposta uniforme podendo ser observado alguns picos de variação em todas as cargas de trabalho utilizadas. Portanto, conclui-se que para um cenário de execução do *ENLACE* em um servidor na nuvem o desempenho da arquitetura foi satisfatório mesmo para cargas de trabalhos maiores.

Figura 34 – Tempo de resposta dos serviços *HRA + FA* em um dispositivo local.

Figura 35 – Tempo de resposta do serviço *HRA* em um dispositivo local.

Figura 36 – Tempo de resposta do serviço *FA* em um dispositivo local.

Figura 37 – Tempo de resposta dos serviços *HRA* +Figura 38 – Tempo de resposta do serviço *HRA* em um servidor local.

Figura 39 – Tempo de resposta do serviço *FA* em um servidor local.

5.7 Considerações Finais

Este capítulo apresentou as avaliações experimentais conduzidas nesta dissertação. Foram detalhados os experimentos dirigidos e exposta uma análise estatística sobre os resultados. Em todos os experimentos, a avaliação de desempenho teve como finalidade auxiliar na identificação dos *overheads* de desempenho das soluções propostas. Cada cenário de avaliação teve seus resultados condensados em artigos publicados/submetidos em conferências/*journals* científicos relacionados à área de pesquisa. Constata-se, por meio dos resultados obtidos, um bom desempenho do *IoTDSM*, *IoTM²B* e *ENLACE*, o que possibilita a investigação de novos trabalhos em aplicações *IoT* de larga escala.

Figura 40 – Tempo de resposta dos serviços *HRA* +Figura 41 – Tempo de resposta do serviço *HRA* em *FA* em um ambiente de nuvem.

Figura 42 – Tempo de resposta do serviço *FA* em um ambiente de nuvem.

PUBLICAÇÕES E SUBMISSÕES

Durante o desenvolvimento desta dissertação foram obtidos resultados que culminaram com a publicações de artigos e apresentação de pôsteres em veículos de divulgação científica. Os trabalhos descritos a seguir são diretamente relacionados ao tema de pesquisa desta dissertação. Além dos artigos já publicados, planeja-se o desenvolvimento de novos artigos: (i) analisar a abordagem desenvolvida considerando outros cenários de aplicações (e.g., hospitais, cidades ou fazendas inteligentes); (ii) aplicar cenários que envolvam tomada de decisão com base dos dados armazenados pela abordagem; (iii) análise das estratégias de conversão e comunicação de dados em aplicações *IoT*. A seguir, encontram-se os resultados publicados oriundos deste projeto:

- **BARROS, V. A.; ESTRELLA, J. C.; PRATES, L. B.; BRUSCHI, S. M.** . *An IoT-DaaS Approach for the Interoperability of Heterogeneous Sensor Data Sources*. In: *the 21st ACM International Conference, 2018, Montreal. Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWIM '18*, 2018. p. 275 (Full Paper - Qualis B1);
- **BARROS, V. A.; BAPTISTA, S. A.; BRUSCHI, S. M.; ESTRELLA, J. C.; MONACO, F. J.** . *An IoT Multi-Protocol Strategy for the Interoperability of Distinct Communication Protocols applied to Web of Things*. In: *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, 2019, Rio de Janeiro (Full Paper - Qualis B2);
- **MANO, L. Y.; BARROS, V. A.; NUNES, L. H.; SAWADA, L. O.; ESTRELLA, J. C.; UEYAMA, J.** . *Combining Layer-based Architecture and Communication for Emotion Monitoring in Healthcare*. In: *Mobile Information Systems (Impact Factor: 1.635)*;
- **BARROS, V. A.; SERRA, H. M. ; ESTRELLA, J. C.** . *Avaliação de Desempenho do Middleware Global Sensor Network (GSN) em execução em ambientes IoT-Fog*. In: *10ª Escola Regional de Alto Desempenho*, 2017, São Carlos. Erad-SP 2019 (Short Paper);

- **BARROS, V. A.**; NUNES, L. H. ; SOUZA, P. S. L. ; ESTRELLA, J. C. . Paralelização do Algoritmo Fast Non-Dominated Sorting para o Ranqueamento de Cidades de Acordo com Informações do Clima. In: 8ª Escola Regional de Alto Desempenho, 2017, São Carlos. Erad-SP 2017 (*Short Paper*);
- **BARROS, V. A.**; SERRA, H. M. ; ESTRELLA, J. C. . Avaliação de Desempenho do Global Sensor Network (GSN) para o Provisionamento de Serviços em Internet das Coisas em ambientes de Computação em Névoa. In: 10 Simposio Brasileiro de Computacao Ubiqua e Pervasiva, 2018, Natal, SBCUP 2018. (Pôster - Qualis B4); e
- **BARROS, V. A.**; ESTRELLA, J. C. . Uma abordagem para a Integração de Diferentes Fontes de Dados Provenientes de Redes de Sensores no Contexto de Internet das Coisas. EPPC 2017 (Pôster).

CONCLUSÕES

Esta dissertação apresentou um conjunto de mecanismos que auxiliam na interoperabilidade entre fontes heterogêneas de dados de sensores. Para isso foi apresentado o *Internet of Things Data as Service Module (IoTDSM)* que permitiu a integração de diferentes fontes e formato de dados utilizados por aplicações *IoT*. Foi apresentada também o *Internet of Things Multi-Protocol Message Broker (IoTM²B)* que possibilitou a integração do *IoTDSM* com diferentes protocolos de comunicação. Tais mecanismos objetivam possibilitar uma melhor interoperabilidade de aplicações *IoT*, proporcionando uma melhor integração na aquisição, processamento e armazenamento de dados de sensores. Além disso, com base no *IoTDSM* foi desenvolvido a arquitetura *ENLACE* que tinha como objetivo classificar as emoções de usuários de uma casa inteligente por meio de fotos da face e batimentos cardíacos.

A fim de cumprir com os objetivos propostos, foram conduzidos diferentes cenários de avaliação de desempenho. Primeiramente, investigou-se o *Middleware Global Sensor Network (GSN)* que serviu com base para os demais avaliações conduzidas. Em seguida, foi examinado o *IoTDSM* e *IoTM²B* de forma isolada, com a finalidade de verificar os gargalos relacionados à interoperabilidade. Além disso, é importante notar que todas as avaliações experimentais conduzidas evidenciaram que os mecanismos apresentam um bom desempenho, o que sinaliza a sua utilização em diferentes cenários de aplicações *IoT*. A seguir, são descritas as contribuições, limitações e perspectivas de trabalhos futuros desta dissertação.

7.1 Contribuições e Limitações do Trabalho

As seguintes contribuições foram alcançadas com esta dissertação:

- Definição de uma interface de comunicação (*API*) que possibilitou a integração com fontes distintas de dados de sensores. Além disso, definiu-se um mecanismo, chamado de *Wrapper*, que permitiu de forma flexível o pré-processamento dos dados de entrada,

viabilizando a sua inserção em modelos de banco de dados relacionais (*SQL*) e não relacionais (*NoSQL*);

- Desenvolvimento de um repositório de dados genérico que permitiu o gerenciamento de dados heterogêneos de sensores em sistemas de bancos de dados distintos (*SQL* ou *NoSQL*). Além do mais, foi apresentado um mecanismo multi-protocolos que permitiu a interoperabilidade entre os protocolos de comunicação *HTTP*, *MQTT* e *CoAP*;
- Aplicação de diferentes cenários de avaliação de desempenho que evidenciou a utilização das abordagens propostas em diferentes cenários de aplicações *IoT*. Além disso, os mecanismos propostos apresentaram um bom desempenho, o que possibilita o seu uso em aplicações de larga escala (Cidades, Hospitais ou Casas Inteligentes);

Entretanto, apesar das contribuições apresentadas neste trabalho as seguintes limitações precisam ser evidenciadas:

- Para a integração de novas fontes de dados ou protocolos de comunicação é necessária uma implementação prévia de novos *wrappers*/conectores. Este processo é feito por meio de codificação, o que impede que usuários leigos possam adaptar o sistema aos seus requisitos. Uma possível solução consiste na criação de sistema de *workflow*, onde *wrappers*/conectores, previamente desenvolvidos, possam ser conectados por meio de interface gráfica amigável, possibilitando assim, a construção de *pipelines* de conversão de dados simplificados e;
- O foco deste trabalho foi o desafio relacionado à interoperabilidade de aplicações *IoT*. Entretanto, outros requisitos são importantes, por exemplo, escalabilidade e segurança. Aplicações que necessitam de estabilidade horizontal em relação à estratégia de armazenamento e processamento de dados não são completamente suportadas. Para isso, novos estudos envolvendo a utilização de sistemas de bancos de dados e motores de processamento distribuído precisam ser analisados. Além disso, os mecanismos *IoTDSM* e *IoT^{M2B}* não oferecem um mecanismo de controle de acesso, o que torna inviável seu uso em sistema que envolvem dados sensíveis. Neste sentido, estratégias de controle de acesso e autorização precisam ser implementadas.

7.2 Trabalhos Futuros

Esta pesquisa apresentou mecanismos genéricos para interoperabilidade de aplicações *IoT*. Portanto, focou-se em avaliações pontuais de aplicações práticas de cenários de avaliações dos mecanismos propostos. Como perspectiva de trabalhos futuros conduziremos novos cenários de avaliações com objetivos que vão além de auxiliar a interoperabilidade. Por exemplo, pode-se instanciar novos cenários de avaliações, que tenham como foco aplicações em cidades, casas,

fazendas ou hospitais inteligentes. Contudo, novos requisitos precisam serem avaliados, como a escalabilidade, segurança e qualidade dos dados armazenados. Ademais, por se tratar de mecanismos mais focados em aplicações que em interação com usuário, é possível estender o trabalho para prover uma ferramenta de visualização de dados, por meio de *dashboards*, com o objetivo de auxiliar a tomada de decisão com base nos dados analisados. Por fim, outro possível trabalho futuro consiste na aplicação de modelos de otimização multi-objetivo ou de inteligência artificial (aprendizados supervisionadas, não supervisionados ou por reforço) que auxiliem na solução de problemas de otimização que envolvam aplicações de dados de sensores.

REFERÊNCIAS

AAZAM, M.; HUH, E.-N. Fog computing and smart gateway based communication for cloud of things. In: IEEE. **Future Internet of Things and Cloud (FiCloud), 2014 International Conference on**. [S.l.], 2014. p. 464–470. Citado nas páginas 26 e 33.

AHMED, M.; CHOWDHURY, A.; AHMED, M.; RAFEE, M. M. H. *et al.* An advanced survey on cloud computing and state-of-the-art research issues. **IJCSI International Journal of Computer Science Issues**, v. 9, n. 1, p. 1694–0814, 2012. Citado na página 42.

AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. A survey on sensor networks. **IEEE Communications magazine**, IEEE, v. 40, n. 8, p. 102–114, 2002. Citado nas páginas 15, 30 e 31.

AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys Tutorials**, v. 17, n. 4, p. 2347–2376, Fourthquarter 2015. ISSN 1553-877X. Citado na página 37.

ALAM, S.; NOLL, J. A semantic enhanced service proxy framework for internet of things. In: IEEE COMPUTER SOCIETY. **Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing**. [S.l.], 2010. p. 488–495. Citado na página 47.

ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R.; KONWINSKI, A.; LEE, G.; PATTERSON, D.; RABKIN, A.; STOICA, I. *et al.* A view of cloud computing. **Communications of the ACM**, ACM, v. 53, n. 4, p. 50–58, 2010. Citado na página 41.

ATLAM, H.; ALENEZI, A.; ALHARTHI, A.; WALTERS, R.; WILLS, G. *et al.* Integration of cloud computing with internet of things: challenges and open issues. 2017. Citado na página 42.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. Citado nas páginas 25 e 29.

BABU, S. M.; LAKSHMI, A. J.; RAO, B. T. A study on cloud based internet of things: Cloudiot. In: IEEE. **Communication Technologies (GCCT), 2015 Global Conference on**. [S.l.], 2015. p. 60–65. Citado na página 42.

BARI, N.; MANI, G.; BERKOVICH, S. Internet of things as a methodological concept. In: IEEE. **Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on**. [S.l.], 2013. p. 48–55. Citado na página 26.

BASSI, A.; HORN, G. Internet of things in 2020: A roadmap for the future. **European Commission: Information Society and Media**, 2008. Citado na página 29.

BHUYAN, B.; SARMA, H. K. D.; SARMA, N.; KAR, A.; MALL, R. Quality of service (qos) provisions in wireless sensor networks and related challenges. **Wireless Sensor Network**, Scientific Research Publishing, v. 2, n. 11, p. 861, 2010. Citado na página 26.

BOTTA, A.; DONATO, W. D.; PERSICO, V.; PESCAPÉ, A. Integration of cloud computing and internet of things: a survey. **Future Generation Computer Systems**, Elsevier, v. 56, p. 684–700, 2016. Citado na página 42.

BOTTERMAN, M. Internet of things: an early reality of the future internet. In: **Workshop Report, European Commission Information Society and Media**. [S.l.: s.n.], 2009. Citado na página 30.

BOVET, G.; HENNEBERT, J. Distributed semantic discovery for web-of-things enabled smart buildings. In: IEEE. **New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on**. [S.l.], 2014. p. 1–5. Citado na página 47.

BURANGE, A. W.; MISALKAR, H. D. Review of internet of things in development of smart cities with data management & privacy. In: IEEE. **Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in**. [S.l.], 2015. p. 189–195. Citado na página 35.

CHAQFEH, M. A.; MOHAMED, N. Challenges in middleware solutions for the internet of things. In: IEEE. **Collaboration Technologies and Systems (CTS), 2012 International Conference on**. [S.l.], 2012. p. 21–26. Citado nas páginas 39 e 41.

CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. In: **Mobile Networks and Applications**. [S.l.: s.n.], 2014. Citado nas páginas 25 e 38.

COUNCIL, N. Six technologies with potential impacts on us interests out to 2025. **Disruptive Civil Technologies2008**, 2008. Citado na página 31.

DATTA, S. K.; BONNET, C. Internet of things and m2m communications as enablers of smart city initiatives. In: IEEE. **Next Generation Mobile Applications, Services and Technologies, 2015 9th International Conference on**. [S.l.], 2015. p. 393–398. Citado na página 47.

DATTA, S. K.; BONNET, C.; NIKAEIN, N. An iot gateway centric architecture to provide novel m2m services. In: IEEE. **Internet of Things (WF-IoT), 2014 IEEE World Forum on**. [S.l.], 2014. p. 514–519. Citado nas páginas 48 e 53.

DAUBERT, J.; WIESMAIER, A.; KIKIRAS, P. A view on privacy & trust in iot. In: IEEE. **Communication Workshop (ICCW), 2015 IEEE International Conference on**. [S.l.], 2015. p. 2665–2670. Citado na página 35.

DESAI, P.; SHETH, A.; ANANTHARAM, P. Semantic gateway as a service architecture for iot interoperability. In: IEEE. **Mobile Services (MS), 2015 IEEE International Conference on**. [S.l.], 2015. p. 313–319. Citado nas páginas 48 e 53.

DOHR, A.; MODRE-OPSRIAN, R.; DROBICS, M.; HAYN, D.; SCHREIER, G. The internet of things for ambient assisted living. In: IEEE. **Information Technology: New Generations (ITNG), 2010 Seventh International Conference on**. [S.l.], 2010. p. 804–809. Citado na página 29.

FAN, P.-f.; ZHOU, G.-z. Analysis of the business model innovation of the technology of internet of things in postal logistics. In: IEEE. **Industrial Engineering and Engineering Management (IE&EM), 2011 IEEE 18Th International Conference on**. [S.l.], 2011. p. 532–536. Citado na página 26.

FERSI, G. Middleware for internet of things: A study. In: **2015 International Conference on Distributed Computing in Sensor Systems**. Institute of Electrical & Electronics Engineers (IEEE), 2015. Disponível em: <<http://dx.doi.org/10.1109/DCOSS.2015.43>>. Citado nas páginas 33 e 35.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. **Hypertext Transfer Protocol – HTTP/1.1**. United States: RFC Editor, 1999. Citado na página 36.

GANTZ, J.; REINSEL, D. Extracting value from chaos. **IDC iview**, v. 1142, p. 1–12, 2011. Citado na página 38.

GAVRAS, A.; KARILA, A.; FDIDA, S.; MAY, M.; POTTS, M. Future internet research and experimentation: the fire initiative. **ACM SIGCOMM Computer Communication Review**, ACM, v. 37, n. 3, p. 89–92, 2007. Citado na página 38.

GLOMBITZA, N.; MIETZ, R.; RÖMER, K.; FISCHER, S.; PFISTERER, D. Self-description and protocol conversion for a web of things. In: IEEE. **Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on**. [S.l.], 2010. p. 229–236. Citado na página 48.

GOMES, M.; RIGHI, R. da R.; COSTA, C. A. da. Internet of things scalability: Analyzing the bottlenecks and proposing alternatives. In: IEEE. **Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on**. [S.l.], 2014. p. 269–276. Citado na página 35.

GUINARD, D.; TRIFA, V.; KARNOUSKOS, S.; SPIESS, P.; SAVIO, D. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. **IEEE transactions on Services Computing**, IEEE, v. 3, n. 3, p. 223–235, 2010. Citado nas páginas 48 e 53.

GUINARD, D.; TRIFA, V.; WILDE, E. A resource oriented architecture for the web of things. In: IEEE. **Internet of Things (IOT), 2010**. [S.l.], 2010. p. 1–8. Citado na página 49.

GUPTA, V.; POURSOHI, A.; UDUPI, P. Sensor.network: An open data exchange for the web of things. In: IEEE. **Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on**. [S.l.], 2010. p. 753–755. Citado nas páginas 49 e 53.

HSU, Y.-C.; LIN, C.-H.; CHEN, W.-T. Design of a sensing service architecture for internet of things with semantic sensor selection. In: IEEE. **Ubiquitous Intelligence and Computing, 2014 IEEE 11th Intl Conf on and IEEE 11th Intl Conf on and Autonomic and Trusted Computing, and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UTC-ATC-ScalCom)**. [S.l.], 2014. p. 290–298. Citado na página 49.

JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. [S.l.]: John Wiley & Sons, 1990. Citado na página 67.

KESHAVARZI, A.; HAGHIGHAT, A. T.; BOHLOULI, M. Research challenges and prospective business impacts of cloud computing: A survey. In: **2013 IEEE 7th International Conference**

on **Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)**. [S.l.: s.n.], 2013. v. 02, p. 731–736. Citado na página 42.

KHAN, M.; SILVA, B. N.; HAN, K. A web of things-based emerging sensor network architecture for smart control systems. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 17, n. 2, p. 332, 2017. Citado nas páginas 46 e 53.

KHAN, R.; KHAN, S. U.; ZAHEER, R.; KHAN, S. Future internet: The internet of things architecture, possible applications and key challenges. In: **2012 10th International Conference on Frontiers of Information Technology**. Institute of Electrical & Electronics Engineers (IEEE), 2012. Disponível em: <<http://dx.doi.org/10.1109/FIT.2012.53>>. Citado nas páginas 33 e 34.

KIM, J.; LEE, J.-W. Openiot: An open service framework for the internet of things. In: IEEE. **Internet of Things (WF-IoT), 2014 IEEE World Forum on**. [S.l.], 2014. p. 89–93. Citado na página 40.

KORESHOFF, T. L.; ROBERTSON, T.; LEONG, T. W. Internet of things: A review of literature and products. In: **Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration**. New York, NY, USA: ACM, 2013. (OzCHI '13), p. 335–344. ISBN 978-1-4503-2525-7. Disponível em: <<http://doi.acm.org/10.1145/2541016.2541048>>. Citado na página 29.

KOTHARI, A.; BODDULA, V.; RAMASWAMY, L.; ABOLHASSANI, N. Dqs-cloud: A data quality-aware autonomic cloud for sensor services. In: IEEE. **Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on**. [S.l.], 2014. p. 295–303. Citado na página 50.

KULKARNI, R. V.; FORSTER, A.; VENAYAGAMOORTHY, G. K. Computational intelligence in wireless sensor networks: A survey. **IEEE communications surveys & tutorials**, IEEE, v. 13, n. 1, p. 68–96, 2011. Citado na página 30.

LIN, S.-Y.; CHUNG, C.-C.; HU, W.-C.; HUNG, C.; CHEN, S.-L.; LIN, T.-L. Automated knowledge discovery and semantic annotation for network and web services. **International Journal of Distributed Sensor Networks**, SAGE Publications Sage UK: London, England, v. 12, n. 7, p. 1550147716657925, 2016. Citado na página 50.

LOGMEIN, I. **Xively**. 2015. Disponível em: <<https://www.xively.com/>>. Citado na página 41.

NAGIB, A. M.; HAMZA, H. S. Sighted: A framework for semantic integration of heterogeneous sensor data on the internet of things. **Procedia Computer Science**, Elsevier, v. 83, p. 529–536, 2016. Citado nas páginas 50 e 53.

NAIK, N. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In: **2017 IEEE International Systems Engineering Symposium (ISSE)**. [S.l.: s.n.], 2017. p. 1–7. Citado nas páginas 36 e 37.

NUNES, L.; ESTRELLA, J.; NAKAMURA, L.; LIBARDI, R. de; FERREIRA, C.; JORGE, L.; PERERA, C.; REIFF-MARGANIEC, S. A distributed sensor data search platform for internet of things environments. **arXiv preprint arXiv:1606.07932**, 2016. Citado nas páginas 15, 56 e 57.

NUNES, L. H. **Métodos de análise de decisão multicritério para a seleção de recursos em ambientes IoT**. Tese (PhD thesis) — Instituto de Ciências Matemáticas e de Computação ICMC/USP, 2018. Citado nas páginas 15, 29, 30, 34, 35, 55 e 57.

- O'BYRNE, D.; BRENNAN, R.; O'SULLIVAN, D. Implementing the draft w3c semantic sensor network ontology. In: IEEE. **Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on**. [S.l.], 2010. p. 196–201. Citado na página 39.
- ORESTIS, A.; DIMITRIOS, A.; IOANNIS, C. Towards integrating iot devices with the web. In: IEEE. **Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on**. [S.l.], 2012. p. 1–4. Citado nas páginas 51 e 53.
- PARIDEL, K.; BAINOMUGISHA, E.; VANROMPAY, Y.; BERBERS, Y.; MEUTER, W. D. Middleware for the internet of things, design goals and challenges. **Electronic Communications of the EASST**, v. 28, 2010. Citado na página 38.
- PARWEKAR, P. From internet of things towards cloud of things. In: IEEE. **Computer and Communication Technology (ICCCCT), 2011 2nd International Conference on**. [S.l.], 2011. p. 329–333. Citado na página 26.
- PATIDAR, S.; RANE, D.; JAIN, P. A survey paper on cloud computing. In: IEEE. **Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on**. [S.l.], 2012. p. 394–398. Citado na página 42.
- PERERA, C.; JAYARAMAN, P. P.; ZASLAVSKY, A.; GEORGAKOPOULOS, D.; CHRISTEN, P. Sensor discovery and configuration framework for the internet of things paradigm. In: IEEE. **Internet of Things (WF-IoT), 2014 IEEE World Forum on**. [S.l.], 2014. p. 94–99. Citado nas páginas 51 e 53.
- PERERA, C.; VASILAKOS, A. V. A knowledge-based resource discovery for internet of things. **Knowledge-Based Systems**, Elsevier, v. 109, p. 122–136, 2016. Citado na página 46.
- PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Ca4iot: Context awareness for internet of things. In: IEEE. **Green Computing and Communications (Green-Com), 2012 IEEE International Conference on**. [S.l.], 2012. p. 775–782. Citado na página 51.
- PERERA, C.; ZASLAVSKY, A.; LIU, C.; COMPTON, M.; CHRISTEN, P.; GEORGAKOPOULOS, D. Sensor search techniques for sensing as a service architecture for the internet of things. **IEEE Sensors Journal**, v. 14, n. 2, p. 406–420, 2014. Citado na página 26.
- PETROLO, R.; MITTON, N.; SOLDATOS, J.; HAUSWIRTH, M.; SCHIELE, G. Integrating wireless sensor networks within a city cloud. In: IEEE. **Sensing, Communication, and Networking Workshops (SECON Workshops), 2014 Eleventh Annual IEEE International Conference on**. [S.l.], 2014. p. 24–27. Citado nas páginas 52 e 53.
- RAWAT, P.; SINGH, K. D.; CHAOUCHI, H.; BONNIN, J. M. Wireless sensor networks: a survey on recent developments and potential synergies. **The Journal of supercomputing**, Springer, v. 68, n. 1, p. 1–48, 2014. Citado na página 30.
- RAZZAQUE, M. A.; MILOJEVIC-JEVRIĆ, M.; PALADE, A.; CLARKE, S. Middleware for internet of things: a survey. **IEEE Internet of Things Journal**, IEEE, v. 3, n. 1, p. 70–95, 2016. Citado nas páginas 17, 38 e 39.
- SALEHI, A. **Design and implementation of an efficient data stream processing system**. Tese (Doutorado) — ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2010. Citado na página 39.

- SANDU, F.; COSTACHE, C.; BALAN, T. Semantic data aggregation in heterogeneous learning environments. In: IEEE. **Design and Technology in Electronic Packaging (SIITME), 2015 IEEE 21st International Symposium for**. [S.l.], 2015. p. 409–412. Citado nas páginas 52 e 53.
- SAYS, G. 4.9 billion connected "things" will be in use in 2015. **Gartner [online]. [cit. 2015-11-09]. Disponível z: <http://www.gartner.com/newsroom/id/2905717>**, 2014. (Accessed on 13/05/2017). Citado na página 31.
- SOLDATOS, J.; KEFALAKIS, N.; HAUSWIRTH, M.; SERRANO, M.; CALBIMONTE, J.-P.; RIAHI, M.; ABERER, K.; JAYARAMAN, P. P.; ZASLAVSKY, A.; ŽARKO, I. P. *et al.* Openiot: Open source internet-of-things in the cloud. In: **Interoperability and open-source solutions for the internet of things**. [S.l.]: Springer, 2015. p. 13–25. Citado na página 40.
- SUNDMAEKER, H.; GUILLEMIN, P.; FRIESS, P.; WOELFFLÉ, S. Vision and challenges for realising the internet of things. **Cluster of European Research Projects on the Internet of Things, European Commission**, 2010. Citado na página 31.
- Tolk, A. Interoperability, composability, and their implications for distributed simulation: Towards mathematical foundations of simulation interoperability. In: **2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications**. [S.l.: s.n.], 2013. p. 3–9. ISSN 1550-6525. Citado na página 35.
- VERMESAN, O.; FRIESS, P. **Internet of things: converging technologies for smart environments and integrated ecosystems**. [S.l.]: River Publishers, 2013. Citado na página 35.
- VERMESAN, O.; FRIESS, P.; GUILLEMIN, P.; GUSMEROLI, S.; SUNDMAEKER, H.; BASSI, A.; JUBERT, I. S.; MAZURA, M.; HARRISON, M.; EISENHAUER, M.; DOODY, P. Internet of things strategic research roadmap. **Internet of Things-Global Technological and Societal Trends**, p. 9–52, 2011. Citado na página 29.
- VERMESAN, O.; FRIESS, P.; GUILLEMIN, P.; SUNDMAEKER, H.; EISENHAUER, M.; MOESSNER, K.; GALL, F. L.; COUSIN, P. Internet of things strategic research and innovation agenda. **RIVER PUBLISHERS SERIES IN COMMUNICATIONS**, p. 7, 2013. Citado nas páginas 30 e 36.
- WHITMORE, A.; AGARWAL, A.; XU, L. D. The internet of things—a survey of topics and trends. **Information Systems Frontiers**, Springer, v. 17, n. 2, p. 261–274, 2015. Citado na página 29.
- WITCHALLS, C.; CHAMBERS, J. The internet of things business index: A quiet revolution gathers pace. **The Economist Intelligence Unit**. Retrieved from <http://www.economistinsights.com/analysis/internet-things-business-index>, 2013. (Accessed on 13/05/2017). Citado nas páginas 15, 31 e 32.
- XU, L. D.; HE, W.; LI, S. Internet of things in industries: A survey. **IEEE Transactions on industrial informatics**, IEEE, v. 10, n. 4, p. 2233–2243, 2014. Citado na página 26.
- YASSEIN, M. B.; SHATNAWI, M. Q.; ALJWARNEH, S.; AL-HATMI, R. Internet of things: Survey and open issues of mqtt protocol. In: **2017 International Conference on Engineering MIS (ICEMIS)**. [S.l.: s.n.], 2017. p. 1–6. Citado na página 37.

ZASLAVSKY, A.; PERERA, C.; GEORGAKOPOULOS, D. Sensing as a service and big data. **arXiv preprint arXiv:1301.0159**, 2013. Citado na página 26.

ZHANG, M.; SUN, F.; CHENG, X. Architecture of internet of things and its key technology integration based-on RFID. In: **2012 Fifth International Symposium on Computational Intelligence and Design**. Institute of Electrical & Electronics Engineers (IEEE), 2012. Disponível em: <<http://dx.doi.org/10.1109/ISCID.2012.81>>. Citado na página 33.

ZHANG, S.; YAN, H.; CHEN, X. Research on key technologies of cloud computing. **Physics Procedia**, Elsevier, v. 33, p. 1791–1797, 2012. Citado na página 42.

ZHOU, J.; LEPPANEN, T.; HARJULA, E.; YLIANTTILA, M.; OJALA, T.; YU, C.; JIN, H.; YANG, L. T. Cloudthings: A common architecture for integrating the internet of things with cloud computing. In: IEEE. **Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on**. [S.l.], 2013. p. 651–657. Citado na página 42.